

**Τμήμα  
Μηχανικών  
Πληροφορικής τ.ε.**

Τεχνολογικό Εκπαιδευτικό Ίδρυμα  
Δυτικής Ελλάδας

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**«Ανάπτυξη εφαρμογής κράτησης και  
αγοράς εισιτηρίου σε πολύ-κινηματογράφο»**

**Μουρουζίδης Ιωάννης**

Εποπτεύων Καθηγητής: Κος Τσακνάκης Ιωάννης

**ΠΑΤΡΑ – 2014**



Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, ...../...../.....

#### ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. ...., .....
2. ...., .....
3. ...., .....



## ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του Τμήματος «Μηχανικών Πληροφορικής Τ.Ε.» του «Τεχνολογικού Εκπαιδευτικού Ιδρύματος Ελλάδος» .

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω θερμά τον Επίκουρο Καθηγητή κ. Τσακνάκη Ιωάννη, για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα θέμα που με ενδιέφερε πολύ, καθώς και για τις αμέτρητες γνώσεις που απέκτησα και συνδύασα με αυτές που αποκτήθηκαν από το πρόγραμμα σπουδών, ώστε να φέρω τον στόχο μου εις πέρας.

Τέλος, θα ήθελα να ευχαριστήσω όλους τους ανθρώπους γύρω μου για την συμπαράσταση και την στήριξη που μου παρείχαν, παρά τις δυσκολίες που αντιμετώπισα, ώστε να έχω την ευκαιρία να γράφω αυτόν τον πρόλογο.

Σας ευχαριστώ όλους,

Μουρουζίδης Ιωάννης  
Πάτρα, 2014



## ΠΕΡΙΛΗΨΗ

Σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη μίας διαδικτυακής εφαρμογής κράτησης και αγοράς εισιτηρίου σε πολυκινηματογράφο και ως εργαλείο του διαχειριστή θα στοχεύει ταυτόχρονα στην εξυπηρέτηση του πελάτη. Υλοποιείται κυρίως σε PHP και MySQL, όπως παρουσιάστηκαν στα πλαίσια του Προπτυχιακού Προγράμματος Σπουδών.

Η ανάπτυξη της εφαρμογής χωρίζεται σε τέσσερα επίπεδα, ανάλογα με την ιδιότητα και τον σκοπό του προσώπου που την χρησιμοποιεί. Σε αυτό το σημείο κρίθηκε σκόπιμο να υπάρχουν τρία ήδη επιπέδων χρηστών της εφαρμογής. Το επίπεδο του «Διαχειριστή», που θα είναι ο υπεύθυνος του υλικού που δημοσιεύεται στην σελίδα, του «Υπάλληλου», υπεύθυνου για την εξυπηρέτηση του πελάτη, ο «Πελάτης», που διαχωρίζεται σε δύο επίπεδα, το «Πελάτης» του πολυκινηματογράφου που εξυπηρετείται από τον «Υπάλληλο» και το «Μέλος» της διαδικτυακής εφαρμογής που αναλαμβάνει την αυτοεξυπηρέτηση των δικών του κρατήσεων και αγορών .

Συγκεκριμένα, για τις ανωτέρω ανάγκες, τα τρία επίπεδα χρηστών, με διαφορετικά δικαιώματα και αρμοδιότητες, είναι:

1. Ο Διαχειριστής: είναι ο κύριος υπεύθυνος στην διαχείριση της ιστοσελίδας, θα έχει την δυνατότητα να ελέγχει τις ταινίες, τις προβολές και τον αριθμό των αιθουσών που βρίσκονται στην Βάση Δεδομένων, την επίβλεψη όλων των χρηστών, συμπεριλαμβανομένων και των υπαλλήλων, καθώς και πρόσβαση σε χρήσιμα στατιστικά στοιχεία που θα βοηθούν στην διαχείριση της επιχείρησης.
2. Ο Υπάλληλος: είναι υπεύθυνος για την εξυπηρέτηση του πελάτη, θα έχει την δυνατότητα να ελέγχει την κίνηση στο σύστημα κρατήσεων και έκδοσης εισιτηρίων είτε είναι μέλη της ιστοσελίδας είτε απλοί πελάτες.
3. Το Μέλος: οποιοσδήποτε πελάτης το επιθυμεί μπορεί να γίνει μέλος ώστε να του παρέχονται υπηρεσίες για κράτηση σε μια προβολή, σε όποια θέση/εις και μέρα επιθυμεί, με το δικαίωμα είτε ακύρωσης της κράτησης είτε αποπληρωμής ηλεκτρονικά είτε πληρωμής στο κατάστημα επιδεικνύοντας την Κάρτα Μέλους.

Στα επόμενα κεφάλαια θα παρουσιαστεί αναλυτικά η σχεδίαση της ιστοσελίδας, η δομή και η χρήση της. Για την λειτουργία και την συνεργασία των παραπάνω, θα δημιουργηθεί μια Βάση Δεδομένων και ένα, φιλικό προς τον χρήστη γραφικό περιβάλλον, με την βοήθεια των προγραμμάτων XAMPP και Notepad++ και των γλωσσών προγραμματισμού PHP, MySQL, HTML και CSS.





## ABSTRACT

The purpose of this thesis is to develop a ticket reservation and purchase web application in a multi-screen cinema and will consist as a manager tool aiming simultaneously to customer service. Mainly, was implemented with PHP and MySQL, as presented in the context of the curriculum.

Application development is divided into four layers, depending on nature and purpose of the person using it. At this point it was considered desirable to have three types of application user levels. The "Administrator" level, which will be responsible for the material posted on the website, the "Employee" level, which will be responsible for customer service, the "Client" level, which is divided into two layers, the "Client" layer of multi-screen cinema which is served by the "Employee" and the "Member" layer which is self-handling its own reservations and purchases.

Specifically, for the above needs, the three user layers, which have different rights and responsibilities, are:

1. The Administrator: he is primarily responsible for the administration of the website, he will have the ability to control the movies, the screenings and the number of halls which are in the database, overseeing all users, including employees, as well as access to useful statistics which will facilitate business management.
2. The employee: he is responsible for customer service, he will have the ability to control the reservations and ticketing system changes either from Members of the website or plain Customers.
3. The member: any customer, who wishes to become a member can actually become one, so that services for making reservations are offered, in any seat and on any day he wishes, with the right either to cancel the reservation or pay online or pay at the store showing the Membership Card.

In the following chapters, the design, the structure and the use of the website will be described in detail. . A Database and a user friendly Graphical User Interface will be created for the operation and cooperation of the above , with the assistance of programs XAMPP and Notepad++ and programming languages PHP, MySQL, HTML and CSS.



# ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	ΕΙΣΑΓΩΓΗ .....	xiii
ΚΕΦΑΛΑΙΟ 2	ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΣΧΕΔΙΑΣΗ .....	xv
2.1	Σκοπός .....	xv
2.2	Στόχοι .....	xvi
2.3	Δομή της εφαρμογής .....	xvii
2.3.1	Ο Διαχειριστής .....	xvii
2.3.2	Ο Υπάλληλος .....	xviii
2.3.3	Το Μέλος .....	xviii
2.3.4	Ο Επισκέπτης .....	xix
2.4	Από την ανάλυση στον σχεδιασμό .....	xix
2.4.1	Ο Διαχειριστής .....	xx
2.4.2	Ο Υπάλληλος .....	xxii
2.4.3	Το Μέλος .....	xxiii
2.4.4	Ο επισκέπτης .....	xxiv
ΚΕΦΑΛΑΙΟ 3	ΕΠΙΛΟΓΗ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ ΛΟΓΙΣΜΙΚΟΥ .....	xxv
3.1	Ο εξυπηρετητής .....	xxv
3.2	Οι Γλώσσες Προγραμματισμού .....	xxviii
3.3	Ανάπτυξη του Κώδικα .....	xxix
ΚΕΦΑΛΑΙΟ 4	ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	xxxi
4.1	Γενικά .....	xxxi
4.2	Εργαλεία Διαχειριστή .....	xxxv
4.2.1	Προσθήκη .....	xxxv
4.2.2	Επισκόπηση .....	xxxvi
4.2.3	Αίθουσες .....	xxxviii
4.2.4	Στατιστικά .....	xxxix
4.2.5	Χρήστες .....	xli
4.3	Εργαλεία Υπαλλήλου .....	xlili
4.3.1	Επισκόπηση .....	xlili
4.3.2	Χρήστες .....	xliv
4.3.3	Κράτηση .....	xlvii
4.3.4	Εξυπηρέτηση .....	xliv
4.4	Εργαλεία Μέλους .....	l
4.4.1	Κράτηση .....	l
4.4.2	Προφίλ .....	l

4.5	Βοηθητικές Λειτουργίες.....	liv
4.5.1	Βοηθητικά Μηνύματα .....	liv
4.5.2	Ενημερωτικά Μηνύματα .....	liv
ΚΕΦΑΛΑΙΟ 5 ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....		lvi
5.1	header.php .....	lvi
5.2	footer.php .....	lviii
5.3	login.php .....	lix
5.4	logout.php.....	lx
5.5	index.php .....	lx
5.6	screenings.php .....	lxi
5.7	soon.php.....	lxi
5.8	contact.php .....	lxi
5.9	reservations.php .....	lxii
5.10	register.php.....	lxiii
5.11	userprofile.php .....	lxiv
5.12	adminadd.php.....	lxvi
5.13	overview.php .....	lxvi
5.14	halls.php .....	lxviii
5.15	stats.php .....	lxviii
5.16	stats.php .....	lxx
5.17	service.php .....	lxxi
5.18	buycredits.php .....	lxxiii
5.19	availability.php .....	lxxiv
5.20	printclubcard.php .....	lxxvii
5.21	dbcon.php.....	lxxviii
5.22	functions.php .....	lxxviii
ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ.....		cxxi
ΚΕΦΑΛΑΙΟ 7 ΒΙΒΛΙΟΓΡΑΦΙΑ .....		cxxiii
7.1	Βιβλία.....	cxxiii
7.1.1	Ελληνική Βιβλιογραφία .....	cxxiii
7.1.2	Ξένη Βιβλιογραφία.....	cxxiii
7.2	Πηγές στο Διαδίκτυο .....	cxxiii

## ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ

Η μορφή του διαδικτύου, όπως το ξέρουμε σήμερα, ξεκίνησε το 1989 ως απόγονος ενός στρατιωτικού προγράμματος επικοινωνίας (ARPAnet) και ενός δικτύου πανεπιστημίων (NSFnet). Προβλέποντας την ραγδαία ανάπτυξη που θα ακολουθούσε, δημιουργήθηκαν πολλές επιχειρήσεις κολοσσοί, με το πρώτο ηλεκτρονικό κατάστημα να ιδρύεται το 1995. Έτσι, πέρα από εργαλείο ενημέρωσης και επικοινωνίας, το διαδίκτυο εξελίσσεται σε άλλο ένα μέσω διακίνησης προϊόντων και υπηρεσιών.

Μόλις το 1998 ανοίγει και το πρώτο ηλεκτρονικό κατάστημα στην Ελλάδα και μέχρι το 2014 λειτουργούν πάνω από 4000. Παρά το ελληνικό χρέος, την οικονομική κατάσταση της χώρας και το καθημερινό κλείσιμο επιχειρήσεων ανεξαρτήτως μεγέθους, ο συνολικός τζίρος των ηλεκτρονικών αγορών B2C (Business To Consumer) των Ελλήνων καταναλωτών, λαμβάνοντας υπόψη την μεταβολή του από το 2008 έως το 2013, παρουσιάζει μια θεαματική αύξηση του 600%, με τα επίμαχα έτη 2012-2013 να αυξάνεται κατά 25 ποσοστιαίες μονάδες και να είναι 3,2 δις €. Όσον αφορά τα επόμενα χρόνια εκτιμάται ότι υπό προϋποθέσεις, ο ετήσιος τζίρος θα φτάσει τα 6 δις €. Είναι, επίσης, σημαντικό να αναφερθεί ότι στην κατάσταση προτιμήσεων των χρηστών που χρησιμοποιούν το διαδίκτυο ως μέσο αγορών, η κατηγορία «Εισιτήρια για εκδηλώσεις (συναυλίες, θέατρα κτλ.)» κατέχει την 5<sup>η</sup> θέση της κατάταξης και ήταν μία από τις περισσότερο αναπτυσσόμενες για το 2013.<sup>1</sup>

Αναμφίβολα λοιπόν, μία επιχείρηση όπως ο πολύ-κινηματογράφος, οφείλει να επιδιώξει την είσοδο στον χώρο του διαδικτυακού εμπορίου, ώστε να εκμεταλλευτεί τα οφέλη και την προτίμηση των χρηστών για αυτόν τον χώρο. Σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη μίας διαδικτυακής εφαρμογής που θα εισάγει μία επιχείρηση αυτού του χώρου στο διαδίκτυο, επιτρέποντας τόσο είσοδο στις διαδικτυακές συναλλαγές όσο και εργαλεία και διευκολύνσεις που θα προσφέρει στον χρήστη, αλλά και στην διαχείριση της ιστοσελίδας και της επιχείρησης.

Αρχικά, ένα από τα εργαλεία που χρησιμοποιήθηκαν είναι το XAMPP, το οποίο είναι ένα σύνολο προγραμμάτων ΕΛ/ΛΑΚ, που μπορεί να εγκατασταθεί σε οποιοδήποτε λειτουργικό σύστημα και περιέχει, εκτός άλλων, τον εξυπηρετητή (server) Apache, το Σύστημα Διαχείρισης Βάσης Δεδομένων MySQL, έναν διερμηνέα για κώδικα γραμμένο σε PHP και έναν εξυπηρετητή κίνησης αρχείων ftp. Το επόμενο βήμα ήταν η επιλογή κειμενογράφου που να υποστηρίζει γλώσσες προγραμματισμού ιστοσελίδων PHP, HTML, CSS και JavaScript. Η τελική επιλογή ήταν το ΕΛ/ΛΑΚ Notepad++, το οποίο τονίζει τις λέξεις που ανήκουν στις παραπάνω γλώσσες προγραμματισμού, αλλά ο κύριος λόγος που επιλέχθηκε ήταν η δυνατότητα επεξεργασίας πολλαπλών γραμμών, αυτόματης συμπλήρωσης και πελάτη ftp (client). Ακόμη, δημιουργήθηκε εικονικός υπολογιστής που παρέχεται δωρεάν από το «Υπολογιστικό Νέφος της Εκπαίδευσης» (Okeanos Cloud Computing Service) του Εθνικού Δικτύου Έρευνας και Τεχνολογίας (GRNet/ΕΔΕΤ) όπου εγκαταστάθηκε ο εξυπηρετητής. Τέλος, δημιουργήθηκε λογαριασμός, στην δωρεάν υπηρεσία στατικής διεύθυνσης NO-IP, για κατοχύρωση μοναδικής διεύθυνσης πρόσβασης στην εφαρμογή.

Ουσιαστικά, πρόκειται για ένα σύστημα προγραμμάτων, εργαλείων και υπηρεσιών τα οποία συνεργάζονται ώστε να παρέχουν οργανωμένη πληροφορία σε συνδυασμό με ένα φιλικό γραφικό περιβάλλον. Με αυτό τον τρόπο θα επιτρέπεται στον χρήστη να βλέπει τις ταινίες οι οποίες θα είναι διαθέσιμες προσεχώς και τις ταινίες που είναι διαθέσιμες για

<sup>1</sup> | Πηγή: ELTRUN (Εργαστήριο Ηλεκτρονικού Εμπορίου και Επιχειρείν, Οικονομικό Πανεπιστήμιο Αθηνών)  
<http://www.eltrun.gr/meletes/annual-e-commerce-survey/>  
<http://www.eltrun.gr/meletes/e-shops-survey/>

προβολή, μαζί με κάποιες πληροφορίες ανά ταινία. Επίσης, θα έχει την δυνατότητα αφού προμηθευτεί μία κάρτα μέλους, να μπορεί να προσθέσει διαδικτυακά πόντους (credits), που θα του δίνουν πρόσβαση στο σύστημα κράτησης. Σε αυτό το σημείο ο «Χρήστης - Μέλος» έρχεται σε επαφή με τις προβολές, είτε ανά ταινία, είτε ανά ημερομηνία και αφού επιλέξει όποια προβολή επιθυμεί, θα εμφανίζεται μια εικονική αίθουσα με ευδιάκριτα χαρακτηριστικά για την τοποθεσία των διαδρόμων και της οθόνης. Θα είναι στην κρίση του να επιλέξει ποιες θέσεις προτιμά, με μόνη προϋπόθεση να μην έχουν επιλεγεί ήδη από κάποιον άλλο. Στην συνέχεια θα εμφανίζεται μία φόρμα επιβεβαίωσης με τον τίτλο της ταινίας, την ώρα και τις θέσεις τις οποίες επέλεξε και θα καλείται να προβεί σε κράτηση ή αγορά, δίνοντάς του την δυνατότητα ακύρωσης της διαδικασίας κράτησης.

Σε αυτό το σημείο δημιουργήθηκε ο προβληματισμός με ποιον τρόπο θα γίνεται η διαχείριση του μέχρι σήμερα συστήματος κρατήσεων. Για αυτό το λόγο δημιουργώντας το επίπεδο χρήστη «Υπάλληλος», του δίνεται η δυνατότητα, εκτός από την πρόσβαση στις κρατήσεις που έγιναν διαδικτυακά, να μπορεί να ακολουθήσει την ίδια διαδικασία για κράτηση όπως ένας απλός χρήστης. Στο σημείο όμως που εμφανίζεται η φόρμα επιβεβαίωσης, θα μπορεί να καταχωρήσει σε ένα πεδίο κειμένου, που εμφανίζεται μόνο σε αυτόν, το όνομα μίας κράτησης που έγινε, χάριν παραδείγματος, τηλεφωνικά. Με αυτόν τον τρόπο, ο «Υπάλληλος» μπορεί σε δεύτερο χρόνο κατά την άφιξη του πελάτη, σε αντίστοιχο σημείο του μενού, είτε να προβεί σε αναζήτηση κράτησης ονομαστικά, είτε να εμφανίσει τις κρατήσεις οποιουδήποτε μέλους με την επίδειξη της «Κάρτας Μέλους». Με αντίστοιχη διαδικασία γίνεται και ακύρωση κρατήσεων.

Βέβαια, απαραίτητο ήταν να δημιουργηθεί και το επίπεδο του «Διαχειριστή», ώστε να μπορεί να συντηρεί, να επιβλέπει και να ενημερώνει όλα τα παραπάνω. Στο μενού πλοήγησής του εμφανίζονται περισσότερες επιλογές, οι οποίες του δίνουν πρόσβαση σε εργαλεία τέτοιου είδους ώστε να μπορεί να προσθέσει, να επεξεργαστεί και να διαγράψει ταινίες, προβολές και αίθουσες. Επιπλέον θα μπορεί να επιβλέπει την δραστηριότητα στην σελίδα τόσο των «Χρηστών» όσο και των «Υπαλλήλων». Πιο συγκεκριμένα, μπορεί να επιβλέπει ημερήσιους και συνολικούς τζίρους κάθε χρήστη, ενώ στην περίπτωση που ο χρήστης του οποίου προβάλλει το προφίλ είναι «Υπάλληλος», ο ημερήσιος τζίρος θα αντιστοιχεί στο ταμείο του. Ακόμη, ένα πολύ σημαντικό εργαλείο στο οποίο έχει πρόσβαση είναι το μενού των στατιστικών. Πρόκειται για ένα ζωτικής σημασίας μέρος της εφαρμογής, που του παρέχει πολύτιμη πληροφορία όχι μόνο για την πορεία της ιστοσελίδας, αλλά και για την εποπτεία της επιχείρησης και την οργάνωση στρατηγικών μάρκετινγκ. Με στόχο λοιπόν την κάλυψη των διοικητικών αναγκών της επιχείρησης, δίνεται η δυνατότητα στον «Διαχειριστή» να εμφανίσει τα έσοδα, μεταξύ οποιονδήποτε ημερομηνιών ή συγκεκριμένης μέρας, από την στιγμή που ξεκίνησε η λειτουργία της εφαρμογής. Τέλος, δημιουργήθηκε ένα σύστημα συλλογής στοιχείων από την Βάση Δεδομένων, που επιτρέπει την προβολή εσόδων, πλήθους και ποσοστών ανά ταινία, ηλικία, ημέρα της εβδομάδας και διαστημάτων ωρών λειτουργίας του πολύ-κινηματογράφου.

Έχοντας λοιπόν όλα τα παραπάνω υπόψη, σε κάθε κεφάλαιο αυτής της εργασίας θα αναλυθούν τα μέσα που χρησιμοποιήθηκαν για την δημιουργία της διαδικτυακής εφαρμογής, ο τρόπος υλοποίησής της καθώς και προβληματισμοί που γεννήθηκαν κατά την διάρκεια της σχεδίασης και πως αντιμετωπίστηκαν.

## ΚΕΦΑΛΑΙΟ 2 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΣΧΕΔΙΑΣΗ

Σε αυτό το κεφάλαιο θα αναλυθούν ο σκοπός και οι στόχοι που τέθηκαν για την δημιουργία της διαδικτυακής εφαρμογής κατά την διάρκεια του σχεδιασμού αλλά και του σταδιακού ελέγχου λειτουργίας της εφαρμογής. Θα αναλυθεί η δομή της εφαρμογής και πώς προέκυψε το κάθε τμήμα της από την ανάλυση του προβλήματος στον τελικό σχεδιασμό.

### 2.1 Σκοπός

Ο κύριος σκοπός της πτυχιακής εργασίας είναι ο σχεδιασμός και η ανάπτυξη μίας διαδικτυακής εφαρμογής που θα εισάγει έναν πολύ-κινηματογράφο στην διαδικτυακή κράτηση ή/και αγορά εισιτηρίου. Δεύτερο, αλλά εξίσου σημαντικό, είναι αυτή η εφαρμογή να προσφέρει μία σειρά από εργαλεία διευκόλυνσης και διαχείρισης από τον εκάστοτε επιχειρηματία, που θα χρησιμοποιείται και από τους υφιστάμενους της επιχείρησης μέχρι την κατώτερη βαθμίδα του χρήστη, ώστε να διευκολύνει τις συναλλαγές του και να του δίνει τον έλεγχο επιλογών που δεν θα είχε από το φυσικό κατάστημα.

Ειδικότερα, η πρώτη απαίτηση που γεννάται είναι η ύπαρξη ενός υπεύθυνου για το υλικό που προστίθεται στην ιστοσελίδα αλλά και η συντήρηση και η ενημέρωση αυτού. Πρωταρχικός σκοπός δηλαδή είναι ο «Διαχειριστής» να μπορεί να έχει στην διάθεση του εργαλεία που θα καθιστούν εύκολη την ολοκλήρωση της ευθύνης του προς την επιχείρηση. Ο τρόπος να καταστεί αυτή η εφαρμογή «εργαλείο» είναι η παροχή, με απλό τρόπο, της δυνατότητας προσθήκης και επεξεργασίας ταινίας, προβολής και αίθουσας, η ύπαρξη συγκεντρωτικών πινάκων στους οποίους θα εμφανίζονται με φίλτρα οι αποθηκευμένες πληροφορίες ανάλογα με το είδος τους. Σημαντική όμως είναι και η διαχείριση του συστήματος κρατήσεων που θα του επιτρέπει να ακυρώνει τις κρατήσεις οι οποίες δεν έχουν πληρωθεί σε συγκεκριμένο χρονικό διάστημα πριν από μία προβολή, σύμφωνα με την κρίση του. Πρέπει ακόμη να έχει την δυνατότητα να επιβλέπει τους υφιστάμενους του («Υπάλληλους») και τα «Μέλη», όσον αφορά το ιστορικό των κρατήσεων, συναλλαγών και τα έσοδα που έχει από τον καθένα. Τέλος, θα πρέπει να μπορεί εύκολα να αναζητά χρήστες και κρατήσεις, να μπορεί να ελέγχει την εγκυρότητα ενός εισιτηρίου ή μίας κάρτας μέλους, αλλά και να έχει πρόσβαση σε στατιστικά στοιχεία που θα καθιστούν τον προγραμματισμό και την διεύθυνση της επιχείρησης πιο εύκολη.

Δεύτερος, αλλά εξίσου σημαντικός σκοπός, είναι η παροχή εργαλείων στον «Υπάλληλο». Πρέπει να έχει ολοκληρωτική πρόσβαση στο σύστημα κρατήσεων, το οποίο θα πρέπει να προσφέρει την ευκολία της διαχείρισης και εκτός διαδικτύου. Είναι σημαντικό να σχεδιαστεί ένα ενοποιημένο σύστημα που θα δέχεται κρατήσεις, είτε διαδικτυακά, είτε τηλεφωνικά ή με φυσική παρουσία του πελάτη. Συγχρόνως θα παρέχει εργαλεία για την εύκολη ανεύρεση και έκδοση εισιτηρίων ανεξάρτητα από τον τρόπο που έγινε η κράτηση. Αναγκαία επίσης είναι η πρόσβαση στις συνολικές κρατήσεις ανά προβολή και ο έλεγχος μη επιβεβαιωμένων ή μη εξοφλημένων κρατήσεων και η δυνατότητα ακύρωσης τους σε εύλογο χρονικό διάστημα πριν την έναρξη της προβολής, ώστε να μην δεσμεύονται θέσεις που δεν θα καλυφθούν.

Τελικός σκοπός είναι η διευκόλυνση και η ευχαρίστηση του πελάτη. Θα πρέπει να του παρασχεθούν εργαλεία τα οποία όχι μόνο θα τον εξυπηρετούν περισσότερο από τους

παραδοσιακούς μη-διαδικτυακούς τρόπους, αλλά θα είναι συγχρόνως εύχρηστα και θα του δίνουν περισσότερες επιλογές που διαφορετικά δεν θα είχε. Με άλλα λόγια να μπορεί να πληροφορηθεί για τις προβολές που είναι ενεργές, αλλά και για προβολές που θα είναι διαθέσιμες στο μέλλον. Το σύστημα κράτησης πρέπει να είναι απλό στην χρήση λόγω του εύρους ηλικίας πελατών στους οποίους απευθύνεται ένας πολύ-κινηματογράφος. Εκείνο που έχει όμως ιδιαίτερη σημασία είναι να μην υποχρεούται να εξοφλήσει εξ αρχής την κράτηση του αλλά να είναι στην δική του επιλογή αν θα γίνει διαδικτυακά έστω και σε δεύτερο χρόνο ή στο κατάστημα. Επίσης σημαντική είναι η δυνατότητα επιλογής συγκεκριμένων θέσεων που δεν προσφέρει η τηλεφωνική κράτηση.

## 2.2 Στόχοι

Κατανοώντας τους σκοπούς της ζητούμενης εφαρμογής, ο πρώτος στόχος που τέθηκε, αφορούσε το γραφικό περιβάλλον. Πρόκειται για μία διαδικτυακή εφαρμογή την οποία θα επισκέπτονται χρήστες όλων των ηλικιών, τόσο για απλή άντληση πληροφοριών, όσο και για πιο πολύπλοκη χρήση της. Αφενός μεν θα έπρεπε να υπάρχει μία ισορροπία στην φιλικότητα του γραφικού περιβάλλοντος και στην ευκολία χρήσης, αφετέρου δε να είναι καλαίσθητη και συγχρόνως απλή στην πλοήγηση εξαιτίας του εύρους ηλικιών που πιθανό να την χρησιμοποιεί.

Είναι αλήθεια ότι το επόμενο προφανές πρόβλημα ήταν η αποθήκευση και η μεταφορά δεδομένων από και προς την ιστοσελίδα, αλλά η λύση του προβλήματος ήταν εξίσου προφανής, η ύπαρξη ενός Συστήματος Διαχείρισης Βάσης Δεδομένων. Ο αμέσως επόμενος στόχος που τέθηκε ήταν ο σχεδιασμός μίας Βάσης Δεδομένων. Κατά την διαδικασία αυτή δεν πρέπει μόνο να γίνει πρόβλεψη των αναγκών της εφαρμογής αλλά και το πώς τα δεδομένα κάθε πίνακα θα αλληλοσυνδέονται και θα ενημερώνονται αυτόματα, αν προκύψει αλλαγή.

Με αφετηρία την ύπαρξη ενός γραφικού περιβάλλοντος και μίας Βάσης Δεδομένων, διαπιστώθηκε πως η εφαρμογή αυτή θα έχει ένα σύνολο χρηστών που θα χρειάζονται πρόσβαση σε πληροφορίες και εργαλεία τα οποία δεν θα έπρεπε να είναι ελεύθερα στον οποιονδήποτε επισκέπτη. Έτσι προέκυψε η ανάγκη ύπαρξης ενός συστήματος εγγραφής και σύνδεσης χρηστών, ώστε να χωριστούν οι πληροφορίες από τα οφέλη των «Μελών». Εξίσου σημαντικός ήταν και ο διαχωρισμός του απλού χρήστη που αναζητά πληροφορίες από τον υπεύθυνο της διαδικτυακής εφαρμογής ο οποίος πρέπει να παρέχει πληροφορίες. Κατά την διάρκεια ανάλυσης του τρόπου με τον οποίο θα δημιουργούνταν τα επίπεδα των χρηστών προέκυψε αναντίρρητα πως ο «Υπάλληλος» δεν θα έπρεπε να έχει πρόσβαση σε όλα τα εργαλεία και τα στοιχεία που έχει στην διάθεσή του ένας διαχειριστής. Έτσι τέθηκε ο στόχος δημιουργίας τριών επιπέδων χρηστών «Μέλος», «Υπάλληλος», «Διαχειριστής».

Είναι φανερό ότι εκπληρώνοντας τα παραπάνω, πρέπει να δημιουργηθούν όλα τα εργαλεία για τα οποία προορίζεται η εφαρμογή, για κάθε περίπτωση και κάθε επίπεδο χρήστη. Κατά συνέπεια πρέπει να δημιουργηθεί η προσθήκη, το σύστημα προσθήκης στοιχείων στην ιστοσελίδα, ύστερα να δημιουργηθεί όπως περιγράφηκε ένα σύστημα κρατήσεων και τέλος τα μέσα διαχείρισης τους και ο τρόπος συλλογής και παρουσίασης των στατιστικών της επιχείρησης.



## 2.3 Δομή της εφαρμογής

Όπως προαναφέρθηκε, η τελική σκέψη είναι η δημιουργία τριών επιπέδων χρηστών της ιστοσελίδας που υλοποιείται: ο «Διαχειριστής», ο «Υπάλληλος» και το «Μέλος». Σε αυτό το σημείο θα αναλυθούν οι αρμοδιότητες και τα εργαλεία που θα έχει στην διάθεση του κάθε επίπεδο. Ωστόσο, η ανάλυση των επιπέδων θα γίνει ιεραρχικά, που σημαίνει πως ότι δικαιώματα πρόσβασης έχει το κατώτερο επίπεδο χρήστη θα το έχουν και τα ανώτερα. Ωστόσο, σε κάθε επίπεδο θα αναλυθούν τα σημεία της εφαρμογής τα οποία είναι σημαντικότερα και εμπίπτουν στο ενδιαφέρον ή στις αρμοδιότητές του.

### 2.3.1 Ο Διαχειριστής

Κατανοώντας το μέγεθος της πληροφορίας που θα πρέπει να συντηρεί ο «Διαχειριστής» είναι συνετό να έχει πρόσβαση σε εργαλεία που θα καθιστούν την ευθύνη του πιο εύκολη. Έτσι, θα έχει στην διάθεση του ένα μενού στο οποίο θα έχει πρόσβαση μόνο αυτός και θα συγκεντρώνει όλες τις λειτουργίες ανάλογα με την φύση της κάθε μίας ενέργειας που θέλει να εκτελέσει.

Αναλυτικότερα θα μπορεί μέσα από ένα σύνολο φορμών να προσθέσει μία ταινία με το σύνολο των στοιχείων της: τίτλος, διάρκεια, κατηγορία, κατάσταση, υπόθεση κτλ. και να την προσθέσει στην Βάση Δεδομένων του πολύ-κινηματογράφου. Θα μπορεί επίσης να προσθέσει μία προβολή καταχωρώντας τον τίτλο, την επιθυμητή αίθουσα, ημερομηνία, ώρα και τέλος να προσθέσει μία αίθουσα συμπληρώνοντας την χωρητικότητα της.

Εξίσου σημαντική όμως είναι και η επισκόπηση όλων αυτών, οπότε θα πρέπει να έχει πρόσβαση σε καθένα από τα παραπάνω με εύχρηστα φίλτρα. Θα μπορεί να προβάλλει τις ταινίες ανά κατάσταση στην οποία έχουν καταχωρηθεί εμφανίζοντας τις σημαντικότερες πληροφορίες για την κάθε μία σε πίνακα. Αντίστοιχα, οι προβολές θα παρουσιάζονται ανά ημερομηνία με αύξουσα σειρά για πιο εύκολη πρόσβαση. Αναγκαίο είναι να μπορεί εύκολα την στιγμή που θα βρει κάποια λάθος εγγραφή και επιθυμεί να επεξεργαστεί ή να διαγράψει, είτε ταινία, είτε προβολή, να μπορεί να το κάνει πολύ εύκολα από αυτό το σημείο.

Αφού έχει αναλυθεί η επεξεργασία της πληροφορίας που παρέχει και συντηρεί, είναι αναγκαία και η επόπτευση των «Υπαλλήλων» και των «Μελών». Απαραίτητο εργαλείο σε αυτό το σημείο είναι οι φόρμες αναζήτησης χρήστη, είτε μέσω «Κάρτας Μέλους», είτε μέσω ονόματος χρήστη που χρησιμοποιήθηκε κατά την εγγραφή. Θα μπορεί με αυτό τον τρόπο να εμφανίσει το προφίλ κάθε χρήστη και να προβάλλει ενδιαφέρουσες πληροφορίες, όπως το ιστορικό κρατήσεων και συναλλαγών του χρήστη. Επιπλέον θα υπάρχει δυνατότητα προβολής των εσόδων συνολικά αλλά και ημερησίως, γνωρίζοντας με αυτό τον τρόπο τον τζίρο που έχει κάνει το «Μέλος» ή στην περίπτωση του «Υπαλλήλου», να ελέγχει το ταμείο του.

Παράλληλα θα έχει στην διάθεση του ένα σύνολο οικονομικών στοιχείων που θα του επιτρέπουν την καλύτερη οργάνωση και διεύθυνση της επιχείρησης. Στον πίνακα που προαναφέρθηκε θα υπάρχει μία στήλη με τα συνολικά έσοδα από κάθε ταινία και έτσι θα μπορεί να θέτει στόχους και να επιβλέπει την πορεία της κάθε μίας. Συμπληρωματικά, θα μπορεί να επιβλέπει τα έσοδα οποιουδήποτε ημερολογιακού διαστήματος ελέγχοντας την πορεία της επιχείρησης, ενώ δεν θα λείπουν ενδιαφέροντα στατιστικά στοιχεία όπως έσοδα και ποσοστά ανά ηλικία, ημέρα της εβδομάδας και διαστήματα ωρών λειτουργίας,

επιτρέποντας έτσι την οργάνωση στρατηγικών μάρκετινγκ που θα βοηθήσουν την ανάπτυξης της επιχείρησης.

### 2.3.2 Ο Υπάλληλος

Το επίπεδο χρήστη του «Υπαλλήλου» είναι το ίδιο σημαντικό με το επίπεδο του «Διαχειριστή» διότι παρόλα τα λιγότερα δικαιώματα που έχει στην εφαρμογή, έχει διαφορετικές αρμοδιότητες. Η κύρια υποχρέωσή του είναι η διευκόλυνση και η γρήγορη εξυπηρέτηση του πελάτη και με αυτά ως γνώμονα θα αναλυθούν τα εργαλεία που θα έχει στην διάθεση του.

Αρχικά, το κύριο εργαλείο του «Υπαλλήλου» θα είναι το σύστημα κρατήσεων. Σε αυτό το σημείο γίνεται κατανοητό ότι κατά την διάρκεια μίας τηλεφωνικής επικοινωνίας ή φυσικής παρουσίας του πελάτη για κράτηση εισιτηρίων, κύριος στόχος είναι η γρήγορη εξυπηρέτηση του. Για αυτό το λόγο οι επιλογές πρέπει να σχεδιαστούν ώστε η διαδικασία της κράτησης να προχωράει βήμα – βήμα και εν τέλει να υπάρχει η δυνατότητα καταχώρησης του επιθυμητού ονόματος κράτησης.

Θα αποτελούσε παράληψη αν δεν υπήρχε η δυνατότητα διαγραφής μίας κράτησης. Επομένως, δίνεται η δυνατότητα, πριν από μία συγκεκριμένη ώρα που θα κρίνει η επιχείρηση, να εμφανίζονται οι συνολικές κρατήσεις μίας προβολής και να διαγράφονται αυτές που δεν έχουν εξοφληθεί, όπως και να ακυρωθούν εξοφλημένες κρατήσεις όπου οι χρήστες φρόντισαν να ενημερώσουν έγκαιρα και να ζητήσουν ακύρωσή της.

Η εξυπηρέτηση του πελάτη στο κατάστημα θα γίνεται μέσω εργαλείων που θα επιτρέπουν την αναζήτηση μίας κράτησης ονομαστικά, χωρίς να είναι απαραίτητο να αναζητήσουν ακριβώς το ονοματεπώνυμο με δυνατότητα επίσης εξόφλησης και ακύρωσης. Στην περίπτωση που ο πελάτης είναι «Μέλος» και έχει κάνει μία κράτηση διαδικτυακά, επιδεικνύοντας την «Κάρτα Μέλους» του, ο «Υπάλληλος» θα μπορεί να ελέγχει την κατάσταση της κράτησης και να εκδίδει, να εξοφλά ή να ακυρώνει κατ' επιθυμία του πελάτη.

Επίσης, θα έχει πρόσβαση στο προφίλ των «Μελών», αλλά μόνο στις πληροφορίες που είναι αναγκαίο να έχει πρόσβαση. Έτσι διαχωρίζεται από το επίπεδο του «Διαχειριστή» αποκλείοντας, ακόμη, την πρόσβαση στην προβολή εσόδων, προσθήκης και επισκόπησης ταινιών και προβολών.

### 2.3.3 Το Μέλος

Κάθε χρήστης της εφαρμογής θα έχει την δυνατότητα, δίνοντας τα στοιχεία του, να γίνεται «Μέλος». Από το κατάστημα θα εκδίδεται μία «Κάρτα Μέλους» που θα πρέπει να επιδεικνύεται όταν θέλει να επωφεληθεί από διαδικτυακές κρατήσεις. Μετά την έκδοση θα υπάρχει επιλογή που θα προσθέτει πόντους (credits) στην κάρτα, με αξία κάθε πόντου στην κρίση του καταστήματος.

Γνωρίζοντας την ποσότητα της πληροφορίας που θα δέχεται, απαιτείται να είναι και σωστά οργανωμένη. Όσον αφορά το σύστημα κράτησης θα μπορεί να ακολουθήσει την ίδια διαδικασία με τον «Υπάλληλο», επιλέγοντας τίτλο ταινίας, ημερομηνία και ώρα αρχίζοντας, είτε ανά τίτλο, είτε ανά ημερομηνία. Έτσι θα έχει την δυνατότητα να επιλέξει όποιες θέσεις επιθυμεί χωρίς όμως να εμφανίζεται το πεδίο που καταχωρεί το όνομα κράτησης ο «Υπάλληλος», διαχωρίζοντας έτσι τα δύο επίπεδα. Σε αυτό το σημείο θα έχει

την επιλογή, είτε να κάνει αγορά ξοδεύοντας πόντους, είτε απλή κράτηση χωρίς να υποχρεώνεται να εξοφλήσει ηλεκτρονικά.

Στην διάθεσή του είναι επίσης η προβολή του προφίλ του με το ιστορικό των κρατήσεων και με ένδειξη ποιες έχουν πληρωθεί και ποιες εκκρεμούν καθώς και το ιστορικό των συναλλαγών του αγοράς πόντων χάριν επιβεβαίωσης και ελέγχου. Επίσης θα υπάρχει επιλογή να εμφανίσει τις κρατήσεις των οποίων εκκρεμεί η πληρωμή και είτε να τις ακυρώσει, είτε να τις εξοφλήσει.

Τέλος, θα υπάρχει η δυνατότητα αλλαγής προσωπικού κωδικού του λογαριασμού του. Σε αυτό το επίπεδο δεν θα εμφανίζονται τα εισοδήματα διαχωρίζοντας το από τον «Διαχειριστή», αλλά και τα άλλα επίπεδα δεν θα έχουν πρόσβαση στην αλλαγή κωδικού του «Μέλους».

### **2.3.4 Ο Επισκέπτης**

Παρόλο που ο επισκέπτης της ιστοσελίδας δεν θεωρείται επίπεδο αξίζει να αναφέρουμε ότι θα έχει πρόσβαση στην ιστοσελίδα στα ελεύθερα τμήματά της. Πιο συγκεκριμένα θα μπορεί να γίνει μέλος και κατά την πρώτη του επίσκεψη να παραλαμβάνει την κάρτα μέλους, ωστόσο θα έχει πρόσβαση στις προβολές και τα βασικά στοιχεία των ταινιών, αλλά και στις ταινίες που αναμένεται να προβληθούν. Θα μπορεί ακόμη να προβάλλει τα τηλέφωνα, σε περίπτωση που θέλει να επικοινωνήσει με την επιχείρηση για πληροφορίες ή κράτηση και προσθέτοντας την διεύθυνση του να λαμβάνει οδηγίες για να βρει τον πολύ-κινηματογράφο.

## **2.4 Από την ανάλυση στον σχεδιασμό**

Σε αυτό το σημείο της διαδικασίας εκπόνησης της διαδικτυακής εφαρμογής, ακολουθήθηκαν μία σειρά από βήματα ώστε να περιοριστεί ο φόρτος από την έκταση των δυνατοτήτων της ιστοσελίδας. Αφετηρία για την επίτευξη αυτού ήταν η διάσπαση κάθε στοιχείου και δυνατότητας σε ένα υποπρόβλημα, βελτιώνοντας έτσι τον τρόπο σκέψης και δημιουργώντας στοιχεία που θα ήταν σε θέση να επαναχρησιμοποιηθούν σε διαφορετικά σημεία της εφαρμογής. Σημαντικό ρόλο κατέβαλε επίσης η σειρά των στόχων που τέθηκαν παραπάνω και αποτέλεσε ένα πλάνο εργασιών που βοήθησαν στην ανάπτυξη του κώδικα και την επίλυση προβλημάτων.

Ξεκινώντας με το γραφικό περιβάλλον και κατανοώντας την πληθώρα μεταβλητών στην εκτίμηση της διάθεσης και προτιμήσεων του επισκέπτη, σχεδιάστηκε το κυρίως σώμα της ιστοσελίδας. Οι απαιτήσεις καλύφθηκαν διατηρώντας την πλοήγηση απλή και την εμφάνιση του περιεχόμενου σε σημεία που δεν θα δυσκόλευαν τον χρήστη και δεν θα του προέβαλλαν πληροφορίες που δεν χρειαζόταν. Κατά το σχεδιασμό αυτό έπρεπε να επιλυθεί επίσης το πρόβλημα της πλοήγησης των χρηστών που θα είχαν πρόσβαση σε διαφορετικό περιεχόμενο από τον απλό επισκέπτη και ενός σημείου που θα εμφανιζόταν η φόρμα εισόδου. Το αποτέλεσμα ήταν η δημιουργία μίας βάσης και η πρώτη απεικόνιση για τον προορισμό εμφάνισης κάθε πληροφορίας. Στα κύρια μέρη της, ξεκινώντας από το πάνω μέρος, σχεδιάστηκε ένας χώρος για το λογότυπο της επιχείρησης και ένα οριζόντιο μενού πλοήγησης με τα βασικά μέρη της εφαρμογής με στοίχιση στο κέντρο και ελαφρώς μικρότερο πλάτος. Ο κεντρικός χώρος, στο ίδιο πλάτος με τον χώρο της πλοήγησης, χωρίστηκε σε δύο μέρη. Στα αριστερά, καταλαμβάνοντας τον περισσότερο χώρο θα εμφανίζεται το περιεχόμενο, ενώ στα δεξιά θα εμφανίζεται μία μικρότερη στήλη με την φόρμα εισόδου και το μενού των χρηστών που έχουν πραγματοποιήσει είσοδο με τον

χώρο από κάτω να προορίζεται για προωθητικές πληροφορίες της επιχείρησης. Στο κάτω μέρος της ιστοσελίδας σχεδιάστηκε ένα υποσέλιδο καλύπτοντας ολόκληρο το διαθέσιμο πλάτος που προορίζεται για συνοπτικές πληροφορίες της εφαρμογής και της επιχείρησης.

Εν συνεχεία, έχοντας τα θεμέλια και την οργάνωση των πληροφοριών και το μέρος που θα εμφανίζονται, επόμενη προτεραιότητα ήταν η διαχείριση. Αρχικά αναλύθηκε περαιτέρω το πρόβλημα του πολύ-κινηματογράφου και έγινε μία προσπάθεια να βρεθούν οι προς οργάνωση πληροφορίες. Έπειτα χωρίστηκαν σε κατηγορίες που μετατράπηκαν σε πίνακες και στοιχεία που τοποθετήθηκαν σε στήλες. Έτσι σαν πρώτο πλάνο δημιουργήθηκαν οι πίνακες των χρηστών, των ταινιών, των κρατήσεων, των αιθουσών και των προβολών, ενώ στην πορεία της ανάπτυξης προέκυψε και η ανάγκη ύπαρξης ενός πίνακα όπου θα αποθηκεύονταν όλες οι συναλλαγές τις επιχείρησης ανεξαρτήτως προέλευσης, είτε διαδικτύου, είτε φυσικών συναλλαγών.

Σε αυτό το σημείο, δεδομένου ενός συστήματος εισόδου και διαχωρισμού επιπέδου χρηστών, σχεδιάστηκαν πρώτα τα εργαλεία του «Διαχειριστή» λόγω του γεγονότος ότι είναι ο υπεύθυνος για την προσθήκη και συντήρηση της πληροφορίας. Όπως και κατά την διάρκεια της ανάλυσης της δομής της εφαρμογής, θα περιγραφεί η διαδικασία σχεδίασης των εργαλείων κάθε επιπέδου χρήστη με ιεραρχική δομή, περιγράφοντας σε κάθε επίπεδο τα εργαλεία που αποτελούν αρμοδιότητα ή κύριο ενδιαφέρον για το καθένα. Αυτό συνεπάγεται πως στα εργαλεία των κατώτερων επιπέδων θα έχουν πρόσβαση όλα τα ανώτερα.

#### **2.4.1 Ο Διαχειριστής**

Στο δεξιό τμήμα της σελίδας, όπως προαναφέρθηκε, θα πραγματοποιεί είσοδο και τότε θα εμφανίζεται το μενού πλοήγησής του. Μέσω αυτού θα έχει πρόσβαση σε όλα τα εργαλεία που του παρέχονται τα οποία θα αφορούν την διαχείριση της ιστοσελίδας και της επιχείρησης του.

Για λόγους διευκόλυνσης, το μενού θα χωριστεί ανάλογα με την ενέργεια που εκτελείται, στοχεύοντας έτσι στην ευκολότερη πλοήγηση. Ξεκινώντας με την πρόσθεση πληροφορίας, σχεδιάστηκε μία φόρμα για την προσθήκη ταινιών που θα του δίνει την δυνατότητα να βρίσκει αυτόματα της πληροφορίες και την αφίσα της ταινίας που θέλει να προσθέσει και στην συνέχεια, αφού επεξεργαστεί τα στοιχεία της ταινίας και αφού επιλέξει την αρχική κατάσταση της ταινίας, δηλαδή αν θα προστεθεί στο αρχείο, στις ταινίες που προβάλλονται ή θα είναι διαθέσιμες προσεχώς, να την αποθηκεύσει. Απαραίτητο εργαλείο ήταν και η προσθήκη προβολών στο οποίο θα εμφανίζονται, σε μορφή φόρμας, οι ταινίες που είναι καταχωρημένες ότι προβάλλονται, η αίθουσα και η επιθυμητή ημερομηνία και ώρα. Επίσης, θα εμφανίζεται μία φόρμα που θα μπορεί να προσθέσει μία αίθουσα καταχωρώντας την χωρητικότητα της.

Το στοιχείο που επιλέχθηκε να ακολουθήσει την πρόσθεση πληροφορίας ήταν η επισκόπηση όλων των πληροφοριών που έχει προσθέσει. Αυτό το σημείο της εφαρμογής είναι πολύ σημαντικό καθώς θα επιτρέπει την επίβλεψη της συνολικής ροής πληροφοριών προς τα «Μέλη». Συγκεντρώνοντας έναν τόσο μεγάλο όγκο πληροφορίας το σημείο αυτό απαιτούσε ιδιαίτερη προσοχή ώστε η προβολή του υλικού να μην δυσκολέψει την διαχείρισή του. Ο αρχικός διαχωρισμός ήταν προφανές ότι έπρεπε να χωριστεί σε στοιχεία που αφορούν τις ταινίες και στοιχεία που αφορούν τις προβολές. Στο πάνω μέρος τοποθετήθηκαν όλες οι ταινίες που έχουν προστεθεί στην Βάση Δεδομένων σε μορφή πίνακα, εμφανίζοντας βασικά χαρακτηριστικά όπως τίτλος, διάρκεια, είδος,

κατάσταση και τα συνολικά έσοδα από την κάθε μία. Τα συγκεκριμένα στοιχεία επιλέχθηκαν ως παράγοντες που θα έκριναν την πορεία μίας ταινίας, παραδείγματος χάρη αν έπρεπε να σταματήσει η προβολή της ή να προστεθούν περισσότερες. Στο πάνω μέρος αυτού του πίνακα εμφανίζονται φίλτρα, δηλαδή η δυνατότητα εμφάνισης, όλων των ταινιών, αυτών που είναι στο αρχείο, που προβάλλονται ή θα προβληθούν προσεχώς καθώς και το πλήθος σε κάθε κατηγορία. Κάτω από το πίνακα των ταινιών, σχεδιάστηκε το εργαλείο επίβλεψης των προβολών. Πρόκειται για καρτέλες σε ημερολογιακή σειρά που θα εμφανίζουν την ημερομηνία, την ημέρα της εβδομάδας και το σύνολο των προβολών που περιέχουν. Επιλέγοντας ο «Διαχειριστής» μία ημερομηνία, θα επιβλέπει τις προβολές της ημέρας εμφανίζοντας τον τίτλο, την αίθουσα, την ώρα και την πληρότητα της προβολής. Στα δεξιά και των δύο πινάκων και σε κάθε ταινία και προβολή ξεχωριστά, θα υπάρχουν δύο εικονίδια για άμεση διαγραφή και επεξεργασία, ενώ επιλέγοντας επεξεργασία θα εμφανίζεται μία αναδυόμενη φόρμα.

Οι αίθουσες σχεδιάστηκαν σε ξεχωριστή ενότητα λόγω της μικρής πιθανότητας για επεξεργασία ή προσθήκης, εκτός από την αρχική λειτουργία της εφαρμογής. Ωστόσο, σχεδιάστηκε πίνακας όπου θα εμφανίζονται όλες οι αίθουσες με την χωρητικότητα τους και εικονίδιο άμεσης διαγραφής, ενώ κάτω από αυτόν σχεδιάστηκαν φόρμες επεξεργασίας και προσθήκη νέας.

Μετά την σχεδίαση όλων των εργαλείων προσθήκης και διαχείρισης των στοιχείων που αφορούν τον χρήστη, ακολουθεί η σχεδίαση της ενότητας των στατιστικών. Σε αυτό το μέρος της εφαρμογής, όπως διατυπώθηκε και κατά την ανάλυση, πρέπει να συγκεντρωθεί όλο το υλικό που βρίσκεται στην Βάση Δεδομένων και να οργανωθεί με τέτοιο τρόπο ώστε να αποτελέσει μέσο οργάνωσης της επιχείρησης και εργαλείο σχεδίασης στρατηγικών μάρκετινγκ. Σχεδιάστηκε ένα σύστημα καρτελών που θα εμφανίζει τα έσοδα της επιχείρησης σε μορφή ποσού και σε μορφή ποσοστών ανάλογα με την κατηγορία. Στην πρώτη καρτέλα θα εμφανίζονται τα έσοδα ανά ημερομηνία, μέσω μίας φόρμας που θα δέχεται δύο ημερομηνίες που θα διαμορφώνουν ένα χρονικό διάστημα και πατώντας υποβολή θα εμφανίζεται ο συνολικός τζίρος γι αυτό το διάστημα. Η επόμενη καρτέλα στην σειρά αφορά την ηλικία των μελών όπου θα υπάρχει ένας πίνακας με διάφορα διαστήματα ηλικιών, ενώ κάτω από κάθε διάστημα θα εμφανίζεται το πλήθος των εγγεγραμμένων μελών, το ποσοστό και ο συνολικός τζίρος που ανήκει σε κάθε κατηγορία. Κατόπιν, οι επόμενες δύο καρτέλες που σχεδιάστηκαν αφορούν τις προβολές ανά ημέρα της εβδομάδας και ανά διαστήματα ωρών και στις δύο καρτέλες ξεχωριστά θα εμφανίζεται το πλήθος των προβολών, το ποσοστό και το σύνολο των εσόδων ανά κατηγορία.

Το τελευταίο σημείο της πλοήγησης ανήκει στον έλεγχο των χρηστών της ιστοσελίδας. Θα υπάρχουν δύο φόρμες για την αναζήτηση, είτε μέσω του μοναδικού αναγνωριστικού που αντιστοιχεί, είτε μέσω του ονόματος χρήστη. Βρίσκοντας τον χρήστη που επιθυμεί, θα εμφανίζει το προφίλ του μέσω του οποίου θα μπορεί να πλοηγηθεί με καρτέλες. Οι σημαντικότερες καρτέλες για τον διαχειριστή είναι η αλλαγή δικαιωμάτων που προάγει ένα απλό μέλος της εφαρμογής σε υπάλληλο, τα έσοδα από αυτόν που θα εμφανίζει έναν πίνακα με την πρώτη γραμμή να είναι το σύνολο και να ακολουθούν οι μη μηδενικές ημερομηνίες σε φθίνουσα σειρά και τέλος τα στοιχεία του. Το σημείο των εσόδων είναι πολύ σημαντικό γιατί στην περίπτωση προβολής προφίλ υπαλλήλου μπορεί να ελεγχθεί το ημερήσιο ταμείο του.

### 2.4.2 Ο Υπάλληλος

Κατά την σχεδίαση των εργαλείων του υπαλλήλου η κύρια πρόκληση ήταν η διαδικασία που θα ακολουθηθεί να γίνει με τέτοιο τρόπο ώστε η εξυπηρέτηση να είναι όσο το δυνατόν γρηγορότερη. Έτσι, τοποθετήθηκαν σε στρατηγικές θέσεις και συγχρόνως οργανώθηκαν ώστε οι ενέργειες να γίνονται βήμα – βήμα. Με αυτό τον τρόπο ο «Υπάλληλος» θα μπορεί να ακολουθεί την επιθυμία του πελάτη, μετατρέποντας κάθε πληροφορία σε αποτέλεσμα έως την τελική παροχή των υπηρεσιών που προσφέρει ένας πολύ-κινηματογράφος. Αφού πραγματοποιήσει σύνδεση με το μοναδικό όνομα χρήστη και τον κωδικό του, θα έχει πρόσβαση από το μενού πλοήγησης σε μία σειρά από εργαλεία που θα βοηθούν στην εξυπηρέτηση των πελατών.

Ξεκινώντας από το σύστημα κρατήσεων, που θα βρίσκεται στο κεντρικό μενού πλοήγησης της σελίδας, θα έρχεται σε επαφή με δύο φόρμες. Στο πάνω μέρος, θα μπορεί να διαλέξει τίτλο ταινίας και στην συνέχεια θα επιλέγει ημερομηνία και ώρα. Στο κάτω μέρος, θα μπορεί να επιλέξει με ημερολογιακή σειρά, ημερομηνίες που διαθέτουν προβολές και στην συνέχεια θα εμφανίζονται οι μικρογραφίες των ταινιών, ο τίτλος, η ώρα της προβολής και η αίθουσα. Αφότου γίνει η επιλογή ταινίας, ημερομηνίας και ώρας, ανεξαρτήτου από το ποια φόρμα επιλέχθηκε θα εμφανίζεται μία εικονική αίθουσα. Σε αυτό το σημείο τα χαρακτηριστικά της αίθουσας θα είναι ευδιάκριτα εμφανίζοντας την τοποθεσία της οθόνης και των διαδρόμων καθώς και τις μη διαθέσιμες θέσεις. Θα χρησιμοποιηθούν εικονίδια καθισμάτων με χρώμα ανοιχτό γκρι για τις διαθέσιμες θέσεις, κόκκινο για αυτές που υπάρχει ενεργής κράτηση και πράσινο για τις επιλεγμένες. Αφού επιλεγούν οι επιθυμητές θέσεις, θα εμφανίζεται μία φόρμα επιβεβαίωσης με τα στοιχεία της προβολής και ένα πεδίο σημειώσεων που σε περίπτωση κράτησης, ο υπάλληλος θα μπορεί να αποθηκεύσει το όνομα και το επώνυμο του πελάτη. Παράλληλα, από την ίδια φόρμα θα μπορεί να προβεί σε έκδοση εισιτηρίου επιλέγοντας αγορά.

Ήταν πολύ σημαντικό να επιτρέψουμε στην επιχείρηση, σε εύλογο χρονικό διάστημα πριν από μία προβολή, με αρμοδιότητα του «Υπαλλήλου», να ακυρώνονται οι κρατήσεις που δεν έχουν εξοφληθεί. Για αυτό το λόγο ο υπάλληλος θα έχει πρόσβαση στο μενού «Επισκόπηση» του διαχειριστή, με δικαιώματα προβολής όμως μόνο της επίβλεψης των προβολών, χωρίς δικαίωμα επεξεργασίας και διαγραφής. Θα έχει στην διάθεση του με αύξουσα σειρά ημερομηνίες που θα περιέχουν τις προβολές και στον πίνακα κάθε προβολής θα υπάρχει πεδίο που θα ενημερώνεται αυτόματα για την πληρότητα των θέσεων κάθε μίας. Πατώντας πάνω θα ανοίγει μία καινούρια σελίδα όπου θα εμφανίζονται όλες οι κρατηθείσες θέσεις με φίλτρα για προβολή όλων, των πληρωμένων ή που εκκρεμεί η πληρωμή τους. Σε κάθε κατηγορία θα υπάρχει ή δυνατότητα επιλογής, είτε μία – μία, είτε επιλογή όλων και ακύρωση.

Στην συνέχεια, σχεδιάστηκε το σύστημα εξυπηρέτησης πελάτη που αφορά τις κρατήσεις. Στην περίπτωση που ο πελάτης είναι «Μέλος», ο «Υπάλληλος» έχει πρόσβαση στην αναζήτηση χρηστών που περιγράφηκε κατά την σχεδίαση του «Διαχειριστή». Μέσω της αναζήτησης, είτε της κάρτας μέλους, είτε του ονόματος χρήστη, θα μπορεί να προβάλει το προφίλ του όπου με ένα σύστημα καρτελών θα μπορεί να επιλέξει την προβολή του ιστορικού των κρατήσεων, με φθίνουσα ημερολογιακή σειρά και το ιστορικό των συναλλαγών του για πληροφόρηση και επίλυση οποιουδήποτε προβλήματος. Έπειτα θα εμφανίζονται οι ενεργές κρατήσεις που δεν έχουν εξοφληθεί ώστε να μπορεί να τις εκδώσει ή να τις ακυρώσει. Τέλος, θα μπορεί να προβάλει τα στοιχεία του «Μέλους» μαζί με τον αριθμό των καρτών που έχουν εκδοθεί, για λόγους

ταυτοποίησης, και θα είναι σε θέση να εκδίδει την κάρτα μέλους. Πατώντας στην έκδοση της κάρτας θα ανοίγει νέο παράθυρο που θα εμφανίζει την κάρτα για να ενημερώνει ειδικό πρόγραμμα εκτύπωσης.

Στην επόμενη ενότητα του μενού πλοήγησης που αφορά την περίπτωση που ο πελάτης δεν είναι «Μέλος» αλλά έχει κράτηση, σχεδιάστηκε μία φόρμα αναζήτησης του ονόματος κράτησης που θα εμφανίζει τις ενεργές κρατήσεις οι οποίες αντιστοιχούν στις κρατήσεις που ταίριαζαν με το πεδίο κειμένου. Στην συνέχεια θα εμφανίζονται μικρές περιοχές πολλαπλής επιλογής, με ορισμένα στοιχεία και θα δίνεται η δυνατότητα πληρωμής ή ακύρωσής τους. Επίσης θα είναι διαθέσιμη η δυνατότητα ελέγχου εγκυρότητας ενός εισιτηρίου που θα εμφανίζει κατάλληλο μήνυμα όταν είναι άκυρο ή όταν είναι έγκυρο αν έχει εξοφληθεί η αξία του.

### 2.4.3 Το Μέλος

Αφού ακολουθήσει την διαδικασία εγγραφής και συνδεθεί το «Μέλος», θα αποκτή πρόσβαση αρχικά στο σύστημα κράτησης όπως περιγράφηκε προηγουμένως. Αφού γίνει επιλογή ημερομηνίας, τίτλου και ώρας με οποιαδήποτε από τα δύο εργαλεία θα μπορεί όπως και ο «Υπάλληλος» να διαλέξει τις θέσεις που επιθυμεί με κάποιους περιορισμούς. Δεν επιτρέπεται να επιλέξει πάνω από πέντε θέσεις και στην φόρμα επιβεβαίωσης δεν θα είναι σε θέση να εμφανίσει το πεδίο που ο «Υπάλληλος» πρόσθετε το όνομα ωστόσο μπορεί να εκτελέσει κανονικά κράτηση ή αγορά.

Στο δεξί μέρος της εφαρμογής θα εμφανίζονται ο αριθμός των «Πόντων» που διαθέτει στον λογαριασμό του και το μενού πλοήγησης του. Πατώντας πάνω στον αριθμό των πόντων θα ανακατευθύνεται σε μία φόρμα που μέσω της πιστωτικής ή χρεωστικής του κάρτας θα μπορεί να αγοράσει πόντους. Για λόγους ασφαλείας δεν θα μπορεί κάθε φορά να επιλέξει την αγορά άνω των πέντε πόντων.<sup>2</sup>

Από το μενού πλοήγησης θα έχει πρόσβαση στο προφίλ του το οποίο σχεδιάστηκε με σκοπό ενημερωτικό αλλά και λειτουργικό. Αρχικά θα έρχεται σε επαφή με ένα σύστημα καρτελών για την εσωτερική πλοήγηση στο προφίλ. Όσον αφορά το πληροφοριακό μέρος του, θα έχει πρόσβαση στο ιστορικό των κρατήσεων και των χρηματικών του συναλλαγών με το σύστημα των πόντων. Το ιστορικό κρατήσεων σχεδιάστηκε με σκοπό να ενημερώνει το «Μέλος» για την κατάσταση των κρατήσεων, αν είναι δηλαδή εξοφλημένες ή όχι. Θα εμφανίζονται οι ημερομηνίες που έχει γίνει κράτηση ξεκινώντας από την πιο πρόσφατη. Αφού γίνει η επιλογή ημερομηνίας, θα εμφανίζονται οι κρατήσεις με κάποια στοιχεία της ταινίας και της προβολής όπως μικρογραφία της αφίσας, τίτλος, ώρα, αίθουσα και διάρκεια. Στα δεξιά αυτού του πίνακα θα εμφανίζεται ένδειξη της κατάστασης στην οποία βρίσκεται η κράτηση. Στο ιστορικό των συναλλαγών σχεδιάστηκε ένας πίνακας με τις ημερομηνίες σε επίσης φθίνουσα σειρά που θα αναφέρει το ποσό, την ημερομηνία και την ώρα που έγινε συναλλαγή με το σύστημα αγοράς πόντων.

Όσον αφορά την διαχείριση του λογαριασμού του και των κρατήσεων που έχει κάνει, σχεδιάστηκαν οι καρτέλες Εξόφληση/Ακύρωση, όπου θα εμφανίζονται οι κρατήσεις με δυνατότητα πολλαπλής επιλογής και επιλογές για εξόφληση ή ακύρωση. Στην συνέχεια σχεδιάστηκε καρτέλα για αλλαγή κωδικού, όπου δημιουργήθηκε φόρμα που το «Μέλος» θα εισάγει τον τρέχοντα κωδικό του και τον επιθυμητό σε δύο πεδία για επιβεβαίωση για να πραγματοποιήσει αλλαγή κωδικού.

<sup>2</sup> | Σημ.: Για λόγους ασφαλείας και όρων χρήσης των υπηρεσιών της ΕΔΕΤ που αφορούν την χρήση των server για μη κερδοσκοπικούς σκοπούς καθώς και για αποφυγή αποθήκευσης ευαίσθητων τραπεζικών στοιχείων, η φόρμα δεν θα επαληθεύει τα στοιχεία που εισάγονται και θα προσθέτει τους πόντους χωρίς να αποθηκεύει κανένα στοιχείο.

#### 2.4.4 Ο επισκέπτης

Κατά την είσοδο στην εφαρμογή ο «Επισκέπτης» θα έρχεται σε επαφή με κάποιες σύντομες πληροφορίες για την επιχείρηση. Στην διάθεση του βρίσκεται το κεντρικό μενού πλοήγησης που διαθέτει την αρχική σελίδα, τις προβολές, τις κρατήσεις, την ενότητα προσεχώς και επικοινωνία.

Επιλέγοντας το μενού προβολές θα έρχεται σε επαφή με ξεχωριστές περιοχές των ταινιών που ο «Διαχειριστής» έχει επιλέξει ότι βρίσκονται σε προβολή. Σε κάθε περιοχή θα εμφανίζεται η αφίσα της ταινίας, ο τίτλος, το είδος, η διάρκεια, η υπόθεση και μία σύντομη αναφορά στις προβολές της. Σε κάθε περιοχή θα υπάρχει εικονίδιο του IMDB (Internet Movie Database) το οποίο θα ανοίγει σε νέα σελίδα την αντίστοιχη ταινία για περισσότερες πληροφορίες. Ομοίως σχεδιάστηκε και το μενού «Προσεχώς» χωρίς να εμφανίζει στοιχεία για προβολές, ενώ επιλέγοντας κράτηση ο «Επισκέπτης» θα προτρέπεται να συνδεθεί. Στο μενού «Επικοινωνία» θα εμφανίζονται τα τηλέφωνα της επιχείρησης για τηλεφωνική κράτηση ή πληροφορίες ενώ θα εμφανίζεται και φόρμα που θα του επιτρέπει, βάζοντας μία διεύθυνση, να λάβει σε νέο παράθυρο οδηγίες πλοήγησης στον πολύ-κινηματογράφο.

Στο δεξιό μέρος στην θέση του μενού πλοήγησης των εγγεγραμμένων χρηστών εμφανίζεται η φόρμα εισόδου και το μήνυμα που τον προτρέπει να γίνει μέλος. Ακολουθώντας τον σύνδεσμο ανακατευθύνεται σε μία φόρμα εγγραφής που θα πρέπει να εισάγει όνομα χρήστη, κωδικό, όνομα, επώνυμο, ημερομηνία γέννησης και διεύθυνση ηλεκτρονικής επικοινωνίας. Πραγματοποιώντας εγγραφή κατά την πρώτη του κράτηση ή αγορά εισιτηρίου, θα πρέπει να το αναφέρει στον «Υπάλληλο» ώστε να του εκδώσει την κάρτα μέλους.



## ΚΕΦΑΛΑΙΟ 3 ΕΠΙΛΟΓΗ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

Σε αυτό το κεφάλαιο θα γίνει η περιγραφή της διαδικασίας που ακολουθήθηκε για την επιλογή, την εγκατάσταση και την λειτουργία των εφαρμογών που χρειάζονται για να σχεδιαστεί και να λειτουργήσει μία διαδικτυακή εφαρμογή όπως η ιστοσελίδα του πολύ-κινηματογράφου.

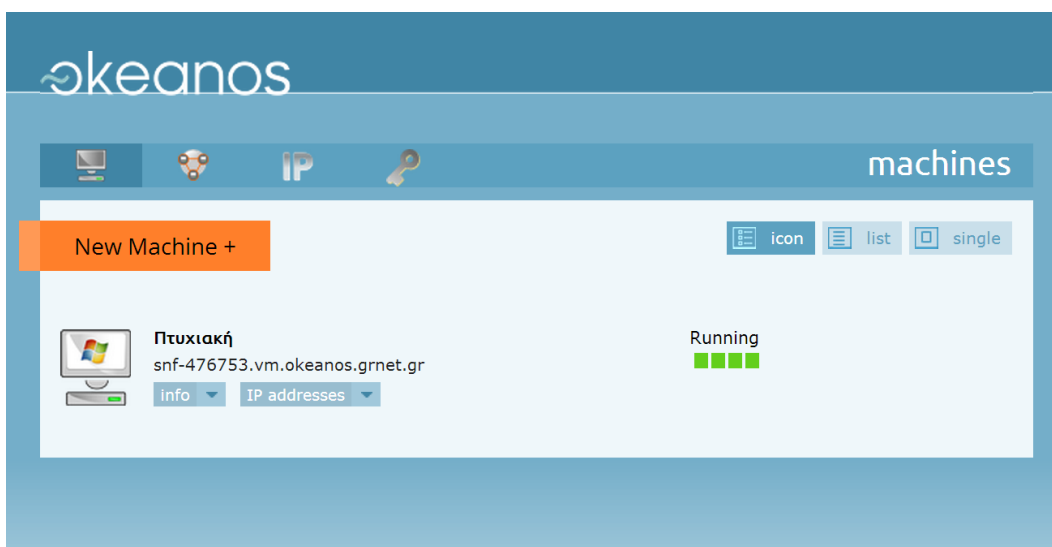
### 3.1 Ο εξυπηρετητής

Η ιστοσελίδα μίας επιχείρησης απαιτείται να είναι αναρτημένη συνεχώς στο διαδίκτυο και συγχρόνως να διαθέτει μεγάλη επεξεργαστική ισχύ για να ανταπεξέλθει στην επισκεψιμότητα των χρηστών. Αυτές οι ανάγκες καλύπτονται από έναν εξυπηρετητή. Το Εθνικό Δίκτυο Έρευνας και Τεχνολογίας (ΕΔΕΤ) παρέχει στην ελληνική ακαδημαϊκή, ερευνητική και εκπαιδευτική κοινότητα προηγμένες υπηρεσίες εθνικής διασύνδεσης υψηλών ταχυτήτων, εξυπηρετώντας όλα τα Πανεπιστήμια, ΤΕΙ, τα ερευνητικά κέντρα της χώρας και τα σχολεία, μέσω του Πανελληνίου Σχολικού Δικτύου. Μία από αυτές τις υπηρεσίες είναι η ~Okeanos IaaS Cloud που προσφέρει εργαλεία υπολογιστικού νέφους Cyclades και Pithos και μπορεί να αποκτήσει κάποιος πρόσβαση από την ιστοσελίδα <https://cyclades.okeanos.grnet.gr>.



Εικόνα 3.1-1, Λογότυπο ~Okeanos

Το εργαλείο που θα βοηθήσει στις παραπάνω απαιτήσεις της διαδικτυακής εφαρμογής είναι το Cyclades. Πρόκειται για μία υπηρεσία παροχής εικονικών υπολογιστών και δικτύων. Πραγματοποιώντας είσοδο με τα ακαδημαϊκά στοιχεία αποκτήθηκε πρόσβαση στην δημιουργία εικονικής μηχανής και ως εκ τούτου στην απομακρυσμένη επιφάνεια εργασίας.



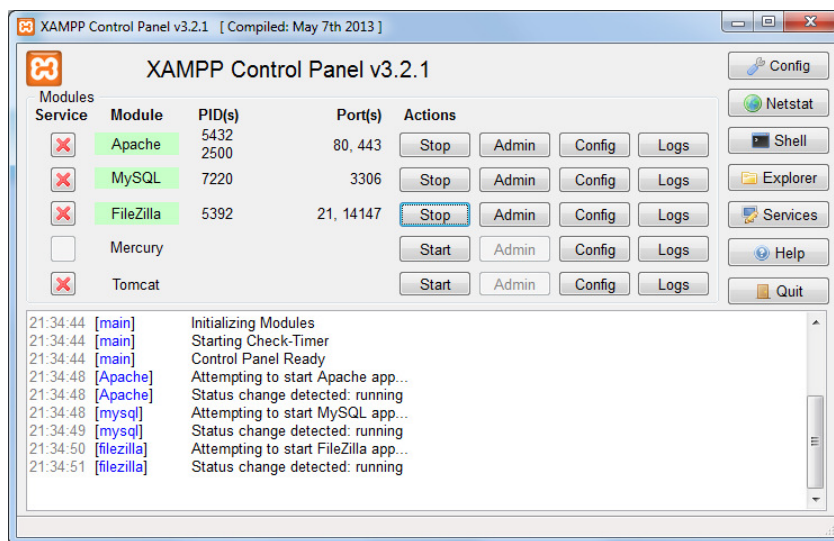
Εικόνα 3.1-2, Πίνακας Ελέγχου Εικονικών Μηχανών

Αφού εξασφαλίστηκε η ανάγκη συνεχούς ομαλής λειτουργίας και συνεχούς σύνδεσης στο διαδίκτυο ενός εξυπηρετητή, χρειάστηκε ένα πρόγραμμα διακομιστή που θα αναλάμβανε την κίνηση HTTP καθώς και ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων. Το πρόγραμμα που επιλέχθηκε είναι το XAMPP.



Εικόνα 3.1-3, Λογότυπο XAMPP

Το XAMPP είναι μία σουίτα λογισμικού που ανήκει στην κατηγορία Ελεύθερο Λογισμικό/Λογισμικό Ανοιχτού Κώδικα (ΕΛ/ΛΑΚ) και το οποίο περιέχει μεταξύ άλλων τον διακομιστή Apache, το Σύστημα Διαχείρισης Βάσης Δεδομένων MySQL, έναν μεταγλωτιστή για κώδικα γραμμένο σε PHP και έναν εξυπηρετητή κίνησης αρχείων ftp και εκτός αυτών εγκαθίσταται σε οποιοδήποτε λειτουργικό σύστημα. Ο καθένας μπορεί να αποκτήσει πρόσβαση στο πρόγραμμα ανακτώντας το αρχείο που αντιστοιχεί στο λειτουργικό του σύστημα από την ιστοσελίδα <https://www.apachefriends.org/index.html>. Αφού ολοκληρωθεί η εγκατάσταση ανοίγει ο πίνακας ελέγχου του προγράμματος μέσω του οποίου είναι διαθέσιμα όλα τα εργαλεία.

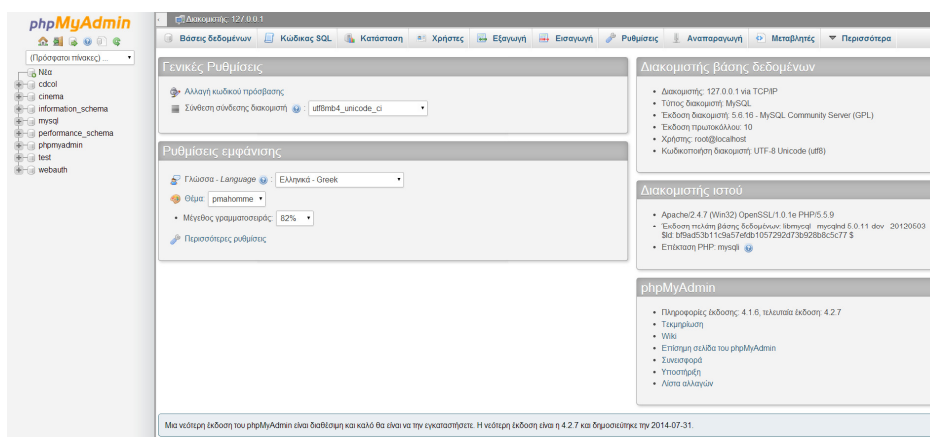


Εικόνα 3.1-4, Παράθυρο Πίνακα Ελέγχου XAMPP

Αναλυτικότερα, αρχικά ενεργοποιήθηκε ο Apache που είναι ένα πρόγραμμα εξυπηρετητή ο οποίος, όταν ένας χρήστης θα επισκέπτεται την ιστοσελίδα μέσω του πρωτοκόλλου HTTP, θα παράγει τις ιστοσελίδες και θα τις στέλνει στον φυλλομετρητή του. Στην συνέχεια, ενεργοποιήθηκε το FileZilla, είναι ένα πρόγραμμα εξυπηρετητή FTP (File Transfer Protocol) που βοηθά στην μεταφορά αρχείων από και προς τον εξυπηρετητή που δημιουργήθηκε και τέλος η MySQL που είναι ένα από τα σημαντικότερα εργαλεία που εξυπηρετούν στην μεταφορά δεδομένων. Πρόκειται για ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων το οποίο είναι συγχρόνως εξυπηρετητής που επιτρέπει την μεταφορά δεδομένων από και προς μία βάση δεδομένων.

Ύστερα από την εγκατάσταση και την ενεργοποίηση των απαραίτητων υπηρεσιών, στον φάκελο εγκατάστασης βρίσκεται ο φάκελος «htdocs». Αποθηκεύοντας αρχεία php, html, css επιτυγχάνεται η πρόσβαση σε αυτά από εξωτερικές συνδέσεις δίνοντας έτσι την δυνατότητα προβολής της εφαρμογής.

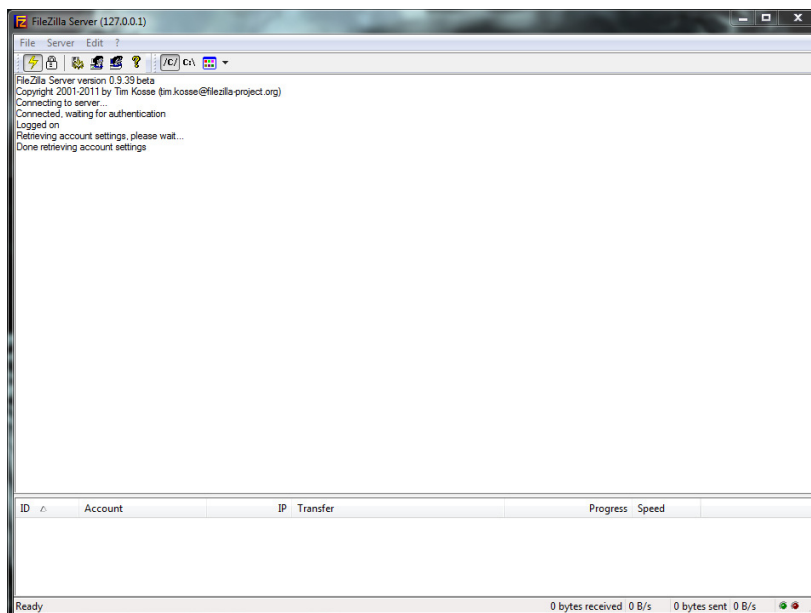
Ένα ακόμη εργαλείο που προσφέρει η σουίτα XAMPP είναι το phpmyadmin. Μεταβαίνοντας στην τοπική τοποθεσία <http://localhost/phpmyadmin> ερχόμαστε σε επαφή με ένα γραφικό περιβάλλον που σχεδιάστηκε με σκοπό την διαχείριση της MySQL με την χρήση του επιθυμητού προγράμματος περιήγησης.



Εικόνα 3.1-5, Γραφικό Περιβάλλον PhpMyAdmin

Με αυτό το εργαλείο μπορούμε να δημιουργήσουμε βάσεις δεδομένων και πίνακες με γραφικό τρόπο ενώ δεν λείπει η δυνατότητα παρακολούθησης των περιεχομένων και η αποθήκευση αντιγράφου ασφαλείας.

Για την μεταφορά αυτών των αρχείων, αφού ενεργοποιήθηκε η υπηρεσία του Filezilla εγκαταστάθηκε το πρόγραμμα Filezilla Server που είναι ελεύθερης πρόσβασης στην ιστοσελίδα <https://filezilla-project.org/>.



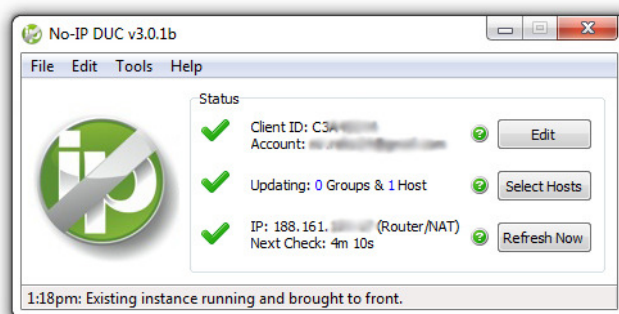
Εικόνα 3.1-6, Παράθυρο FileZilla Server

Μέσω αυτού του προγράμματος καθορίζεται η διαδρομή του δίσκου στην οποία θα δίνεται πρόσβαση μέσω ftp καθώς και το όνομα χρήστη και ο κωδικός πρόσβασης που θα πρέπει να εισαχθούν.

Τέλος, για την αλλαγή της διεύθυνσης από αριθμητικής μορφής σε συνηθισμένη μορφή διεύθυνσης ιστοσελίδας και την δημιουργία μίας στατικής διεύθυνσης για πρόσβαση στην εφαρμογή, πραγματοποιήθηκε εγγραφή στην NO-IP που είναι μία εταιρία παροχής στατικής διεύθυνσης και κατοχύρωσης ονόματος τομέα (domain name).



Εικόνα 3.1-7, Λογότυπο No-IP



Εικόνα 3.1-8, Παράθυρο Αυτόματης Ενημέρωσης

Το NO-IP επιτρέπει την δημιουργία μίας διεύθυνσης που θα αντιστοιχείται με την διεύθυνση IP του εξυπηρετητή, ενώ η διάθεση είναι ελεύθερη για συγκεκριμένες καταλήξεις της διεύθυνσης. Για την αποφυγή χειροκίνητης ενημέρωσης σε περίπτωση αλλαγής της διεύθυνσης, χρησιμοποιήθηκε ένα εργαλείο που παραχωρείται από την ανωτέρω εταιρεία που επιτρέπει την αυτόματη ενημέρωση (Εικόνα 3.1-8). Με αυτό τον τρόπο μας δίνεται η δυνατότητα, εισάγοντας τα διαπιστευτήρια που επιλέξαμε κατά την εγγραφή, να επιτρέψουμε στο πρόγραμμα ανά πέντε λεπτά τον αυτόματο έλεγχο της διεύθυνσης και την αυτόματη αντιστοίχιση με την τρέχουσα διεύθυνση IP. Η εγγραφή, η κατοχύρωση ονόματος τομέα και η ανάκτηση του εργαλείου για εγκατάσταση, μπορούν να γίνουν από την διεύθυνση <http://www.noip.com/>.

## 3.2 Οι Γλώσσες Προγραμματισμού

Για την λειτουργία μίας δυναμικής ιστοσελίδας απαιτείται η ανάπτυξη κώδικα σε μία σειρά από γλώσσες προγραμματισμού για τις λειτουργίες και την σχεδίαση. Αυτές οι γλώσσες είναι οι PHP, HTML, CSS, JavaScript και SQL.



Εικόνα 3.2-1, Λογότυπα Γλωσσών Προγραμματισμού

Η PHP είναι μία γλώσσα προγραμματισμού για την δημιουργία σελίδων με δυναμικό περιεχόμενο. Μεταβαίνοντας ο επισκέπτης σε μία σελίδα που έχει αναπτυχθεί με κώδικα PHP, ο εξυπηρετητής Apache τον χρησιμοποιεί για να παράγει σε μορφή κώδικα HTML το τελικό περιεχόμενο σε πραγματικό χρόνο και να τον αποστείλει στο

πρόγραμμα περιήγησής του. Με αυτόν τον τρόπο δεν είναι ποτέ δυνατή η εμφάνιση του κώδικα απομακρυσμένα αλλά μόνο η έξοδος του κώδικα.

Η HTML είναι το βασικό δομικό στοιχείο κάθε ιστοσελίδας και συντάσσεται υπό μορφή ετικετών. Ο σκοπός χρήσης της είναι κατά την λήψη του κώδικα από τον φυλλομετρητή να ερμηνεύεται και να παράγεται η δομή της ιστοσελίδας. Η σύνταξη γίνεται σε μορφή ετικετών και αφού σχεδιαστεί η δομή της, η HTML μας δίνει την δυνατότητα να προσθέσουμε σε κάθε ετικέτα κάποια χαρακτηριστικά. Τα δύο ίσως πιο σημαντικά χαρακτηριστικά χρησιμοποιούνται για την οργάνωση των ετικετών παραχωρώντας τους μία μοναδική ταυτότητα (id) για κάθε ετικέτα ή οργάνωσής τους σε κλάσεις (class) ετικετών. Με αυτόν τον τρόπο μπορούμε να επέμβουμε στην δομή της ιστοσελίδας ώστε να ελέγξουμε τον τρόπο εμφάνισης της και μορφοποίησής τους.

Συγκεκριμένα, με την χρήση της γλώσσας CSS (Cascading Style Sheets) ελέγχουμε την εμφάνιση του εγγράφου HTML. Με την χρήση της CSS 3 αποκτάται πρόσβαση σε μία σειρά από ελέγχους της μορφοποίησης και στον τρόπο ελέγχου εμφάνισης των ετικετών, όπως τον έλεγχο στην διαμόρφωση μεγεθών, διάταξης, χρωμάτων, στοίχισης, γραμματοσειρών, κίνησης και άλλων.

Η JavaScript, αντίθετα με την PHP, είναι μία γλώσσα προγραμματισμού σχεδιασμένη για χρήση στην πλευρά του προγράμματος περιήγησης. Χρησιμοποιείται για την αλληλεπίδραση του χρήστη με την ιστοσελίδα ανταλλάσσοντας δεδομένα ασύγχρονα και αλλάζοντας δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Ένα επίσης μεγάλο πλεονέκτημα χρήσης της JavaScript είναι η μείωση του φόρτου ανάγκης επεξεργαστικής ισχύς του εξυπηρετητή.

Τέλος, η SQL (Structured Query Language) είναι μία γλώσσα υπολογιστών που σχεδιάστηκε για χρήση σε ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων όπως η MySQL. Η χρήση της γλώσσας, επιτρέπει ανάκτηση και ενημέρωση δεδομένων, δημιουργία και τροποποίηση σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου του τρόπου πρόσβασης στα δεδομένα. Ενσωματώνοντάς την σε συναρτήσεις PHP θα επιτραπεί η αυτόματη κίνηση πληροφοριών από και προς την MySQL.

### 3.3 Ανάπτυξη του Κώδικα

Για την συγγραφή κώδικα στις παραπάνω γλώσσες απαιτείται η χρήση ενός κειμενογράφου και η αποθήκευση με την κατάλληλη κατάληξη. Η επιλογή του προγράμματος έγινε με κριτήρια την παροχή εργαλείων διευκόλυνσης κατά την συγγραφή κώδικα και την ύπαρξη δυνατότητας προσθήκης «πελάτη» ftp (client). Μετά από σύντομη έρευνα έγινε επιλογή του Notepad++.

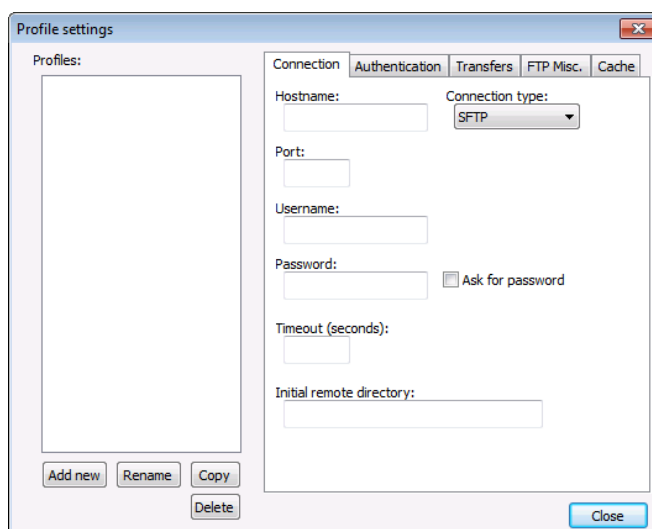


Εικόνα 3.3-1, Λογότυπο Notepad++

Το Notepad++ είναι ένας κειμενογράφος κυρίως για ανάπτυξη κώδικα και μπορεί να ανακτηθεί και να εγκατασταθεί από την ιστοσελίδα <http://notepad-plus-plus.org/>. Κατά

την συγγραφή κώδικα προσφέρει αρκετά εργαλεία διευκόλυνσης ενώ δεν λείπει η δυνατότητα προσθήκης επεκτάσεων. Τα σημαντικότερα εργαλεία που διευκολύνουν τον προγραμματιστή είναι η αρίθμηση των γραμμών κώδικα, η επεξεργασία πολλαπλών γραμμών συγχρόνως, αυτόματο κλείσιμο ετικετών όλων των παραπάνω γλωσσών προγραμματισμού καθώς και ετικετών σχολίων και σημεία στίξης που χρησιμοποιούνται για έναρξη και λήξη βρόγχων. Ακόμη, επιτρέπει την ταυτόχρονη επεξεργασία πολλαπλών αρχείων και η έντονη επισήμανση των λέξεων που ανήκουν στις βιβλιοθήκες των παραπάνω γλωσσών προγραμματισμού.

Ολοκληρώνοντας την εγκατάσταση και προσθέτοντας τις επιθυμητές ρυθμίσεις οργάνωσης και εμφάνισης των αρχείων και του Notepad++ ερχόμαστε σε επαφή με το γραφικό περιβάλλον του. Ανοίγοντας τον διαχειριστή προσθέτων είναι διαθέσιμη πληθώρα επεκτάσεων και αναζητώντας συνεχίζουμε με την εγκατάσταση των προσθέτων ελέγχου ορθογραφίας DSpellCheck και του πελάτη ftp NppFTP. Η χρησιμότητα του προσθέτου πελάτη ftp μας επιτρέπει την ρύθμιση της διεύθυνσης του εξυπηρετητή και την αποθήκευση των διαπιστευτηρίων που επιλέχθηκαν κατά την εγκατάσταση στον FileZilla server, ώστε να πραγματοποιηθεί σύνδεση.



**Εικόνα 3.3-2, Παράθυρο Ρυθμίσεων NppFTP**

Αφού πραγματοποιηθεί σύνδεση, ο συνδυασμός Notepad++ με NppFTP και FileZilla Server, μας επιτρέπει την ανάκτηση αρχείων από τον εξυπηρετητή και την δυνατότητα επεξεργασίας και την εκ νέου μεταφόρτωση. Μία πολύ σημαντική ιδιότητα αυτού του συνδυασμού είναι η αυτόματη μεταφόρτωση στον εξυπηρετητή κατά την αποθήκευση του εγγράφου χωρίς ειδική ρύθμιση. Τέλος, υπάρχει η δυνατότητα αποθήκευσης μίας διαδρομής στον τοπικό δίσκο για την φύλαξη αντιγράφου ασφαλείας.

## ΚΕΦΑΛΑΙΟ 4 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Σε αυτό το κεφάλαιο θα παρουσιαστεί η τελική μορφή της ιστοσελίδας και των εργαλείων που την καθιστούν μέσο κράτησης, αγοράς και διαχείρισης ενός πολύ-κινηματογράφου. Θα δοθεί σημασία στο οπτικό αποτέλεσμα και στις διευκολύνσεις που προσφέρει, ενώ η ανάλυση του κώδικα θα πραγματοποιηθεί στο επόμενο κεφάλαιο.

Η διαδικτυακή εφαρμογή «CinemaCied» πήρε το όνομα της από τον τύπο της επιχείρησης και τον αγγλικό τίτλο του τμήματος στο οποίο απευθύνεται. Η ιστοσελίδα είναι διαθέσιμη και πλήρως λειτουργική για το κοινό στο διαδίκτυο ακολουθώντας την διεύθυνση <http://cinemacied.hopto.org>.

### 4.1 Γενικά

Ξεκινώντας από την πρώτη επαφή με την εφαρμογή, παρουσιάζεται η αρχική σελίδα που εμφανίζει ένα μήνυμα υποδοχής, συνδέσμους για πλοήγηση στον κορμό της ιστοσελίδας, πρόσβαση στο σύστημα σύνδεσης χρηστών και ενημερωτικές προσφορές που αφορούν την επιχείρηση. Σε αυτή τη σελίδα εμφανίζεται επίσης και το πρότυπο που θα ακολουθήσουν οι υπόλοιπες, αφού κάθε περιεχόμενο θα ενσωματώνεται στο κέντρο και τα γύρω στοιχεία θα παραμένουν σταθερά. Ειδικότερα, το λογότυπο και οι σύνδεσμοι πλοήγησης παραμένουν στο πάνω μέρος της σελίδας, όπως επίσης η περιοχή εισόδου και προσφορών του καταστήματος στα αριστερά και το υποσέλιδο στο κάτω μέρος.



Εικόνα 4.1-1, /index.php – Αρχική Σελίδα

Στην συνέχεια του κεφαλαίου θα γίνει παρουσίαση κάθε σελίδας, ακολουθώντας τον χάρτη πλοήγησής της και αναλύοντας τις δυνατότητες κάθε εργαλείου. Επειδή η συγγραφή του κώδικα έγινε σε μορφή συναρτήσεων PHP για την σύμπτυξή του αλλά και

την επαναχρησιμοποίηση του παραγόμενου κώδικα HTML σε πολλά σημεία, τα περισσότερα εργαλεία είναι κοινά για όλα τα επίπεδα χρηστών με μικρές διαφοροποιήσεις. Προκειμένου να παρουσιαστούν αυτές οι διαφορές και ο λόγος ύπαρξης τους θα πραγματοποιηθούν οπτικές συγκρίσεις των εργαλείων και θα εξηγηθεί ποιος είναι ο σκοπός της διαφοροποίησης.

Για την ενημέρωση κάθε ενδιαφερόμενου αναλύθηκε, σχεδιάστηκε και υλοποιήθηκε η σελίδα «Προβολές». Πρόκειται για το σημείο όπου κάθε επισκέπτης θα μπορεί να ενημερώνεται για τις διαθέσιμες προς προβολή ταινίες. Κάθε ταινία εμφανίζεται στην δική της περιοχή και κάθε περιοχή χωρίζεται ανάλογα με την πληροφορία που προβάλλεται. Στο πάνω μέρος εμφανίζεται ο τίτλος της ταινίας, στα αριστερά η αφίσα της, και στα δεξιά πιο λεπτομερή χαρακτηριστικά της, την κατηγορία στην οποία εντάσσεται, την διάρκεια, την υπόθεση και συνοπτικά οι μέρες και ώρες που προβάλλεται. Αυτόματα σε κάθε ταινία αναπαράγεται στο πάνω δεξί μέρος της περιοχής της ένα εικονίδιο της δημοφιλούς Βάσης Δεδομένων Ταινιών (IMDB) όπου ο χρήστης μπορεί να αναζητήσει περισσότερες πληροφορίες και να διαβάσει κριτικές.

Ο χρήστης αφού εισέλθει σε αυτόν τον τομέα βλέπει μόνο τα στοιχεία της ταινίας, όπως φαίνεται στην Εικόνα 4.1-2



**The Maze Runner** 

**Κατηγορία :**  
Action, Mystery, Sci-Fi

**Διάρκεια :**  
153 min

**Υπόθεση :**  
Thomas wakes up in an elevator, remembering nothing but his own name. He emerges into a world of about 60 teen boys who have learned to survive in a completely enclosed environment, subsisting on their own agriculture and supplies. A new boy arrives every 30 days. The original group has been in "The Glade" for two years, trying to find a way to escape through the Maze that surrounds their living space. They have begun to give up hope. Then a comatose girl arrives with a strange note, and their world begins to change. There are some great, fast-paced action scenes, particularly those involving the nightmarish Grievers who plague the boys.

**Εμφάνιση Προβολών**


Εικόνα 4.1-2, Τρόπος Εμφάνισης Ταινίας σε Προβολή

Ενώ αφού επιλεγθεί η εμφάνιση προβολών για την συγκεκριμένη ταινία εμφανίζονται συνοπτικά οι προβολές χωρισμένες ανά ημερομηνία όπως στην εικόνα 4.1-3.





## The Maze Runner



**Κατηγορία :**  
Action, Mystery, Sci-Fi

**Διάρκεια :**  
153 min

**Υπόθεση :**

Thomas wakes up in an elevator, remembering nothing but his own name. He emerges into a world of about 60 teen boys who have learned to survive in a completely enclosed environment, subsisting on their own agriculture and supplies. A new boy arrives every 30 days. The original group has been in "The Glade" for two years, trying to find a way to escape through the Maze that surrounds their living space. They have begun to give up hope. Then a comatose girl arrives with a strange note, and their world begins to change. There are some great, fast-paced action scenes, particularly those involving the nightmarish Grievers who plague the boys.

**Εμφάνιση Προβολών**

- στις **01-09-2014** στις
- στις **19:00** στην Αίθουσα **3**
- στις **20:00** στην Αίθουσα **2**
- στις **02-09-2014** στις
- στις **19:30** στην Αίθουσα **3**
- στις **20:30** στην Αίθουσα **2**
- στις **03-09-2014** στις
- στις **19:00** στην Αίθουσα **3**
- στις **20:00** στην Αίθουσα **2**
- στις **20:00** στην Αίθουσα **2**

Εικόνα 4.1-3, Εμφάνιση Προβολών

Με παρόμοιο τρόπο οι χρήστες θα μπορούν να ενημερώνονται για τις ταινίες που θα είναι σύντομα διαθέσιμες στον πολύ-κινηματογράφο. Αυτό το σημείο ονομάστηκε «Προσεχώς» και η μόνη διαφορά στην εμφάνιση των ταινιών είναι η απουσία του στοιχείου που εμφανίζονται οι προβολές.



## Edge of Tomorrow



**Κατηγορία :**  
Action, Sci-Fi

**Διάρκεια :**  
113 min

**Υπόθεση :**

An alien race has hit the Earth in an unrelenting assault, unbeatable by any military unit in the world. Major William Cage (Cruise) is an officer who has never seen a day of combat when he is unceremoniously dropped into what amounts to a suicide mission. Killed within minutes, Cage now finds himself inexplicably thrown into a time loop-forcing him to live out the same brutal combat over and over, fighting and dying again...and again. But with each battle, Cage becomes able to engage the adversaries with increasing skill, alongside Special Forces warrior Rita Vrataski (Blunt). And, as Cage and Vrataski take the fight to the aliens, each repeated encounter gets them one step closer to defeating the enemy.

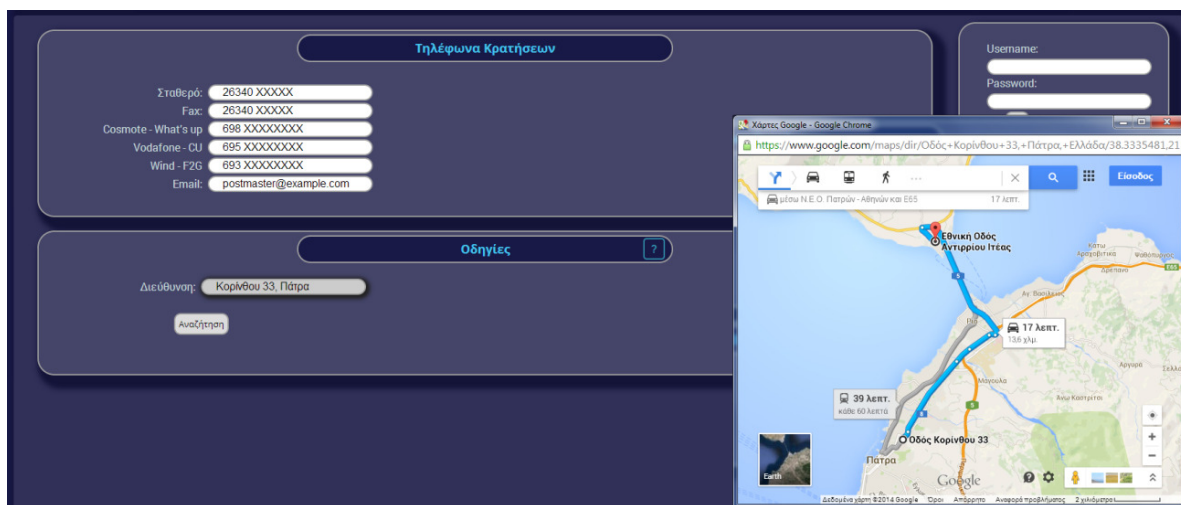
Εικόνα 4.1-4, Τρόπος Εμφάνισης Ταινίας – Προσεχώς

Κλείνοντας την παρουσίαση του βασικού κορμού άξιο αναφοράς είναι το σύστημα των κρατήσεων το οποίο δεν θα είναι προσπελάσιμο χωρίς σύνδεση και προτρέπει τον χρήστη να συνδεθεί με το ακόλουθο μήνυμα,

Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη σελίδα.

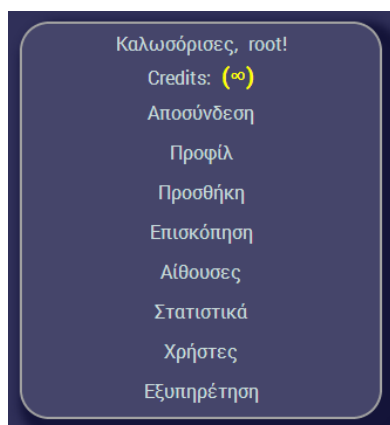
Εικόνα 4.1-5, Σφάλμα Προσπέλασης Μενού Κράτησης Χωρίς Σύνδεση

όπως και η σελίδα «Επικοινωνία» μέσω της οποίας ο χρήστης ενημερώνεται για τα τηλέφωνα, την διεύθυνση ηλεκτρονικού ταχυδρομείου και μπορεί εισάγοντας την διεύθυνση του να λάβει σε νέο παράθυρο οδηγίες τύπου πλοηγού για την ακριβή τοποθεσία του καταστήματος.

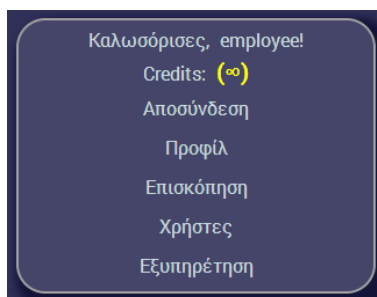


Εικόνα 4.1-6, Επικοινωνία και Οδηγίες

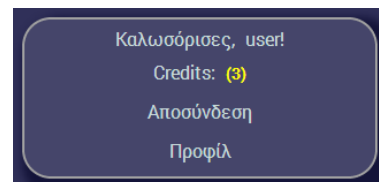
Στις επόμενες ενότητες θα παρουσιαστούν η εμφάνιση και οι λειτουργίες των εργαλείων που αναλύθηκαν και σχεδιάστηκαν με την διαδικασία που περιγράφηκε στα προηγούμενα κεφάλαια. Βασικότερο ίσως εργαλείο όλων των επιπέδων χρήστη είναι το σύστημα εισόδου στην εφαρμογή που δίνει πρόσβαση στο ιδιωτικό, μόνο για εγγεγραμμένους χρήστες, μενού πλοήγησης και διαφοροποιείται ανάλογα με το επίπεδο που διαθέτει.



Εικόνα 4.1-7,  
Μενού Διαχειριστή



Εικόνα 4.1-8,  
Μενού Υπαλλήλου



Εικόνα 4.1-9,  
Μενού Μέλους

Όπως είναι φανερό και αναμενόμενο, υψηλότερα στην ιεραρχία των επιπέδων υπάρχουν και περισσότερα εργαλεία τα οποία και θα παρουσιαστούν ιεραρχικά όπως και στα προηγούμενα κεφάλαια, δηλαδή σε αυτά που παρουσιάζονται στα κατώτερα επίπεδα έχουν πρόσβαση και τα ανώτερα, ωστόσο θα παραληφθούν αυτά τα οποία δεν εμπίπτουν στις άμεσες αρμοδιότητες ή ενδιαφέρον του κάθε επιπέδου.

## 4.2 Εργαλεία Διαχειριστή

Σε αυτή την ενότητα θα παρουσιαστούν τα εργαλεία του «Διαχειριστή» δίνοντας έμφαση στην εμφάνιση αλλά και στον τρόπο λειτουργίας τους. Πιστοποιώντας την ταυτότητά του, αποκτά πρόσβαση στο ιδιωτικό μενού πλοήγησης (Εικόνα 4.1-7) και κατ' επέκταση σε εργαλεία όπως αυτά που αναλύθηκαν και σχεδιάστηκαν.

### 4.2.1 Προσθήκη

Στο μενού προσθήκη ο «Διαχειριστής» έχει στην διάθεση του εργαλεία αυτοματοποιημένα και ενημερωμένα μέσω της βάσης για να προσθέσει ταινίες και προβολές.

Με την παρακάτω φόρμα μπορεί να ανακτά αυτόματα πληροφορίες για οποιαδήποτε ταινία μέσω της δημοφιλούς διαδικτυακής βάσης δεδομένων ταινιών IMDB, στην συνέχεια να επεξεργάζεται τα στοιχεία όπως επιθυμεί και να τα αποθηκεύει με μόνο περιορισμό να μην υπάρχει ήδη η ταινία στην Βάση Δεδομένων της εφαρμογής. Το μόνο στοιχείο που χρειάζεται για να ανακτήσει αυτόματα τα στοιχεία που ζητούνται για την αποθήκευση, είναι ο κωδικός IMDB της ταινίας που είναι εμφανής στην γραμμή διεύθυνσης του προγράμματος περιήγησής του και η μορφή του φαίνεται στο παράδειγμα παρακάτω.

Εικόνα 4.2-1, Λειτουργία Αυτόματης Προσθήκης Ταινίας (Παράδειγμα)

Κάτω από αυτήν τη φόρμα εμφανίζεται η φόρμα προσθήκης προβολών, στην οποία μπορεί να επιλέξει μόνο τις ταινίες που έχουν τεθεί σε προβολή, την αίθουσα που επιθυμεί και να διαλέξει ημερομηνία και ώρα με την βοήθεια ημερολογίου.

Εικόνα 4.2-2, Προσθήκη Προβολής

## 4.2.2 Επισκόπηση

Σε αυτήν ενότητα του μενού πλοήγησης ο «Διαχειριστής» συναντά μία σειρά από εργαλεία που παρουσιάζουν το σύνολο των πληροφοριών που έχει εισαγει. Αρχικά εμφανίζονται όλες οι ταινίες που έχουν εισαχθεί στην Βάση Δεδομένων της εφαρμογής με ορισμένα από τα στοιχεία τους: τον τίτλο, την χρονολογία, την διάρκεια, το είδος, την κατάσταση και τα έσοδα. Επίσης έχει στην διάθεση του φίλτρα, που χωρίζουν τις ταινίες ανάλογα με την κατάσταση που έχουν αποθηκευτεί, με δυνατότητα προβολής όλων των ταινιών, αυτών που είναι στο αρχείο οι οποίες θα είναι διαθέσιμες προσεχώς και αυτών που προβάλλονται αυτή τη χρονική περίοδο. Κάθε φίλτρο διαθέτει ενδεικτικό αριθμό που αντιπροσωπεύει το πλήθος ανά κατηγορία. Επίσης, υπάρχει η λειτουργία άμεσης διαγραφής ή επεξεργασίας μέσω εικονιδίων που προστέθηκαν στα δεξιά κάθε ταινίας (βλέπε Εικόνα 4.2-3).

Τίτλος	Χρονολογία	Διάρκεια	Είδος	Κατάσταση	Έσοδα
The Wolf of Wall Street	2013	180 min	Biography, Comedy, Crime	Αρχείο	195 €
12 Years a Slave	2013	134 min	Biography, Drama, History	Αρχείο	0 €
Gravity	2013	91 min	Drama, Sci-Fi, Thriller	Προσεχώς	0 €
X-Men: Days of Future Past	2014	131 min	Action, Adventure, Fantasy	Προσεχώς	0 €
Live Free or Die Hard	2007	128 min	Action, Crime, Thriller	Αρχείο	0 €
Dallas Buyers Club	2013	117 min	Biography, Drama	Προσεχώς	0 €
Dawn of the Planet of the Apes	2014	130 min	Action, Drama, Sci-Fi	Αρχείο	0 €
Hercules	2014	98 min	Action, Adventure	Αρχείο	0 €
Guardians of the Galaxy	2014	121 min	Action, Adventure, Sci-Fi	Αρχείο	0 €
Transformers: Age of Extinction	2014	165 min	Action, Adventure, Sci-Fi	Αρχείο	0 €
Teenage Mutant Ninja Turtles	2014	123 min	Action, Adventure, Comedy	Αρχείο	0 €
Lucy	2014	132 min	Action, Sci-Fi	Αρχείο	0 €
The Expendables 3	2014	106 min	Action, Adventure, Thriller	Προβάλλεται	0 €
The Hunger Games: Mockingjay - Part 1	2014	147 min	Action, Sci-Fi	Αρχείο	0 €

Εικόνα 4.2-3, Πίνακας Επισκόπησης Ταινιών

Κάτω από αυτόν τον πίνακα εμφανίζονται οι προβολές που έχουν εισαχθεί σε μορφή ημερολογίου. Δημιουργήθηκαν καρτέλες με τις ημερομηνίες που έχουν καταχωρηθεί προβολές και εμφανίζονται με ημερολογιακή σειρά. Επιλέγοντας τις επιθυμητές ημερομηνίες ο χρήστης μπορεί να εμφανίσει και να αποκρύψει τις προβολές κάθε ημέρας. Οι ταινίες εμφανίζονται με κάποια στοιχεία τους σε μορφή πίνακα και διαθέτουν επιλογή άμεσης διαγραφής και επεξεργασίας μέσω εικονιδίων.

Προβολές

01.09.2014 - Δευτέρα (8)

Τίτλος	Αίθουσα	Ημερομηνία	Ώρα	Πληρότητα	Διάρκεια	
Rio 2	1	01-09-2014	18:00	0/300	101 min	
The Expendables 3	1	01-09-2014	20:00	0/300	106 min	
Non-Stop	1	01-09-2014	22:00	0/300	103 min	
The Maze Runner	2	01-09-2014	20:00	0/400	153 min	
Non-Stop	2	01-09-2014	23:00	0/400	103 min	
Rio 2	3	01-09-2014	17:00	0/560	101 min	
The Maze Runner	3	01-09-2014	19:00	0/560	153 min	
The Expendables 3	3	01-09-2014	22:00	0/560	106 min	

02.09.2014 - Τρίτη (8)

03.09.2014 - Τετάρτη (8)

**Εικόνα 4.2-4, Πίνακας Προβολών**

Και στους δύο πίνακες, όπως είναι φανερό από την εικόνα 4.2-3, μετά από έναν αριθμό αποτελεσμάτων εμφανίζεται μπάρα κύλησης για την καλύτερη οργάνωση και πλοήγηση στα δεδομένα. Όταν επιλεγθεί επεξεργασία μίας ταινίας ή προβολής εμφανίζεται σε αναδυόμενο παράθυρο η αντίστοιχη φόρμα με τα στοιχεία της ταινίας συμπληρωμένα. Αφού επεξεργαστεί τα δεδομένα που επιθυμεί μπορεί να αποθηκεύσει την ταινία ή την προβολή. Η επεξεργασία των προβολών, εκτός από αποθήκευση και ακύρωση, διαθέτει μια επιπρόσθετη ιδιότητα, την αποθήκευση της επεξεργασμένης ως νέας, επιτρέποντας έτσι να εξοικονομηθεί χρόνος.

	Τίτλος	Χρονολογία	Διάρκεια
	The Wolf of Wall Street	2013	180 min

Κωδικός IMDB:

Τίτλος:

Χρονολογία:

Διάρκεια:

Κατηγορία:

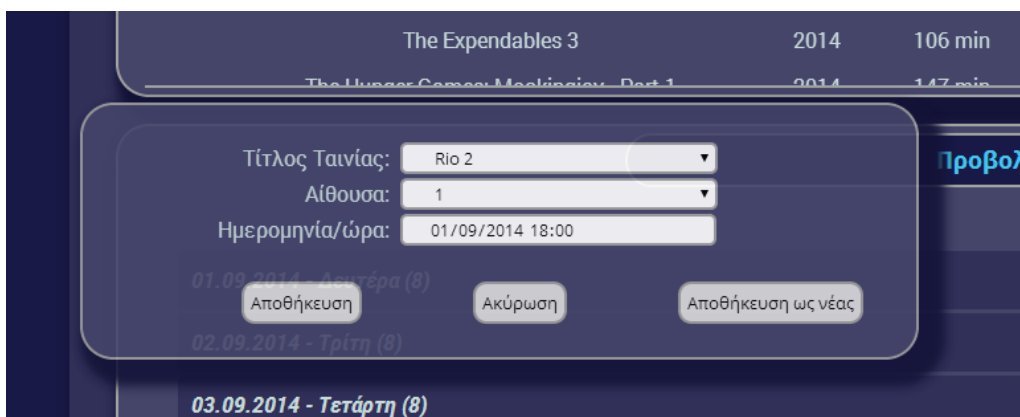
URL Εικόνας:

Κατάσταση Ταινίας:

Πλοκή: 

Based on the true story of Jordan Belfort, from his rise to a wealthy stock-broker living the high life to his fall involving crime, corruption and the federal government.

**Εικόνα 4.2-5, Επεξεργασίας Ταινίας**



Εικόνα 4.2-6, Επεξεργασία Προβολής

### 4.2.3 Αίθουσες

Σε αυτήν την ενότητα βρίσκονται όλα τα εργαλεία που αφορούν τις αίθουσες του πολύ-κινηματογράφου. Συγκεντρώθηκαν όλες οι λειτουργίες σε ένα μέρος καθώς κρίθηκε αμφίβολη η συχνή χρήση του τομέα αυτού. Το συμπέρασμα αυτό προήλθε από την σκέψη πως πέρα από την αρχική προσθήκη τους με την χωρητικότητα που αντιστοιχεί στην κάθε μία, το ενδεχόμενο τροποποίησης είναι εξαιρετικά σπάνιο.

Ωστόσο δίνεται η δυνατότητα διαγραφής, επεξεργασίας ή τροποποίησης όπως φαίνεται στην Εικόνα 4.2-9.

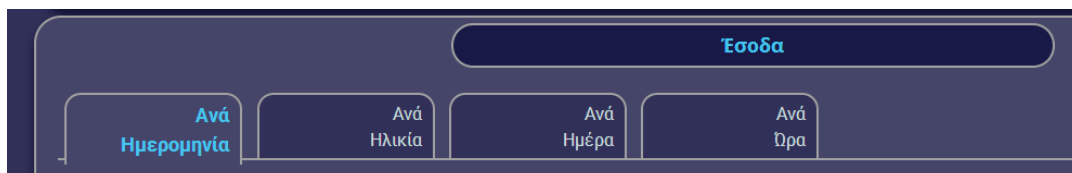


Εικόνα 4.2-7, Σύνολο Εργαλείων Αιθουσών

#### 4.2.4 Στατιστικά

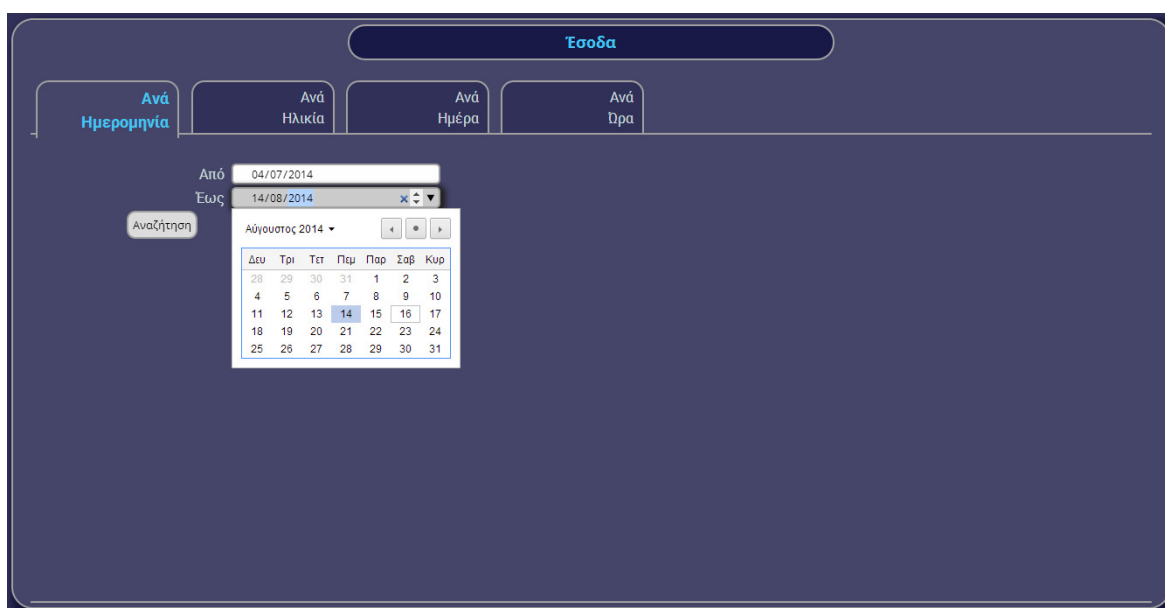
Σε αυτό το σημείο προσφέρονται στον «Διαχειριστή» χρήσιμα εργαλεία που αφορούν έσοδα και ποσοστά που έχουν εισαχθεί στην Βάση Δεδομένων και είναι ζωτικής σημασίας για την διαχείριση της ιστοσελίδας αλλά και της επιχείρησης.

Υπό μορφή καρτελών (Εικόνα 4.2-8), όπως και οι ταινίες, ο «Διαχειριστής» έχει στην διάθεσή του οργανωμένα εργαλεία που αντιστοιχούν στις παραπάνω λειτουργίες και είναι χωρισμένες σε φίλτρα ανά ημερομηνία, διαστήματα ηλικίας των μελών, ημέρα της εβδομάδας και διαστήματα ωρών λειτουργίας του πολύ-κινηματογράφου.

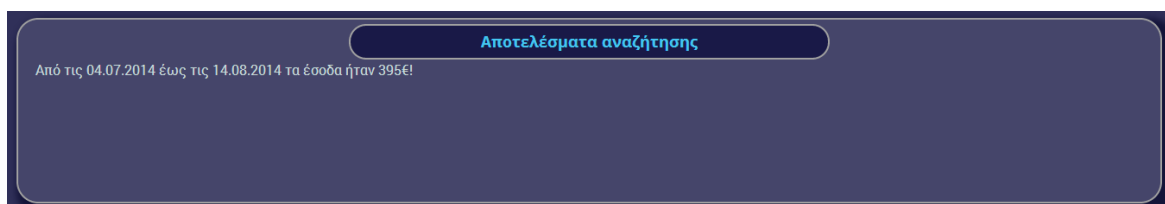


Εικόνα 4.2-8, Φίλτρα Στατιστικών

Στην ημερομηνία εμφανίζεται μία φόρμα που μπορεί να επιλέξει ένα διάστημα οποιονδήποτε ημερομηνιών, με την βοήθεια ημερολογίου, για να εμφανίσει τα συνολικά έσοδα σε αυτό. Η εφαρμογή αναγνωρίζει αν το διάστημα είναι εφικτό και εμφανίζει τα ανάλογα αποτελέσματα. Συγκεκριμένα, σε εφικτό διάστημα εμφανίζει τα έσοδα σε αυτό, στην περίπτωση που εισαχθεί ίδια ημερομηνία εμφανίζει τα έσοδα της συγκεκριμένης μέρας ενώ αν το διάστημα δεν είναι εφικτό εμφανίζει ανάλογο μήνυμα.



Εικόνα 4.2-9, Φόρμα Αναζήτησης Εσόδων



Εικόνα 4.2-10, Αποτελέσματα Αναζήτησης Εσόδων

Στην συνέχεια έχει πρόσβαση σε μορφή πίνακα στα υπόλοιπα στατιστικά που προαναφέρθηκαν κατά την ανάλυση και την σχεδίαση της εφαρμογής. Αρχικά συλλέγονται στοιχεία από την Βάση Δεδομένων μετατρέποντας την ημερομηνία γέννησης των χρηστών σε ηλικία, στην συνέχεια χωρίζονται σε κατηγορίες και μετρίεται το πλήθος και το σύνολο των εσόδων που αντιστοιχούν στα διαστήματα ηλικιών. Έτσι μπορεί να ενημερωθεί για το σύνολο χρηστών ανά κατηγορία ηλικίας, το ποσοστό που διαθέτει κάθε κατηγορία επί του συνόλου εγγεγραμμένων χρηστών και τα έσοδα από κάθε διάστημα με τον τρόπο που παρουσιάζεται στην Εικόνα 4.2-11.

Έσοδα						
Ανά Ημερομηνία	Ανά Ηλικία	Ανά Ημέρα	Ανά Ώρα			
18-25	26-33	34-40	41-60	61+	Σύνολο	
3	3	0	0	0	6	
50%	50%	0%	0%	0%	-	
420€	220€	0€	0€	0€	640€	

Εικόνα 4.2-11, Στατιστικά Ηλικιών Μελών

Ομοίως, δημιουργήθηκε σύστημα συλλογής δεδομένων από όλα τα έσοδα στα οποία υπάρχει πάντα ημερομηνία ανά συναλλαγή. Αυτή η ημερομηνία μετατρέπεται σε ημέρα της εβδομάδας και έτσι καταμετρώνται οι συνολικές πληρωμένες κρατήσεις που έχουν γίνει ανά ημέρα εβδομάδας, το ποσοστό των κρατήσεων αυτών επί των συνολικών κρατήσεων και το σύνολο των εσόδων ανά ημέρα. Τα στοιχεία οργανώνονται σε πίνακα και παρουσιάζονται στον «Διαχειριστή» όπως φαίνεται στην Εικόνα 4.2-12.

Έσοδα							
Ανά Ημερομηνία	Ανά Ηλικία	Ανά Ημέρα	Ανά Ώρα				
Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή	Σάββατο	Κυριακή	Σύνολο
10	0	0	0	39	0	0	49
20.4%	0%	0%	0%	79.6%	0%	0%	-
50€	0€	0€	0€	195€	0€	0€	245€

Εικόνα 4.2-12, Στατιστικά Ημερών της Εβδομάδας



Τέλος, με παρόμοιο τρόπο με τις ημέρες της εβδομάδας, γίνεται συλλογή των εσόδων με την ημερομηνία όμως να μετατρέπεται σε ώρα της ημέρας. Στην συνέχεια χωρίστηκαν διαστήματα ωρών λειτουργίας του πολύ-κινηματογράφου, όπου για κάθε κατηγορία καταμετρώνται οι συνολικές πληρωμένες κρατήσεις που έχουν γίνει στο συγκεκριμένο διάστημα, το ποσοστό αυτών που αντιστοιχεί στο καθένα καθώς και το σύνολο των εσόδων. Αφού γίνει διαλογή των δεδομένων, οργανώνονται σε πίνακα είναι στην διάθεση του «Διαχειριστή» όπως παροσιάζονται στην Εικόνα 4.2-13.

Έσοδα							
Ανά Ημερομηνία	Ανά Ηλικία	Ανά Ημέρα	Ανά Ώρα				
13:00-16:30	16:31-18:30	18:31-19:30	19:31-20:30	20:31-21:30	21:31-22:30	22:31+	Σύνολο
0	10	0	39	0	0	0	49
0%	20.4%	0%	79.6%	0%	0%	0%	-
0€	50€	0€	195€	0€	0€	0€	245€

Εικόνα 4.2-13, Στατιστικά Ωρών Λειτουργίας

### 4.2.5 Χρήστες

Σε αυτό το μενού ο «Διαχειριστής» μπορεί να αναζητήσει τον χρήστη μέσω του μοναδικού του αναγνωριστικού (από την Κάρτα Μέλους σε μορφή QRcode) διαφορετικά μέσω του ονόματος χρήστη από δύο φορμών.

Εύρεση βάση κωδικού ?

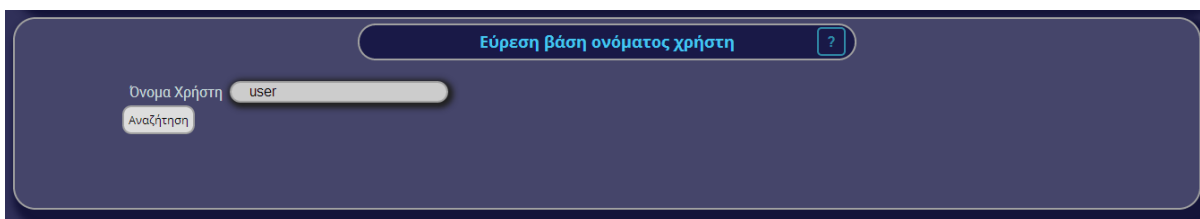
Αναγνωριστικό Χρήστη

Εύρεση βάση ονόματος χρήστη ?

Όνομα Χρήστη

Εικόνα 4.2-14, Τρόποι Αναζήτησης Χρήστη

Ειδικότερα στην δεύτερη φόρμα η αναζήτηση δεν είναι απαραίτητο να αντιπροσωπεύει ακριβώς το όνομα του χρήστη αλλά εμφανίζει τα αποτελέσματα που ταιριάζουν στο αλφαριθμητικό που εισήχθη μαζί με σημαντικές λεπτομέρειες για τον καθένα όπως φαίνεται παρακάτω.

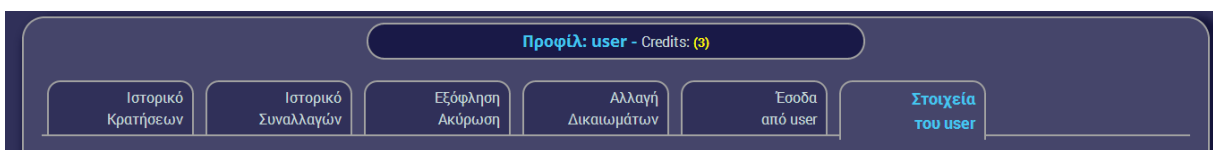


Εικόνα 4.2-15, Αναζήτηση Χρήστη "user"

Όνομα Χρήστη	Αναγνωριστικό	Ονοματεπώνυμο	email	credits	Δικαιώματα
user	bbbbbbbbbbbbbbbbbb	firstname lastname	user@example.com	3	Χρήστης
user2	cccccccccccccccccc	user2 user2	user2@cinemacied.hopto.org	3	Χρήστης
user3	39vrsygo1nv0brdofjoz	user3 user3	user3@cinemacied.hopto.org	0	Χρήστης

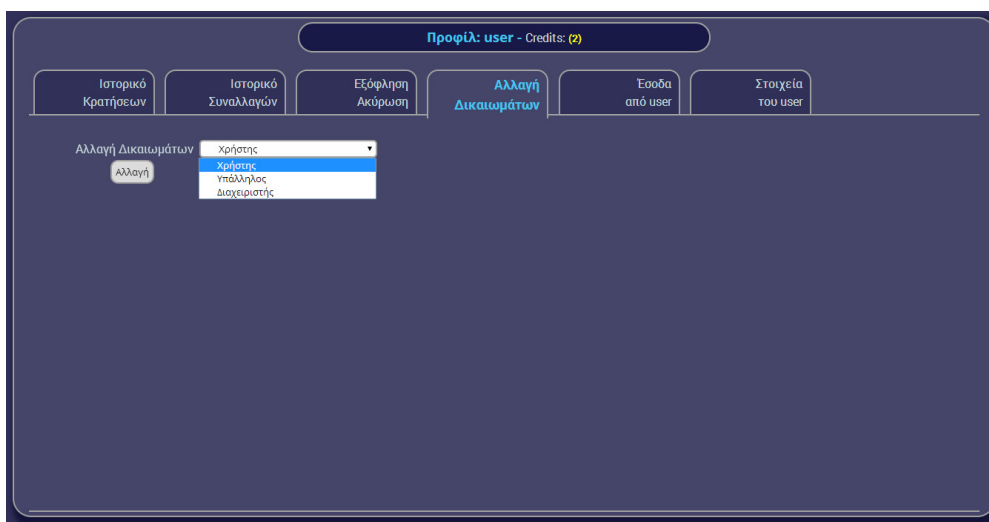
Εικόνα 4.2-16, Αποτελέσματα Αναζήτησης "user"

Ωστόσο αποτέλεσμα και των δύο τρόπων αναζητήσεων είναι το προφίλ του χρήστη. Εμφανίζεται, για εξοικονόμηση χώρου και καλύτερη προβολή, σε μορφή καρτελών περιέχοντας η κάθε μία από ένα εργαλείο ελέγχου ή εξυπηρέτησης. Η διάταξη των καρτελών φαίνεται στην Εικόνα 4.2-17.



Εικόνα 4.2-18, Διάταξης Καρτελών Προφίλ Χρήστη (Διαχειριστής)

Πέρα από τις πρώτες καρτέλες, ιστορικό κρατήσεων, συναλλαγών, εξόφληση/ακύρωση που χρησιμεύουν περισσότερο στον υπάλληλο οπότε και θα παρουσιαστούν στην αντίστοιχη ενότητα παρακάτω, τα εργαλεία που είναι προς εκμετάλλευση του διαχειριστή είναι η αλλαγή δικαιωμάτων και τα έσοδα από τον συγκεκριμένο χρήστη.



Εικόνα 4.2-19, Φόρμα Αλλαγής Δικαιωμάτων

Όσον αφορά τον πίνακα εσόδων στο πάνω μέρος του παρουσιάζεται το σύνολο και σε φθίνουσα σειρά οι ημέρες που υπάρχουν μη μηδενικά έσοδα. Είναι σημαντικό να αναφέρουμε πως στην περίπτωση που το προφίλ ανήκει σε υπάλληλο τα έσοδα αντιστοιχούν στο ημερήσιο ταμείο του.

Προφίλ: user - Credits: (2)					
Ιστορικό Κρατήσεων	Ιστορικό Συναλλαγών	Εξόφληση Ακύρωση	Αλλαγή Δικαιωμάτων	Έσοδα από user	Στοιχεία του user
<b>ΣΥΝΟΛΟ</b>		<b>115€</b>			
	03.08.2014	25€			
	29.07.2014	15€			
	22.06.2014	25€			
	16.06.2014	20€			
	03.06.2014	20€			
	02.06.2014	10€			

Εικόνα 4.2-20, Πίνακας Εσόδων Χρήστη

## 4.3 Εργαλεία Υπαλλήλου

Ο υπάλληλος, που όπως προαναφέρθηκε θα χρησιμοποιεί την εφαρμογή και για τις μη ηλεκτρονικές συναλλαγές που αφορούν την επιχείρηση, θα πρέπει να εισέλθει με τα στοιχεία που έχει επιλέξει ώστε να αποκτήσει πρόσβαση στο μενού των υπαλλήλων (Εικόνα 4.1-8). Έτσι, αποκτά πρόσβαση σε μία σειρά εργαλείων που καθιστούν τον φόρτο των αρμοδιοτήτων του πιο ελαφρύ, καθώς επίσης τον βοηθούν να φέρει εις πέρας τις υποχρεώσεις του προς την επιχείρηση.

### 4.3.1 Επισκόπηση

Σε αυτή την ενότητα ο «Υπάλληλος» έχει πρόσβαση μόνο στις προβολές χωρίς να εμφανίζεται η επισκόπηση των ταινιών. Οι προβολές εμφανίζονται με την ίδια μορφή που εμφανίζονται και στον «Διαχειριστή» χωρίς τα εικονίδια όμως για διαγραφή και επεξεργασία.

Προβολές					
01.09.2014 - Δευτέρα (8)					
Τίτλος	Αίθουσα	Ημερομηνία	Ώρα	Πληρότητα	Διάρκεια
Rio 2	1	01-09-2014	18:00	0/300	101 min
The Expendables 3	1	01-09-2014	20:00	0/300	106 min
Non-Stop	1	01-09-2014	22:00	0/300	103 min
The Maze Runner	2	01-09-2014	20:00	0/400	153 min
Non-Stop	2	01-09-2014	23:00	0/400	103 min
Rio 2	3	01-09-2014	17:00	0/560	101 min
The Maze Runner	3	01-09-2014	19:00	0/560	153 min
The Expendables 3	3	01-09-2014	22:00	0/560	106 min
02.09.2014 - Τρίτη (8)					
03.09.2014 - Τετάρτη (8)					

Εικόνα 4.3-1, Επισκόπηση Προβολών Δικαιώματα Υπαλλήλου

Μέσω αυτής της λειτουργίας ο «Υπάλληλος» μπορεί, κατόπιν οδηγίας της επιχείρησης, συγκεκριμένη ώρα πριν την αρχή της προβολής να διαγράφει τις εκκρεμείς κρατήσεις, ώστε να μην υπάρχουν ανώφελα, απούλητα εισιτήρια. Πατώντας λοιπόν πάνω στην πληρότητα κάθε προβολής ανακατευθύνεται σε νέα καρτέλα, όπου εμφανίζονται όλες οι κρατήσεις με δυνατότητα πολλαπλής επιλογής ή επιλογή όλων και δυνατότητα διαγραφής. Δεν λείπουν τα φίλτρα προβολής όλων των εισιτηρίων, των πληρωμένων ή αυτών που εκκρεμεί η πληρωμή τους, ενώ δίνεται και η δυνατότητα ακύρωσης πληρωμένου εισιτηρίου για ιδιαίτερες καταστάσεις όπως λάθος υπαλλήλου, επιστροφή, λάθος συνεννόηση κτλ.

Rio 2 - 01.09.2014 - 17:00 - Αίθουσα: 3			
(31) Όλες	(17) Πληρωμένες	(14) Εκκρεμεί	
Ακύρωση Επιλεγμένων			
<input type="checkbox"/>	Κωδικός	Κατάσταση	Θέση
<input type="checkbox"/>	zod44pzbbe	OK	466
<input type="checkbox"/>	biy8hwt716	OK	467
<input type="checkbox"/>	zs4m9waxz2	OK	468
<input type="checkbox"/>	nb071hotun	OK	469
<input type="checkbox"/>	uvdluufgjo	OK	470
<input type="checkbox"/>	urls6korj5	OK	488
<input type="checkbox"/>	idcmehkk0f	OK	489
<input checked="" type="checkbox"/>	098y4x3bp5	N.OK	491
<input checked="" type="checkbox"/>	l00tqj9x7	N.OK	492
<input checked="" type="checkbox"/>	2a8f6b1b14	N.OK	493
<input checked="" type="checkbox"/>	cim2m2q1tm	N.OK	494
<input checked="" type="checkbox"/>	1bg8trfhr	N.OK	495

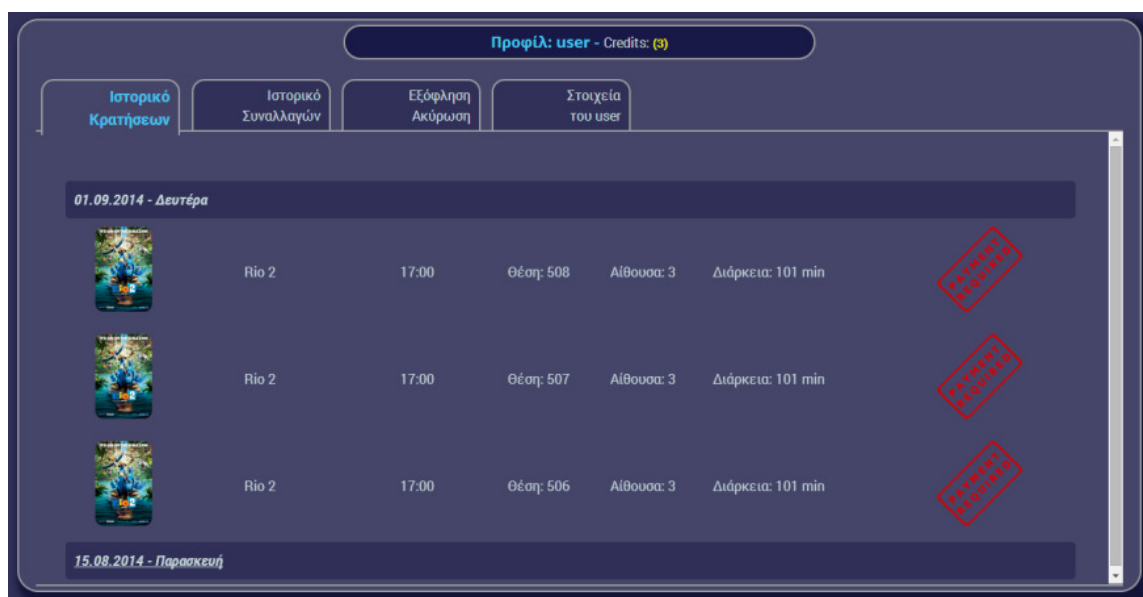
Εικόνα 4.3-2, Διαγραφή Κρατήσεων που Εκκρεμούν

### 4.3.2 Χρήστες

Ο υπάλληλος έχει πρόσβαση εξ ολοκλήρου στο μενού αναζήτησης χρηστών όπως παρουσιάστηκε παραπάνω στην Εικόνα 4.2-14 όπου θα μπορεί να σαρώνει το κώδικα QR που θα βρίσκεται πάνω στην κάρτα μέλους ή με το όνομα χρήστη που επέλεξε το

μέλος κατά την εγγραφή, με τον ίδιο τρόπο που αναζητά και ο «Διαχειριστής», επιτρέποντας την μη ακριβή αναζήτηση του ονόματος και εμφάνιση πολλαπλών αποτελεσμάτων (Εικόνα 4.2-16).

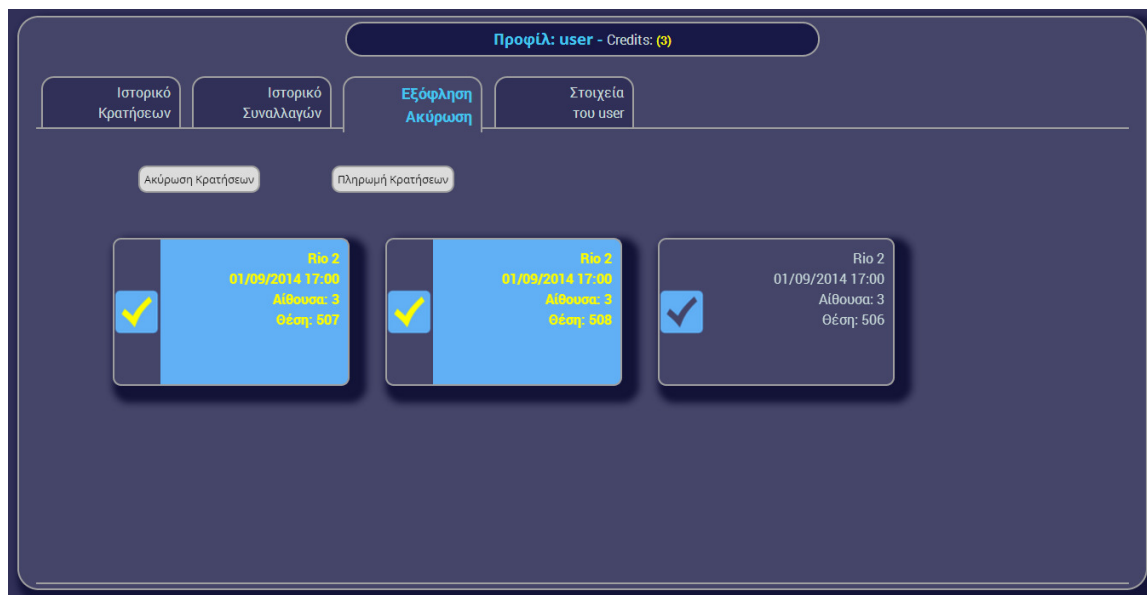
Ωστόσο δεν έχει πρόσβαση σε ορισμένα στοιχεία στο προφίλ του χρήστη. Παρουσιάζονται παρακάτω οι διαθέσιμες καρτέλες και η πρώτη καρτέλα που αφορά το ιστορικό κρατήσεων ενός χρήστη που χρησιμεύει στην εξυπηρέτηση ενός μέλους που μόλις έχει φτάσει και θέλει να εκδώσει πληρωμένο διαδικτυακά εισιτήριο ή να το εξοφλήσει.



**Εικόνα 4.3-3, Προφίλ Χρήστη - Ιστορικό Κρατήσεων - Δικαιώματα Υπαλλήλου**

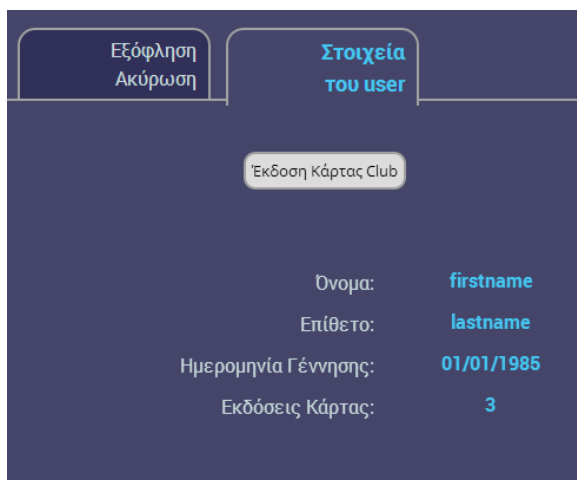
Όπως φαίνεται στην Εικόνα 4.3-3 οι κρατήσεις παρουσιάζονται χωρισμένες σε καρτέλες με ημερομηνίες ξεκινώντας από την πιο πρόσφατη κράτηση και συνεχίζοντας με φθίνουσα σειρά, στην συνέχεια κάποια βασικά στοιχεία των κρατήσεων και τέλος ένδειξη της κατάστασής τους, δηλαδή αν είναι εξοφλημένες ή αν εκκρεμεί η πληρωμή τους.

Στην συνέχεια μεταβαίνοντας στην καρτέλα Εξόφληση/Ακύρωση μπορεί να επιλέξει τις ενεργές κρατήσεις σε μορφή πλακιδίων και όπως φαίνεται στην Εικόνα 4.3-4 να τις ακυρώσει ή να τις εξοφλήσει.



Εικόνα 4.3-4, Εξόφληση/Ακύρωση Εισιτηρίων

Τέλος από την καρτέλα «Στοιχεία του χρήστη» μπορεί να επιβεβαιώσει, στην πρώτη επαφή της επιχείρησης με το μέλος, τα στοιχεία του: όνομα, επίθετο, ημερομηνία γέννησης και πόσες φορές έχει εκδοθεί κάρτα μέλους και επιλογή για εκ νέου έκδοσή της.



Εικόνα 4.3-5, Στοιχεία του Χρήστη



Εικόνα 4.3-6, Κάρτα Μέλους

Εφόσον καλύφθηκε η παρουσίαση των εργαλείων του υπαλλήλου, στην συνέχεια θα παρουσιαστεί αφενός το σύστημα κρατήσεων αναλύοντας τον τρόπο λειτουργίας του και αφετέρου ο τρόπος εξυπηρέτησης ενός πελάτη που έκανε κράτηση ή θέλει να παρακολουθήσει μία προβολή χωρίς να είναι μέλος.

Με τα παρακάτω εργαλεία επιτυγχάνεται η εξ ολοκλήρου χρήση της εφαρμογής από τον πολύ-κινηματογράφο ώστε να μην γίνεται διαχωρισμός πελατών και μελών και λόγω της ανάγκης συλλογής των στατιστικών στοιχείων, παρακολούθησης ταμείων των υπαλλήλων και άλλων.

### 4.3.3 Κράτηση

Αρχικά άξιο αναφοράς είναι το γεγονός ότι το σύστημα κράτησης σχεδιάστηκε και υλοποιήθηκε ώστε να παρουσιάζει πληροφορίες και να προχωρά βήμα – βήμα, ώστε να εξοικονομείται χρόνος και κάθε πληροφορία που προέρχεται από τον πελάτη να φέρνει τον υπάλληλο ένα βήμα πιο κοντά στην επιτυχή εξυπηρέτηση.

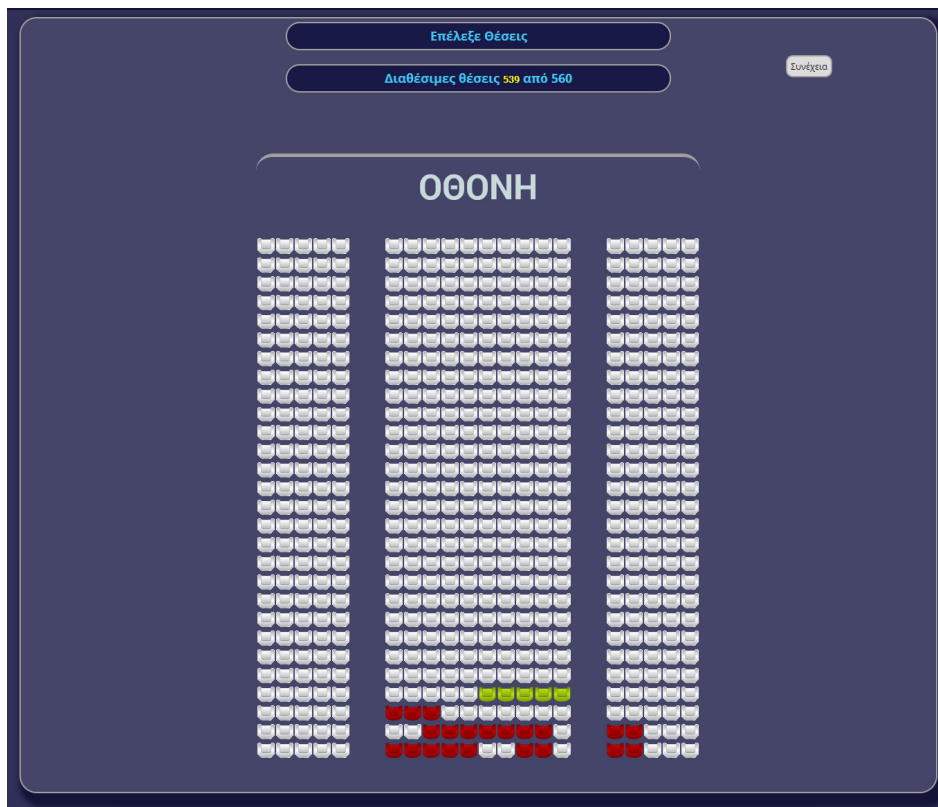
Επιλέγοντας από το κεντρικό μενού «Κράτηση» εμφανίζονται δύο επιλογές, όπως φαίνεται παρακάτω στην Εικόνα 4.3-7, μπορεί να γίνει επιλογή ταινίας ή ημερομηνίας, στην συνέχεια εμφανίζονται οι διαθέσιμες προβολές με ημερομηνία και ώρα ή οι ταινίες που προβάλλονται την συγκεκριμένη ημερομηνία αντίστοιχα ώστε να επιλεγθούν τελικά όλα τα στοιχεία της προβολής, τίτλος, ημερομηνία, ώρα, αίθουσα.



Εικόνα 4.3-7, Σύστημα Κρατήσεων

Αφού καθοριστεί λοιπόν η επιθυμητή προβολή, εμφανίζεται μία εικονική αίθουσα με εμφανή τα χαρακτηριστικά της όπως θέση της οθόνης, των διαδρόμων και εικονίδια θέσεων. Οι θέσεις απεικονίζονται με χρώματα, με το κόκκινο να σημαίνει ότι η θέση είναι ήδη κρατημένη, το γκρι ότι είναι διαθέσιμη για επιλογή και το πράσινο ήδη επιλεγμένη θέση. Ο «Υπάλληλος» μπορεί να επιλέξει τις θέσεις που επιθυμεί ο πελάτης και ύστερα να

πατήσει συνέχεια. Επίσης για λόγους ασφαλείας ώστε να αποφευχθεί η άσκοπη κράτηση θέσεων το σύστημα δεν επιτρέπει επιλογή άνω των πέντε θέσεων.



Εικόνα 4.3-8, Εικονική Αίθουσα Επιλογής Κρατήσεων

Στην συνέχεια μεταφέρεται σε μία φόρμα επιβεβαίωσης από την οποία μπορεί είτε να εκδώσει απευθείας το εισιτήριο είτε να κάνει κράτηση με όνομα.

Εικόνα 4.3-9, Φόρμα Επιβεβαίωσης Έκδοσης/Κράτησης - Υπάλληλος

Χάριν παραδείγματος, υποθέσαμε ότι ο κύριος Ιωάννης Ιωάννου κατόπιν τηλεφωνήματος έκανε κράτηση για την ταινία. Στην επόμενη ενότητα αυτού του κεφαλαίου θα παρουσιαστεί ο τρόπος εξυπηρέτησης των πελατών που δεν είχαν φυσική παρουσία στο κατάστημα.



### 4.3.4 Εξυπηρέτηση

Από αυτό το μενού, ο «Υπάλληλος» έχει πρόσβαση σε δύο φόρμες «Αναζήτησης Κράτησης» και «Ελέγχου Εγκυρότητας Εισιτηρίου» που οι λειτουργίες τους είναι προφανής από τους τίτλους.

The screenshot shows a dark blue interface with two main sections. The top section is titled 'Αναζήτηση Κράτησης' and contains a text input field labeled 'Όνομα Κράτησης:' with a search button below it. The bottom section is titled 'Έλεγχος Εγκυρότητας Εισιτηρίου' and contains a text input field labeled 'Κωδικός Εισιτηρίου:' with a search button below it.

Εικόνα 4.3-10, Μενού Εξυπηρέτηση

Κατά την αναζήτηση μίας κράτησης ο υπάλληλος δεν είναι απαραίτητο να εισάγει ακριβώς το όνομα, καθώς η αναζήτηση γίνεται με τέτοιο τρόπο ώστε να βρεθούν όλα τα ονόματα που έχουν αποθηκευτεί και εμπεριέχουν το κείμενο που αναζητήσε. Αναζητώντας για παράδειγμα την κράτηση που εισάγαμε στην Εικόνα 4.3-9, εμφανίζονται με τα στοιχεία τους, οι κρατήσεις που ταιριάζουν στην αναζήτηση (Εικόνα 4.3-11) και καλείται ο υπάλληλος να επιλέξει τα επιθυμητά ώστε να τα ακυρώσει ή να τα εξοφλήσει.

The screenshot shows the search results for 'ΙΩΑΝΝΟΥ'. At the top, there are two tabs: 'Ακύρωση Κρατήσεων' and 'Πληρωμή Κρατήσεων'. Below the tabs is a grid of booking cards. Each card contains the following information: a checkmark icon, the title 'The Maze Runner', the date and time '01/09/2014 20:00', the number of seats 'Αίθουσα: 2', the seat number 'Θέση: 386', and the name 'ΙΩΑΝΝΗΣ ΠΑΠΑΙΩΑΝΝΟΥ'. There are three such cards in the first row. The second row contains three cards with the title 'Πιο 2', date and time '01/09/2014 17:00', 'Αίθουσα: 3', and seat numbers 491, 492, and 493. The third row contains two cards with the title 'Πιο 2', date and time '01/09/2014 17:00', 'Αίθουσα: 3', and seat numbers 494 and 495.

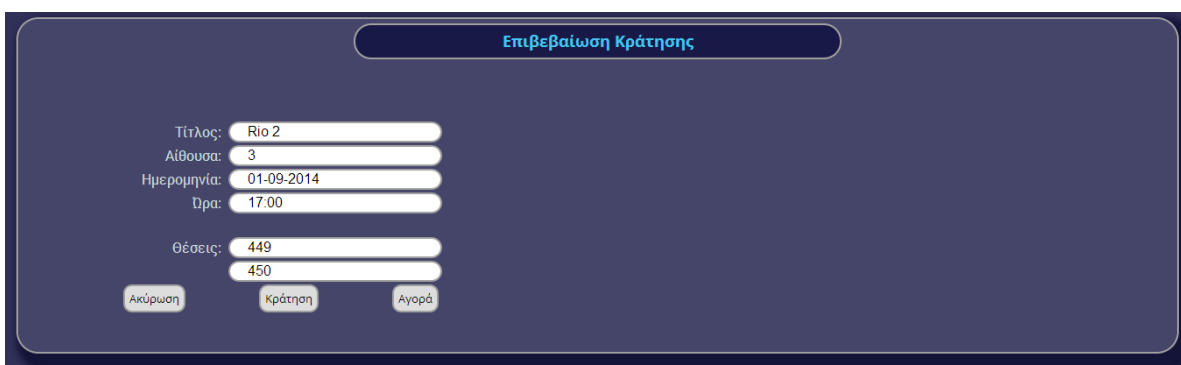
Εικόνα 4.3-11, Αναζήτηση Κράτησης

## 4.4 Εργαλεία Μέλους

Αφού πραγματοποιήσει είσοδο αποκτά πρόσβαση στο σύστημα κρατήσεων και στο μενού πλοήγησης χρηστών Εικόνα 4.1-9.

### 4.4.1 Κράτηση

Μέσα από τον λογαριασμό του μπορεί να κάνει κράτηση ή αγορά ακολουθώντας τα ίδια βήματα που αναλύθηκαν στην ενότητα του υπαλλήλου με την διαφορά ότι δεν εμφανίζεται το πεδίο κειμένου στην φόρμα επιβεβαίωσης τις οποίες εισήγαγε το όνομα της κράτησης ο «Υπάλληλος». Σε αυτό το σημείο επιλέγοντας αγορά αφαιρούνται αυτόματα πόντοι από τον λογαριασμό του, ανάλογοι του πλήθους των επιλεγμένων θέσεων εφόσον υπάρχουν διαθέσιμοι. Έτσι μπορεί να προσέλθει στο κατάστημα με την κάρτα μέλους του για να παραλάβει το εισιτήριο του.



**Επιβεβαίωση Κράτησης**

Τίτλος: Rio 2

Αίθουσα: 3

Ημερομηνία: 01-09-2014

Ώρα: 17:00

Θέσεις: 449

450

Ακύρωση    Κράτηση    Αγορά

Εικόνα 4.4-1, Φόρμα Επιβεβαίωσης Αγοράς/Κράτησης – Μέλος

### 4.4.2 Προφίλ

Το μέλος δεν έχει την δυνατότητα να προβάλει το προφίλ άλλων χρηστών, μπορεί όμως να προβάλει το δικό του προφίλ, χωρίς ωστόσο κάποιες λειτουργίες στις οποίες έχουν πρόσβαση το επίπεδο του υπαλλήλου και του διαχειριστή. Στα δεξιά της ιστοσελίδας μπορεί να ακολουθήσει την ενότητα «Προφίλ» η οποία θα τον ανακατευθύνει σε αυτό. Σκοπός ύπαρξής του είναι η ενημέρωση του μέλους για τις κρατήσεις του σε όλο το διάστημα από την εγγραφή του, η εμφάνιση του ιστορικού των συναλλαγών του με το σύστημα προσθήκης πόντων, εργαλεία διαχείρισης των κρατήσεων του και εργαλείο αλλαγής κωδικού. Τα παραπάνω εργαλεία είναι χωρισμένα σε καρτέλες και εμφανίζονται όπως στην Εικόνα 4.4-3.



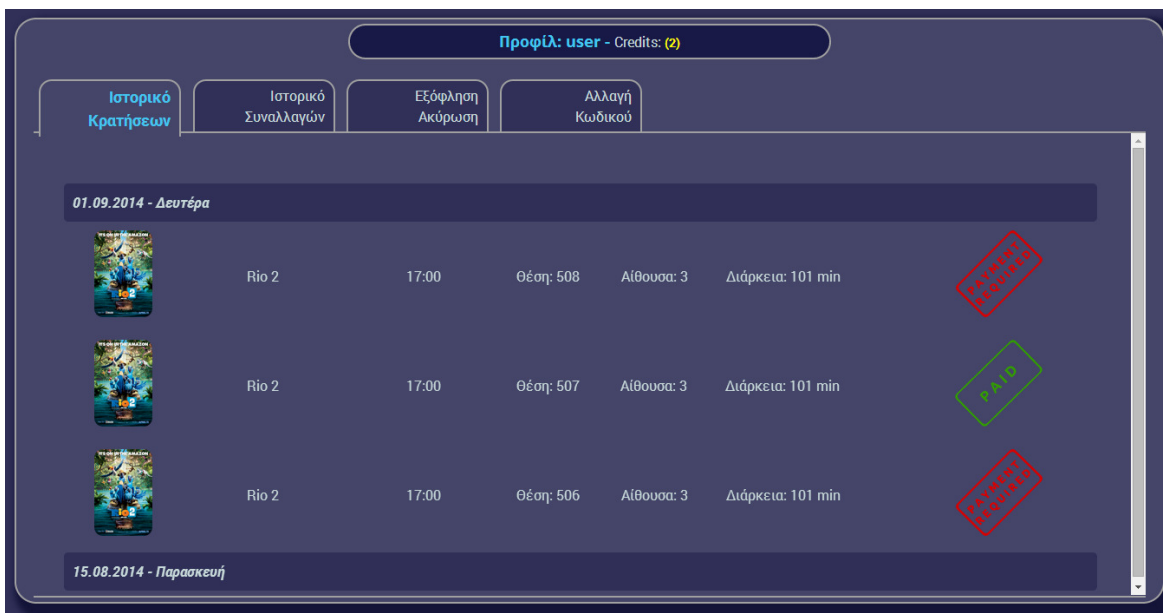
Εικόνα 4.4-2, Προφίλ Χρήστη

Πατώντας πάνω στον τίτλο στην περιοχή που εμφανίζονται οι πόντοι ή στο ίδιο σημείο στο μενού πλοήγησής του, ανακατευθύνεται σε μία φόρμα προσθήκης πόντων στον λογαριασμό του μέσω πιστωτικής ή χρεωστικής κάρτας με ανώτερο αριθμό συναλλαγής τους πέντε πόντους κάθε φορά.<sup>3</sup>

Εικόνα 4.4-3, Φόρμα Προσθήκης Πόντων

Από την καρτέλα ιστορικό κρατήσεων μπορεί να ενημερωθεί για τις κρατήσεις που έχει κάνει στο παρελθόν ώστε να εκπληρώνει τις υποχρεώσεις του προς την επιχείρηση. Οι κρατήσεις εμφανίζονται χωρισμένες ανά ημερομηνία ξεκινώντας με την πιο πρόσφατη και ακολουθώντας φθίνουσα σειρά. Αφού επιλεγθεί ημερομηνία, κάθε κράτηση εμφανίζεται με μία μικρογραφία της αφίσας, τον τίτλο της ταινίας, την ώρα, την θέση, την αίθουσα και την διάρκεια της, μαζί με μία ένδειξη για την κατάστασή της όσον αφορά την πληρωμή.

<sup>3</sup> | Σημ.: Για λόγους ασφαλείας και όρων χρήσης των υπηρεσιών της ΕΔΕΤ που αφορούν την χρήση των server για μη κερδοσκοπικούς σκοπούς καθώς και για λόγους αποφυγής αποθήκευσης ευαίσθητων στοιχείων η φόρμα δεν επαληθεύει τα στοιχεία που εισάγονται και προσθέτει τους πόντους χωρίς να αποθηκεύει κανένα τραπεζικό στοιχείο.



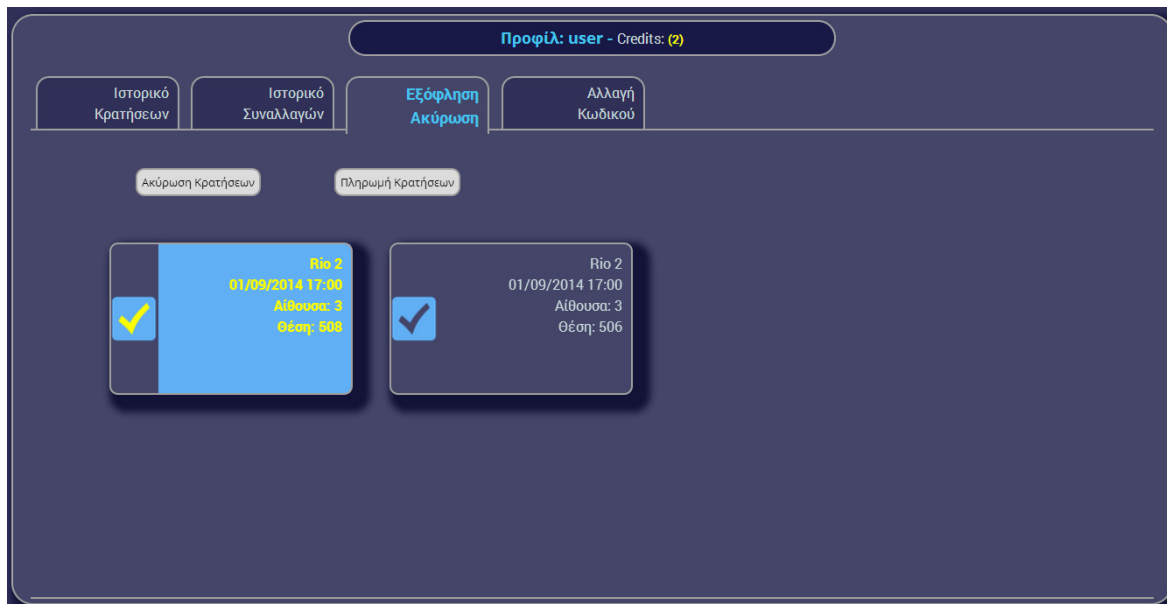
Εικόνα 4.4-4, Ιστορικό Χρήστη

Στην διάθεση του χρήστη έχει επίσης διατεθεί για λόγους ενημέρωσης και ελέγχου το ιστορικό των συναλλαγών του με το σύστημα προσθήκης πόντων. Ο χρήστης αφού κάνει μία συναλλαγή, εκτός από το μήνυμα της επιτυχούς συναλλαγής, μπορεί να επισκεφτεί το συγκεκριμένο σημείο του προφίλ του ώστε να ελέγξει την συναλλαγή αλλά και τις παλαιότερες ξεκινώντας από την πιο πρόσφατη και συνεχίζοντας σε φθίνουσα σειρά.

Ποσό	Ημερομηνία	Ωρα
25 €	03/08/2014	18:08:31
15 €	29/07/2014	16:08:40
25 €	22/06/2014	00:08:55
20 €	16/06/2014	02:08:27
20 €	03/06/2014	02:08:36
10 €	02/06/2014	15:08:26

Εικόνα 4.4-5, Ιστορικό Συναλλαγών Χρήστη

Το «Μέλος» έχει επίσης την δυνατότητα, αφού πραγματοποιήσει μία κράτηση, σε δεύτερο χρόνο να την ακυρώσει ή να την εξοφλήσει. Το εργαλείο για αυτή την διαδικασία είναι στην επόμενη καρτέλα με τίτλο Εξόφληση/Ακύρωση. Εμφανίζονται όπως στην εικόνα 4.4-6 με δυνατότητα πολλαπλής επιλογής όπου αφού επιλέξει τις επιθυμητές μπορεί να εκτελέσει και τις διαθέσιμες λειτουργίες, πληρωμή ή ακύρωση των κρατήσεων.



Εικόνα 4.4-6, Εξόφληση/Ακύρωση Κράτησης Μέλους

Τέλος, από την τελευταία καρτέλα που εμφανίζεται στο προφίλ όλων των χρηστών, έχει την δυνατότητα να αλλάξει τον προσωπικό του κωδικό που χρησιμοποιεί για σύνδεση ως μέλος. Επιλέγοντάς την, εμφανίζεται μία φόρμα αλλαγής κωδικού όπου πρέπει να εισάγει τον παλιό του κωδικό και δύο φορές τον νέο για επιβεβαίωση. Η φόρμα αφού κάνει τους απαραίτητους ελέγχους αλλάζει τον κωδικό και ενημερώνει τον χρήστη.



Εικόνα 4.4-7, Φόρμα Αλλαγής Κωδικού

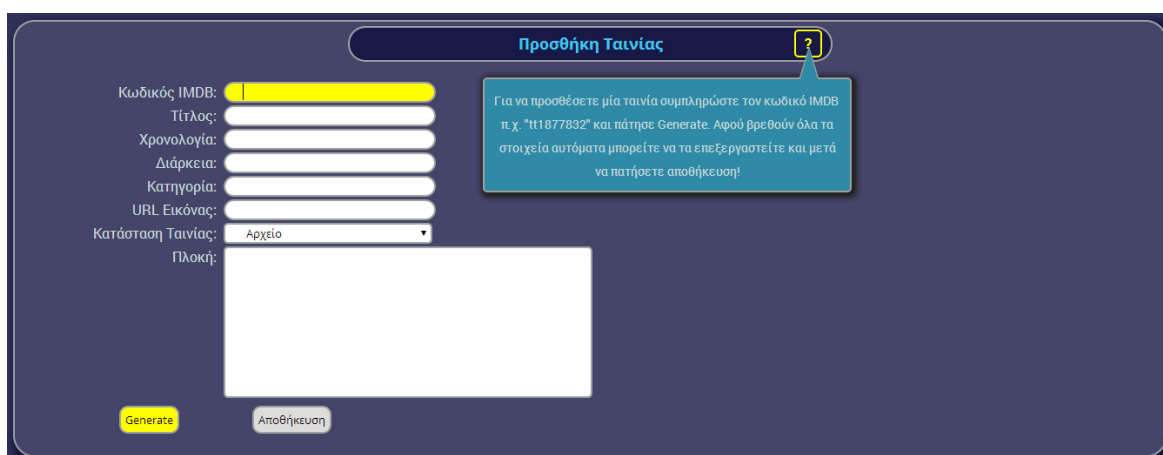
## 4.5 Βοηθητικές Λειτουργίες

Κατά την διάρκεια ανάπτυξης της εφαρμογής κρίθηκε σκόπιμο να υπάρχουν λειτουργίες που θα διευκολύνουν τους χρήστες, κυρίως ενημερωτικού χαρακτήρα. Για αυτόν τον λόγο δημιουργήθηκαν βοηθητικά μηνύματα για τα εργαλεία που διευκρινίζουν τον τρόπο λειτουργίας και μηνύματα αποτελέσματος φορμών ώστε να ενημερώνεται ο χρήστης για την πορεία του αιτήματος που έθεσε στην εφαρμογή.

### 4.5.1 Βοηθητικά Μηνύματα

Σε όλη την έκταση της εφαρμογής, όπου κρίθηκε ότι η λειτουργία κάποιου εργαλείου δεν είναι σαφής, προστέθηκε στον τίτλο του ένα εικονίδιο ερωτηματικού. Περνώντας το δείκτη πάνω από το ερωτηματικό εμφανίζεται ένα μήνυμα που περιγράφει την λειτουργία και τις δυνατότητες τους εργαλείου.

Για παράδειγμα στην φόρμα προσθήκης ταινιών του «Διαχειριστή», που παρουσιάστηκε στην Εικόνα 4.2-1, δεν είναι σαφής ο τρόπος λειτουργίας ανάκτησης των πληροφοριών μίας ταινίας. Με την είσοδο του δείκτη στην περιοχή του ερωτηματικού, εμφανίζεται ανάλογο μήνυμα περιγράφοντας την διαδικασία λεπτομερώς όπως παρουσιάζεται παρακάτω στην Εικόνα 4.5-1.



Εικόνα 4.5-1, Βοηθητικό Μήνυμα

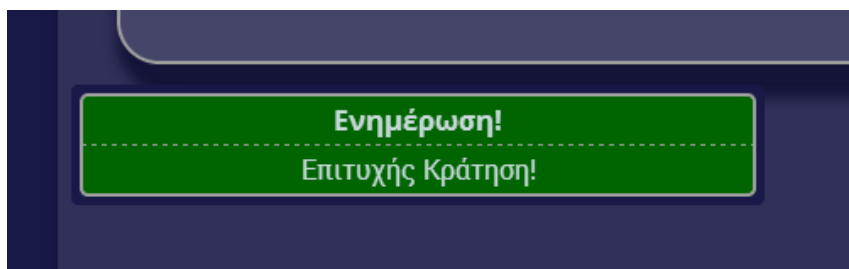
Με αυτόν τον τρόπο οποιαδήποτε πρώτη επαφή με τις λειτουργίες βοηθά τον χρήστη της ώστε να κατανοήσει τον τρόπο που λειτουργεί, ενώ παράλληλα δεν εμφανίζεται και άρα δεν ενοχλεί οπτικά τον χρήστη που γνωρίζει τις διαδικασίες.

### 4.5.2 Ενημερωτικά Μηνύματα

Προκειμένου να αποφευχθούν απλά μηνύματα ενημέρωσης, επιτυχούς καταχώρησης από φόρμες και σφάλματος, σταματώντας έτσι την ροή χρησιμοποίησης της εφαρμογής και για αισθητικούς λόγους, οι παραπάνω έξοδοι από αυτού του είδους τις φόρμες εμφανίζονται μετά από ανακατεύθυνση ως ειδοποίηση στο κάτω αριστερό μέρος της οθόνης.

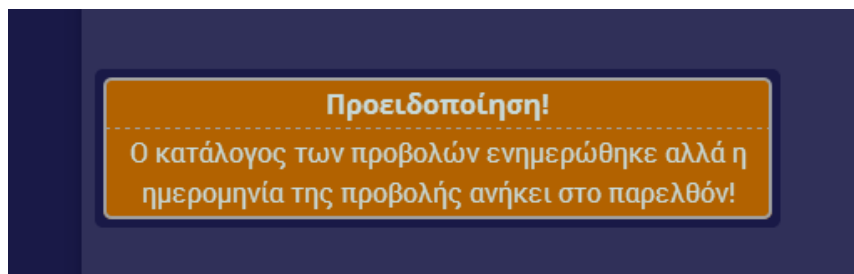
Τα μηνύματα αυτά χωρίζονται σε τρεις κατηγορίες: ενημερώσεις, σφάλματα και προειδοποιήσεις και χρωματίζονται ανάλογα με την ιδιότητά τους. Χάριν παραδείγματος μετά την επιτυχή καταχώρηση της φόρμας επιβεβαίωσης μίας κράτησης (Εικόνα 4.4-1), ο

χρήστης ανακατευθύνεται στην αρχική σελίδα της κράτησης και εμφανίζεται η αντίστοιχη ειδοποίηση.



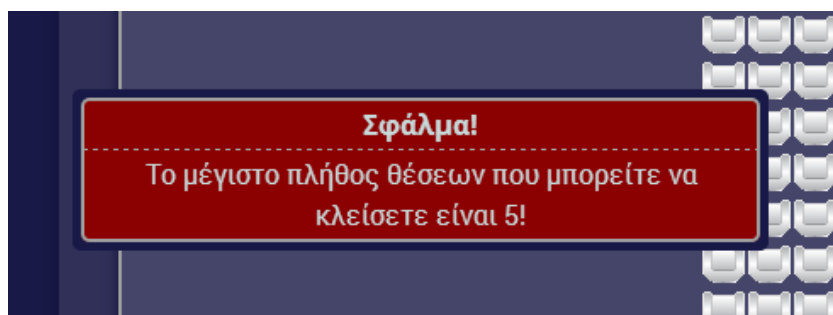
Εικόνα 4.5-2, Ειδοποίηση Ενημερωτικού Χαρακτήρα

Αντίστοιχα φέρνοντας ως παράδειγμα την προσθήκη μίας προβολής (Εικόνα 4.4-2) και αν υποθέσουμε ότι ο «Διαχειριστής» προσθέτει μία προβολή σε λάθος ημερομηνία που ανήκει στο παρελθόν λαμβάνει το παρακάτω μήνυμα.



Εικόνα 4.5-3, Ειδοποίηση Προειδοποιητικού Χαρακτήρα

Τέλος, αν υποθέσουμε ότι κάποιος πατήσει συνέχεια στο σύστημα κρατήσεων κατά την επιλογή θέσεων (Εικόνα 4.3-8) πάνω από 5 θέσεις ταυτόχρονα, ανακατευθύνεται ξανά στο σημείο επιλογής θέσεων και εμφανίζεται το παρακάτω σφάλμα.



Εικόνα 4.5-4, Ειδοποίηση Σφάλματος

## ΚΕΦΑΛΑΙΟ 5 ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Σε αυτό το κεφάλαιο θα παρουσιαστεί ο κώδικας που συντάχθηκε ώστε να λειτουργεί η εφαρμογή όπως παρουσιάστηκε. Λόγω της έκτασής της η ανάπτυξη του κώδικα έγινε με την μορφή βιβλιοθήκης συναρτήσεων. Ακολουθώντας αυτόν τον τρόπο ανάπτυξης επιτρέπεται να επαναχρησιμοποιηθεί ο κώδικας στα διαφορετικά σημεία της εφαρμογής που μπορεί να χρειαστεί αλλά και να αναλύεται κάθε πρόβλημα σε υπό-πρόβλήματα ώστε να είναι πιο εύκολη τόσο η σύνταξη όσο και μελλοντική συντήρηση που ενδέχεται να χρειαστεί.

Ξεκινώντας από το πρότυπο το οποίο ακολούθησαν όλες οι σελίδες της εφαρμογής χρησιμοποιήθηκε σταθερός κώδικας με την μορφή δύο εγγράφων `php`, το `header.php` και το `footer.php`. Ακόμη λόγω της πολυπλοκότητας του κώδικα προτιμήθηκε ο κατακερματισμός του σε υπό-πρόβλήματα για ευκολότερη λύση και η δημιουργία μίας βιβλιοθήκης συναρτήσεων `functions.php` όπου θα είναι διαθέσιμες όλες οι λειτουργίες. Στον κώδικα κάθε σελίδας του ιστότοπου όπως θα παρουσιαστεί παρακάτω εισάγεται στην αρχή και στο τέλος του οι σελίδες `header.php` και `footer.php` δημιουργώντας ένα είδος μοντέλου που θα ακολουθείται από την κάθε μία. Παράλληλα στο αρχείο `header.php` εισάγεται και το έγγραφο `functions.php` ώστε να έχει κάθε σελίδα πρόσβαση στις συναρτήσεις.

### 5.1 header.php

```
<!DOCTYPE HTML>
<?php include("functions.php"); ?>
<?php
session_start();
if(!isset($_SESSION['islogged']))$_SESSION['islogged']=0;
if(!isset($_SESSION['user']))$_SESSION['user']='user';
if(!isset($_SESSION['admin']))$_SESSION['admin']=0;
?>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <title>[CC][CinemaCied]</title>
    <link
href='http://fonts.googleapis.com/css?family=Open+Sans:400italic,700italic,400,300,700&subset=latin,greek' rel='stylesheet' type='text/css'>
    <link
href='http://fonts.googleapis.com/css?family=Roboto:400,700,400italic,700italic&subset=latin,greek' rel='stylesheet' type='text/css'>
    <link rel="stylesheet" type="text/css" href="css/googlefonts.css">
    <link rel="stylesheet" type="text/css" href="css/base.css">
    <link rel="stylesheet" type="text/css" href="css/tabs.css">
    <link rel="stylesheet" type="text/css" href="css/menu.css">
    <link rel="stylesheet" type="text/css" href="css/slider.css">
    <link rel="stylesheet" type="text/css" href="css/animate.css">
    <script language="javascript" type="text/javascript">
      function addEvent(obj, evType, fn){
        if (obj.addEventListener){
          obj.addEventListener(evType, fn, false);
          return true;
        } else if (obj.attachEvent){
          var r = obj.attachEvent("on"+evType, fn);
```



```

        return r;
    } else {
        return false;
    }
}

addEvent(window, 'load', initCheckboxes);

function initCheckboxes() {
    addEvent(document.getElementById('all'), 'click',
setCheckboxesAll);
    addEvent(document.getElementById('ok'), 'click',
setCheckboxesOk);
    addEvent(document.getElementById('nok'), 'click',
setCheckboxesNok);
}

function setCheckboxesAll() {
    var res =
document.getElementById('allform').getElementsByName('input');

    for (var i = 0; i < res.length; i++) {
        res[i].checked = document.getElementById('all').checked;
    }
}

function setCheckboxesOk() {
    var res =
document.getElementById('okform').getElementsByName('input');

    for (var i = 0; i < res.length; i++) {
        res[i].checked = document.getElementById('ok').checked;
    }
}

function setCheckboxesNok() {
    var res =
document.getElementById('nokform').getElementsByName('input');

    for (var i = 0; i < res.length; i++) {
        res[i].checked = document.getElementById('nok').checked;
    }
}
</script>
</head>
<body>
    <div id="top"></div>
    <div id="logo">
</div>
    <div class="base" id="navbar">
        <div id="menu">
            <table>
                <tr>
                    <td>
                        <a <?php if ($page == 'index.php') echo
'id="currentpage"'; ?> href="index.php" >
                            <h1>APXIKH</h1>
                        </a>
                    </td>

```

```

        <td>
            <a <?php if ($page == 'screenings.php') echo
'id="currentpage"'; ?> href="screenings.php" >
                <h1>ΠΡΟΒΟΛΕΣ</h1>
            </a>
        </td>
        <td>
            <a <?php if ($page == 'reservations.php') echo
'id="currentpage"'; ?> href="reservations.php" >
                <h1>ΚΡΑΤΗΣΗ</h1>
            </a>
        </td>
        <td>
            <a <?php if ($page == 'soon.php') echo
'id="currentpage"'; ?> href="soon.php" >
                <h1>ΠΡΟΣΕΧΩΣ</h1>
            </a>
        </td>
        <td>
            <a <?php if ($page == 'contact.php') echo
'id="currentpage"'; ?> href="contact.php" >
                <h1>ΕΠΙΚΟΙΝΩΝΙΑ</h1>
            </a>
        </td>
    </tr>
</table>
</div>
</div>
<div class="base" id="center">
<!-- closing in footer.php -->

```

## 5.2 footer.php

```

<!-- sidebar -->

<div id="info">
    <div id="login" class="box">
        <?php include("login.php"); ?>
    </div><br>
    <div class="box">
        
        
        <embed src="images/banner4.swf" quality="high" />
    </div>
</div>
<!-- closing center -->
</div>
<div class="base" id="footer">
    <p>Πτυχιακή Εργασία - C.I.E.D. - &copy; 2014 - Server Time:
    <?php
        echo date('d/m/Y h:i:s a', time());
    ?>
    </p>
</div>
</body>
</html>

```

## 5.3 login.php

```

<?php
if ($_SESSION['islogged'] == 0) {
    if (!isset($_POST['submit-login'])){
        //The HTML Login form
    }
    <form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="post">
        <p>
            Username: <input type="text" name="username" autofocus/>
            Password: <input type="password" name="password" />
            <input type="submit" name="submit-login" value="GO!" />

            <a <?php if ($page == 'register.php') { echo
'id="currentpage"'; } ?> href="register.php" ><strong>Γίνε μέλος</strong></a>
        </p>
    </form>

<?php
} else {
    $mysqli = dbconnect();
    $username = $_POST['username'];
    $password = $_POST['password'];
    $sql = "SELECT * from users WHERE username LIKE
'{$username}' AND password LIKE '{$password}' LIMIT 1";
    $result = $mysqli->query($sql);
    if (!$result->num_rows == 1) {
        echo notify("Σφάλμα!", "Το όνομα χρήστη ή ο κωδικός πρόσβασης
δεν ήταν σωστά! Προσπαθήστε ξανά.");
    } else {
        $_SESSION['islogged'] = 1;
        $_SESSION['user'] = $username;
        $row = mysqli_fetch_array($result);
        $_SESSION['admin'] = $row['isadmin'];
        header('Location: '. $_SERVER['PHP_SELF']);
    }
}
}
else{
    echo '
    <ul id="menulogin">
        <li>Καλωσόρισες, '. $_SESSION['user'] .'!</li><li>'.
showcredits($_SESSION['user']) .'</li>
        <li><a href="logout.php" >Αποσύνδεση</a></li>
        <li><a '
            if ($page == 'userprofile.php') echo 'id="currentpage"';
            echo 'href="userprofile.php" >Προφίλ</a></li>';
        if ( $_SESSION['admin'] > 90) {
            if ( $_SESSION['admin'] > 95) {
                echo ' <li>
                    <a '
                    if ($page == 'adminadd.php') echo
'id="currentpage"';
                    echo 'href="adminadd.php" >Προσθήκη</a>
                </li>';
            }
            echo ' <li>
                <a '
                if ($page == 'overview.php') echo
'id="currentpage"';
                echo 'href="overview.php" >Επισκόπηση</a>

```

```

        </li>';
    if ( $_SESSION['admin'] > 95) {
        echo ' <li>
            <a ' ;
            if ($page == 'halls.php') echo 'id="currentpage"';
            echo 'href="halls.php" >Αίθουσες</a>
        </li>';
        echo ' <li>
            <a ' ;
            if ($page == 'stats.php') echo 'id="currentpage"';
            echo 'href="stats.php" >Στατιστικά</a>
        </li>';
    }
    echo ' <li>
        <a ' ;
        if ($page == 'users.php') echo 'id="currentpage"';
        echo 'href="users.php" >Χρήστες</a>
    </li>';
    echo ' <li>
        <a ' ;
        if ($page == 'service.php') echo
'id="currentpage"';
        echo 'href="service.php" >Εξυπηρέτηση</a>
    </li>';
    }
    echo '</ul>';
}
?>

```

## 5.4 logout.php

```

<?php
    session_start();
    session_unset();
    header("Location: index.php");
?>

```

## 5.5 index.php

```

<?php $page = 'index.php'; ?>
<?php include("header.php"); ?>
<div id="content"><br>
    <div class="box" id="welcome">
        <span>
            Καλωσορίσατε στο [CC]CinemaCied!<br>
            Είναι μία ιστοσελίδα που υλοποιείται με PHP και MySQL<br>
            στα πλαίσια εκπόνησης της πτυχιακής εργασίας που μου έχει
            ανατεθεί με τίτλο<br>
            "Ανάπτυξη εφαρμογής κράτησης και αγοράς εισιτηρίου σε πολύ-
            κινηματογράφο".
        </span>
    </div>
    <div id="sliderwrapper">
    <div id="content-slider">
        <div id="slider">
            <div id="mask">
            <ul>

```



```

?>
    <div class="box">
        <h3>Τηλέφωνα Κρατήσεων</h3>
        <div>
            <form>
                <label>Σταθερό:</label>                <input type="text"
readonly value="26340 XXXXX"/><br>
                <label>Fax:</label>                    <input type="text"
readonly value="26340 XXXXX"/><br>
                <label>Cosmote - What's up</label> <input type="text"
readonly value="698 XXXXXXXX"/><br>
                <label>Vodafone - CU</label>         <input type="text"
readonly value="695 XXXXXXXX"/><br>
                <label>Wind - F2G</label>           <input type="text"
readonly value="693 XXXXXXXX"/><br>
                <label>Email:</label>                <input type="text"
readonly value="postmaster@example.com"/><br>
            </form>
        </div>
    </div><br>
    <div class="box">
        <h3>Οδηγίες <?php echo showhelp('Εισάγετε μία διεύθυνση για να
λάβετε οδηγίες!<br>π.χ. "Κορίνθου 33, Πάτρα"); ?></h3>
        <form action="<?php echo $_SERVER['PHP_SELF']; ?>"
method="post">
            <label>Διεύθυνση:</label><input type="text"
name="address"><br><br>
            <input type="submit" name="submit-directions"
value="Αναζήτηση" />
        </form>
    </div>
</div>
<?php include("footer.php"); ?>

```

## 5.9 reservations.php

```

<?php $page = 'reservations.php'; ?>
<?php include("header.php"); ?>
    <div id="content">
<?php    if ( $_SESSION['islogged'] == 1 ) {
        if ( isset($_POST['submit-respickmovie']) ) {
            $movie_id = $_POST['id'];
            echo respickdate($movie_id);
        }else if ( isset($_POST['submit-respickdate']) ) {
            $screen_id = $_POST['screen_id'];
            echo respickseat($screen_id);
        }else if ( isset($_POST['submit-respickscreen']) ) {
            $screen_id = $_POST['screen_id'];
            echo respickseat($screen_id);
        }else if ( isset($_POST['submit-respickseat']) ) {
            if ( !isset($_POST['seat']) ) {
                echo notify('Σφάλμα!', 'Δεν επιλέχθηκε καμία θέση!');
            }else if ( count($_POST['seat']) > 5 ) {
                echo notify('Σφάλμα!', 'Το μέγιστο πλήθος θέσεων που
μπορείτε να κλείσετε είναι 5!');
                echo respickseat($_POST['screen_id']);
            }else {
                echo resconfirm();
            }
        }
        }else if ( isset($_POST['submit-rescancel']) ) {

```



```

        $exists = 0;
        $result = $mysqli->query("SELECT username from users WHERE
username = '{$username}' LIMIT 1");
        if ($result->num_rows == 1) {
            $exists = 1;
            $result = $mysqli->query("SELECT email from users WHERE
email = '{$email}' LIMIT 1");
            if ($result->num_rows == 1) $exists = 2;
        } else {
            $result = $mysqli->query("SELECT email from users WHERE
email = '{$email}' LIMIT 1");
            if ($result->num_rows == 1) $exists = 3;
        }

        if ($exists == 1) echo "<p>Το όνομα χρήστη είναι ήδη
κατοχυρωμένο!</p>";
        else if ($exists == 2) echo "<p>Το όνομα χρήστη και το email
είναι ήδη κατοχυρωμένο!</p>";
        else if ($exists == 3) echo "<p>Το email είναι ήδη
κατοχυρωμένο!!</p>";
        else {
            $sql = "INSERT INTO `users` (`id`, `code`, `username`,
`password`, `fname`, `lname`, `birthdate`, `email`)
                VALUES (NULL, '{$code}', '{$username}',
 '{$password}', '{$first_name}', '{$last_name}', '{$birthdate}',
 '{$email}')";
            $result = dbq($sql);
            if ( !$result ) {
                echo "<p>MySQL error no {$mysqli->errno} : {$mysqli-
>error}</p>";
                exit();
            } else {
                echo "<p>Επιτυχής εγγραφή!</p>";
            }
        }
    }
}
?>
</div>
</div>
<?php include("footer.php"); ?>

```

## 5.11 userprofile.php

```

<?php $page = 'userprofile.php'; ?>
<?php include("header.php"); ?>
<div id="content">
<?php
if ( isset( $_POST['submit-cancelres'] ) ) {
    $mysqli = dbconnect();
    $res_ids = $_POST['tickets'];
    $first_flag = 1;
    $query = "DELETE FROM reservations WHERE";
    foreach ( $res_ids as $res_id ) {
        if ( $first_flag != 1 ) {
            $query .= " OR";
        }
        $query .= " id=". intval($res_id);
        $first_flag++;
    }
}

```



```

if ( $mysqli->query($query) ) {
    echo notify('Ενημερωση!', 'Η κρατήσεις που επιλέξατε ακυρώθηκαν!');
}else{
    echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με
τον διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli-
>error}");
}
}else if ( isset( $_POST['submit-payres'] ) ) {
    $mysqli      = dbconnect();
    $res_ids     = $_POST['tickets'];
    $user_id     = $_POST['user_id'];
    $num_of_tickets = count($res_ids);
    $user_credits = getcredits($user_id);
    if ( $_SESSION['admin'] > 90 || $num_of_tickets <= $user_credits ) {
        $first_flag = 1;
        $query      = "UPDATE reservations SET paid=1 WHERE";
        foreach ( $res_ids as $res_id ) {
            if ( $first_flag != 1 ) {
                $query .= " OR";
            }
            $query .= " id=". intval($res_id);
            $first_flag++;
        }
        if ( $mysqli->query($query) ) {
            echo notify('Ενημέρωση!', 'Η πληρωμή ήταν επιτυχής!');
            $credits = $num_of_tickets;
            echo spendcredits($credits);
        }else{
            echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε
με τον διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli-
>error}");
        }
    }else {
        echo notify('Σφάλμα!', 'Ο αριθμός των εισιτηρίων που επιλέξατε ήταν
μεγαλύτερος από τα credits που διαθέτεται!');
    }
}else if ( isset( $_POST['submit-changepassword'] ) ) {
    if ( isset($_POST['password-old']) && isset($_POST['password-new']) &&
isset($_POST['password-check']) ) {
        $user_id     = $_POST['user_id'];
        $old         = $_POST['password-old'];
        $new         = $_POST['password-new'];
        $check       = $_POST['password-check'];
        $result      = dbq("SELECT * FROM users WHERE id={$user_id}");
        $row         = mysqli_fetch_array($result);
        $current     = $row['password'];
        if ( $old != $current ) {
            echo notify('Προειδοποίηση!', 'Ο κωδικός δεν άλλαξε! Ο κωδικός
που πληκτρολογήσατε δεν είναι σωστός!');
        }else if ( $new != $check ) {
            echo notify('Προειδοποίηση!', 'Ο κωδικός δεν άλλαξε! Η
επιβεβαίωση του κωδικού δεν ήταν σωστή!');
        }else if ( $old == $current ) {
            echo notify('Προειδοποίηση!', 'Ο κωδικός δεν άλλαξε! Ο νέος
κωδικός είναι ίδιος με τον παλιό!');
        }else if ( trim($old) == '' || trim($current) == '' || trim($check)
== '' ) {
            echo notify('Προειδοποίηση!', 'Ο κωδικός δεν άλλαξε! Τα πεδία
δεν μπορεί να είναι κενά!');
        }
    }
}

```

```

        }else {
            $query = "UPDATE users SET password='{$new}' WHERE
id={$user_id}";
            $mysqli = dbconnect();
            if ( $mysqli->query($query) ) {
                echo notify('Ενημερωση!', 'Η αλλαγή του κωδικού ήταν
επιτυχής!');
            }else{
                echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε,
επικοινωνήστε με τον διαχειριστή!<br> MySQL error no {$mysqli->errno} :
{$mysqli->error}");
            }
        }
    }else {
        echo notify('Προειδοποίηση!', 'Ο κωδικός δεν άλλαξε! Δεν
συμπληρώθηκαν όλα τα πεδία!');
    }
}
if ( $_SESSION['islogged'] ) {
    echo showprofile($_SESSION['user']);
}else {
    echo '<p>Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη σελίδα.</p>';
}
?>
</div>
<?php include("footer.php"); ?>

```

## 5.12 adminadd.php

```

<?php $page = 'adminadd.php'; ?>
<?php include("header.php"); ?>
<div id="content">
<?php
    if( $_SESSION['admin'] > 95 ){
        addmovie();
        addscreenings();
    }
    else if ( $_SESSION['admin'] <= 95 ){
        echo "<p>Δεν διαθέτετε τα απαραίτητα δικαιώματα για την προβολή
αυτής της σελίδας!</p>";
    }else {
        echo '<p>Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη
σελίδα.</p>';
    }
?>
</div>
<?php include("footer.php"); ?>

```

## 5.13 overview.php

```

<?php $page = 'overview.php'; ?>
<?php include("header.php"); ?>
<div id="content">
<?php
    if ( isset($_POST['submit-editmovie']) ){
        $whereid = $_POST['id'];
        $title = mysql_real_escape_string($_POST['title']);
        $year = mysql_real_escape_string($_POST['year']);
        $runtime = mysql_real_escape_string($_POST['runtime']);
        $genre = mysql_real_escape_string($_POST['genre']);
    }
}

```

```

$imdbID      = mysql_real_escape_string($_POST['imdb_ID']);
$status      = mysql_real_escape_string($_POST['status']);
$plot       = mysql_real_escape_string($_POST['plot']);
$poster     = mysql_real_escape_string($_POST['poster-url']);
$filename    = $imdbID.'.jpg';
if ( $filename != $poster ) {
    $path     = "images/". $filename;
    file_put_contents($path, file_get_contents($poster));
    $poster  = $path;
}
$mysqli = dbconnect();
$query    = "UPDATE movies SET
            title   ='{$title}',
            year    ='{$year}',
            runtime ='{$runtime}',
            genre   ='{$genre}',
            plot    ='{$plot}',
            poster  ='{$poster}',
            status  ='{$status}',
            imdbID  ='{$imdbID}'
            WHERE id = '{$whereid}' ";
if ($mysqli->query($query)) {
    echo notify("Ενημέρωση!", "Ο κατάλογος των ταινιών ενημερώθηκε
επιτυχώς!");
} else {
    echo notify("Σφάλμα!", "Προέκυψε κάποιο σφάλμα, παρακαλούμε
επικοινωνήστε με τον διαχειριστή!<br> MySQL error no {$mysqli->errno} :
{$mysqli->error}");
    exit();
}
} else if ( isset($_POST['submit-editscreening']) ){
    $whereid      = intval($_POST['whereid']);
    $movie_id     = intval($_POST['movie_id']);
    $hall_id      = intval($_POST['hall_id']);
    $datetime_start = $_POST['datetime_start'];
    $datetime_start = strtotime($datetime_start);
    $datetime_start = date("Y-m-d H-i-s", $datetime_start);
    $mysqli       = dbconnect();
    $query        = "UPDATE screenings SET
                    id           = {$whereid},
                    movie_id     = {$movie_id},
                    hall_id      = {$hall_id},
                    datetime_start = '{$datetime_start}'
                    WHERE id = {$whereid}";
    if ($mysqli->query($query)) {
        echo notify("Ενημέρωση!", "Ο κατάλογος των προβολών ενημερώθηκε
επιτυχώς!");
    } else {
        echo notify("Σφάλμα!", "Προέκυψε κάποιο σφάλμα, παρακαλούμε
επικοινωνήστε με τον διαχειριστή!<br> MySQL error no {$mysqli->errno} :
{$mysqli->error}");
        exit();
    }
}
} else if ( isset($_POST['submit-addscreenasnew']) ){
    $movie_id     = intval($_POST['movie_id']);
    $hall_id      = intval($_POST['hall_id']);
    $datetime_start = $_POST['datetime_start'];
    $datetime_start = strtotime($datetime_start);
    $datetime_start = date("Y-m-d H-i-s", $datetime_start);

```

```

    $query          = "INSERT INTO `screenings` (`movie_id`, `hall_id`,
`datetime_start`)
                    VALUES ( {$movie_id}, {$hall_id},
'{$datetime_start}')";
    $mysqli = dbconnect();
    if ( $mysqli->query($query) ){
        echo notify("Ενημέρωση!", "Η προβολή προστέθηκε ως νέα!");
    }else{
        echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με
τον διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli-
>error}");
    }
}else if ( isset ( $_POST['submit-addscreenasnew'] ) ) {
    header("Location: overview.php");
}

if( $_SESSION['admin'] > 95 ){
    echo moviestable();
    echo adminscreencalendar();
}else if( $_SESSION['admin'] > 90 ){
    echo adminscreencalendar();
}else if ( $_SESSION['admin'] <= 90 ) {
    echo "<p>Δεν διαθέτετε τα απαραίτητα δικαιώματα για την προβολή αυτής
της σελίδας!</p>";
}else if ( !isset($_SESSION['admin']) ){
    echo '<p>Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη σελίδα.</p>';
}
?>
</div>
<?php include("footer.php"); ?>

```

## 5.14 halls.php

```

<?php $page = 'halls.php'; ?>
<?php include("header.php"); ?>
<div id="content">
<?php
    if ( $_SESSION['admin'] > 95 ){
        echo hallstable();
        echo edithall();
        echo addhall();
    }
    else if ( $_SESSION['admin'] <= 95 ) {
        echo "<p>Δεν διαθέτετε τα απαραίτητα δικαιώματα για την προβολή
αυτής της σελίδας!</p>";
    }else {
        echo '<p>Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη
σελίδα.</p>';
    }
?>
</div>
<?php include("footer.php"); ?>

```

## 5.15 stats.php

```

<?php $page = 'stats.php'; ?>
<?php include("header.php"); ?>
<div id="content">
<?php
    if ( isset($_POST['submit-searchincome']) ) {

```



```

        echo '<p>Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη σελίδα.</p>';
    }
?>
    </div>
<?php include("footer.php"); ?>

```

## 5.16 stats.php

```

<?php $page = 'users.php'; ?>
<?php include("header.php"); ?>
    <div id="content">
<?php
    if ( $_SESSION['admin'] > 90 ) {
        if ( isset($_POST['submit-searchbycode']) ) {
            $code         = $_POST['code'];
            $result        = searchuserbycode($code);
            $row           = mysqli_fetch_array($result);
            $username      = $row['username'];
            header("Location: adminusercheck.php?username=". $username);
        }else if ( isset($_POST['submit-searchbyusername']) ) {
            $username      = $_POST['username'];
            $result        = searchuserbyusername($username);
            if ( $result->num_rows >= 1 ) {
                if ( $result->num_rows == 1 ) {
                    $row           = mysqli_fetch_array($result);
                    $username      = $row['username'];
                    header("Location: adminusercheck.php?username="
$username);
                }else {
                    echo ' <div class="box">
                        <h3>
                            Αποτελέσματα αναζήτησης
                            '. showhelp('Πατήστε πάνω στο όνομα χρήστη για
να δείτε το προφίλ του!') .'
                        </h3>
                        <table class="admintable">
                        <tr class="adminth">
                            <th>Όνομα Χρήστη</th>
                            <th>Αναγνωριστικό</th>
                            <th>Όνοματεπώνυμο</th>
                            <th>email</th>
                            <th>credits</th>
                            <th>Δικαιώματα</th>
                        </tr>';
                    while ( $row = mysqli_fetch_array($result) ) {
                        $username      = $row['username'];
                        $code         = $row['code'];
                        $firstname    = $row['fname'];
                        $lastname     = $row['lname'];
                        $fullname     = $firstname.' '.$lastname;
                        $email        = $row['email'];
                        $credits      = $row['credits'];
                        $permissions = $row['isadmin'];
                        if ( $permissions == 0 ) {
                            $permissions = 'Χρήστης';
                        }else if ( $permissions == 95 ) {
                            $permissions = 'Υπάλληλος';
                        }else if ( $permissions == 99 ) {
                            $permissions = 'Διαχειριστής';
                        }
                    }
                }
            }
        }
    }

```

```

        echo ' <tr class="adminrow">
                <td><a href="adminusercheck.php?username=' .
$username . "'>
                    ' . $username      .'</a></td>
                <td>' . $code          .'</td>
                <td>' . $fullname     .'</td>
                <td>' . $email        .'</td>
                <td>' . $credits      .'</td>
                <td>' . $permissions  .'</td>
            </tr>';
        }
        echo ' </table></div><br>';
    }
    }else {
        echo notify('Σφάλμα!', 'Δεν βρέθηκε χρήστης με όνομα χρήστη: ' .
$username);
    }
    }
    echo showsearchuserbycode();
    echo showsearchuserbyusername();
}else if ( $_SESSION['admin'] < 90 ) {
    echo "<p>Δεν διαθέτετε τα απαραίτητα δικαιώματα για την προβολή αυτής
της σελίδας!</p>";
}else {
    echo '<p>Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη σελίδα.</p>';
}
?>
</div>
<?php include("footer.php"); ?>

```

## 5.17 service.php

```

<?php $page = 'service.php'; ?>
<?php include("header.php"); ?>
    <div id="content">
<?php
    if ( isset( $_POST['submit-searchres'] ) ) {
        $search = $_POST['search'];
        echo showsearchres($search);
    }else if ( isset( $_POST['submit-searchticket'] ) ) {
        $code = $_POST['code'];
        echo showsearchticket($code);
    }else if ( isset( $_POST['submit-cancelres'] ) ) {
        $mysqli = dbconnect();
        $res_ids = $_POST['tickets'];
        $first_flag = 1;
        $query = "DELETE FROM reservations WHERE";
        foreach ( $res_ids as $res_id ) {
            if ( $first_flag != 1 ) {
                $query .= " OR";
            }
            $query .= " id=" . intval($res_id);
            $first_flag++;
        }
        if ( $mysqli->query($query) ) {
            echo notify('Ενημέρωση!', 'Η κρατήσεως που επιλέξατε ακυρώθηκαν!');
        }else{
            echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με
τον διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli->error}");
        }
    }
}

```

```

    }
} else if ( isset( $_POST['submit-payres'] ) ) {
    $mysqli = dbconnect();
    $res_ids = $_POST['tickets'];
    $user_id = $_POST['user_id'];
    $num_of_tickets = count($res_ids);
    $user_credits = getcredits($user_id);
    if ( $num_of_tickets <= $user_credits ) {
        $first_flag = 1;
        $query = "UPDATE reservations SET paid=1 WHERE";
        foreach ( $res_ids as $res_id ) {
            if ( $first_flag != 1 ) {
                $query .= " OR";
            }
            $query .= " id=". intval($res_id);
            $first_flag++;
        }
        if ( $mysqli->query($query) ) {
            echo notify('Ενημερωση!', 'Η πληρωμή ήταν επιτυχής!');
            $credits = $num_of_tickets;
            echo spendcredits($credits);
        } else {
            echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με τον διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli->error}");
        }
    } else {
        echo notify('Σφάλμα!', 'Ο αριθμός των εισιτηρίων που επιλέξατε ήταν μεγαλύτερος από τα credits που διαθέτεται!');
    }
}
if ( $_SESSION['admin'] > 90 ) {
    echo searchres();
    echo searchticket();
} else if ( $_SESSION['admin'] < 90 ) {
    echo "<p>Δεν διαθέτετε τα απαραίτητα δικαιώματα για την προβολή αυτής της σελίδας!</p>";
} else {
    echo '<p>Παρακαλώ συνδεθείτε για να προβάλλετε αυτή τη σελίδα.</p>';
}
?>
</div>
<?php include("footer.php"); ?>

```



## 5.18 buycredits.php

```

<?php $page = ''; ?>
<?php include("header.php"); ?>
    <div id="content">

<?php
    if ( $_SESSION['islogged'] == 1 ){
        $price = 5;
        if ( !isset($_POST['submit-addcredits']) ) {
            echo '<div class="box">
                <h3 class="yellow"> ΠΡΟΣΟΧΗ!!! ΓΙΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥΣ
ΛΟΓΟΥΣ!</h3><h3>Αγορά Credits </h3><br>
                <p>Τιμή credit '. $price .' €</p><br>
                <form action="'. $_SERVER['PHP_SELF'] .'

```



```

WHERE reservations.screening_id = {$screen_id} AND
reservations.confirmed = 1
ORDER BY res_seat ASC";
$result = dbq($query);
while ( $row = mysqli_fetch_array($result) ){
    $res_id      = $row['res_id'];
    $res_code    = $row['res_code'];
    $res_paid    = $row['res_paid'];
    $res_seat    = $row['res_seat'];
    $movie_title = $row['movie_title'];
    $startingdate = $row['startingdate'];
    $date        = date("d.m.Y", strtotime($startingdate));
    $startingtime = $row['startingtime'];
    $time        = date("H:i", strtotime($startingtime));
    $hall        = $row['hall_id'];
    $title       = $movie_title .' - '. $date .' - '. $time .' -
Αίθουσα: '. $hall;
    if ( $res_paid == 0 ) {
        $nokc++;
        $status = 'N.OK';
        $class  = 'nok';
        $nokbody .= ' <tr class="adminrow '. $class .' ">
class="ticket" name="tickets[]" value="'. $res_id .' "></td>
        <td>'. $res_code .'</td>
        <td>'. $status  .'</td>
        <td>'. $res_seat .'</td>
        </tr>';
    }
    if ( $res_paid == 1 ) {
        $okc++;
        $status = 'OK';
        $class  = 'ok';
        $okbody  .= ' <tr class="adminrow '. $class .' ">
class="ticket" name="tickets[]" value="'. $res_id .' "></td>
        <td>'. $res_code .'</td>
        <td>'. $status  .'</td>
        <td>'. $res_seat .'</td>
        </tr>';
    }
    $allc++;
    $allbody  .= ' <tr class="adminrow '. $class .' ">
class="ticket" name="tickets[]" value="'. $res_id .' "></td>
        <td>'. $res_code .'</td>
        <td>'. $status  .'</td>
        <td>'. $res_seat .'</td>
        </tr>';
}
    $allhead = ' <table class="admintable animated fadeInLeft"
id="reservations">
        <tr class="adminrow">
            <th><input type="checkbox" name="cb"
value="1" id="all" /></th>
            <th>Κωδικός</th>
            <th>Κατάσταση</th>
            <th>Θέση</th>
        </tr>';

```

```

        $okhead          = '<table class="admintable animated fadeInLeft"
id="reservations">
                                <tr class="adminrow">
                                <th><input type="checkbox" name="cb"
value="1" id="ok" /></th>
                                <th>Κωδικός</th>
                                <th>Κατάσταση</th>
                                <th>Θέση</th>
                                </tr>';
        $nokhead         = '<table class="admintable animated fadeInLeft"
id="reservations">
                                <tr class="adminrow">
                                <th><input type="checkbox" name="cb"
value="1" id="nok" /></th>
                                <th>Κωδικός</th>
                                <th>Κατάσταση</th>
                                <th>Θέση</th>
                                </tr>';
        $tablefooter = '</table>';
        echo ' <div class="box"><h3>'. $title .'</h3>
        <div class="tabs">
        <div class="tab">
        <input type="radio" id="tab-1" name="tab-group-
1" checked>
        <label for="tab-1" class="tablabel">('.$allc
        .')<br>Όλες</label>
        <div class="tabcontent">
        <form id="allform" action="'.
$_SERVER['PHP_SELF']. '" method="post">
        <input type="submit" name="submit-
rescancel" value="Ακύρωση Επιλεγμένων">
        <input type="hidden" name="screen_id"
value="'. $screen_id .' ">
        '. $allhead.$allbody.$tablefooter .'
        </form>
        </div>
        </div>
        <div class="tab">
        <input type="radio" id="tab-2" name="tab-group-
1">
        <label for="tab-2" class="tablabel">('.$okc
        .')<br>Πληρωμένες</label>
        <div class="tabcontent">
        <form id="okform" action="'.
$_SERVER['PHP_SELF']. '" method="post">
        <input type="submit" name="submit-
rescancel" value="Ακύρωση Επιλεγμένων">
        <input type="hidden" name="screen_id"
value="'. $screen_id .' ">
        '. $okhead.$okbody.$tablefooter .'
        </form>
        </div>
        </div>
        <div class="tab">
        <input type="radio" id="tab-3" name="tab-group-
1">

```

```

        <label for="tab-3" class="tablabel">(' . $nokc
. ')<br>Εκκρεμεί</label>
        <div class="tabcontent">
            <form id="nokform" action="'.
$_SERVER['PHP_SELF']. '" method="post">
                <input type="submit" name="submit-
rescancel" value="Ακύρωση Επιλεγμένων">
                <input type="hidden" name="screen_id"
value="'. $screen_id .'">
                ' . $nokhead.$nokbody.$tablefooter .'
            </form>
        </div>
    </div>
</div>
</div>';
    }else if ( $_SESSION['admin'] < 90 ) {
        echo "<p>Δεν διαθέτετε τα απαραίτητα δικαιώματα για την προβολή
αυτής της σελίδας!</p>";
    }else {
        echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με
τον διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli->error}");
    }
?>
    </div>
<?php include("footer.php"); ?>

```

## 5.20 printclubcard.php

```

<!DOCTYPE HTML>
<?php include("functions.php"); ?>
<?php
session_start();
if(!isset($_SESSION['islogged']))$_SESSION['islogged']=0;
if(!isset($_SESSION['user']))$_SESSION['user']='user';
if(!isset($_SESSION['admin']))$_SESSION['admin']=0;
?>
<html>
    <head>
        <meta charset="utf-8">
        <title>[CC][CinemaCied]</title>
        <link rel="stylesheet" type="text/css" href="css/clubcard.css">
        <link
href='http://fonts.googleapis.com/css?family=Open+Sans:400italic,700italic,400,
300,700&amp;subset=latin,greek' rel='stylesheet' type='text/css'>
        <link
href='http://fonts.googleapis.com/css?family=Roboto:400,700,400italic,700italic
&amp;subset=latin,greek' rel='stylesheet' type='text/css'>
    </head>
    <body>
<?php
    if ( isset( $_POST['code'] ) && isset( $_POST['firstname'] ) && isset(
$_POST['lastname'] ) && $_SESSION['admin'] > 90 ) {
        $code      = $_POST['code'];
        $firstname  = $_POST['firstname'];
        $lastname   = $_POST['lastname'];
        $query      = "UPDATE users SET `clubcard` = `clubcard` + 1 WHERE
code='{ $code }' LIMIT 1";
        $result     = dbq($query);
        $fullname   = $firstname.' '.$lastname;
        echo ' <div id="card">

```

```

        <div id="logo">
        </div>
        <div id="qr">
            
        </div>
        <div id="fullname">
            <span>'. $fullname .'</span>
        </div>
    </div>';
}
?>

</body>
</html>

```

## 5.21 dbcon.php

```

<?php
const DB_HOST = 'localhost';
const DB_USER = 'root';
const DB_PASS = 'ptixiaki';
const DB_NAME = 'cinema';
?>

```

## 5.22 functions.php

```

<?php
//Καθορισμός Ζώνης Ωρας
date_default_timezone_set('Europe/Athens');

ob_start();

//Στοιχεία Σύνδεσης
require_once("dbcon.php");

//Σύνδεση στη Βάση
function dbconnect(){
    $mysqli = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);
    mysqli_set_charset($mysqli, "utf8");
    //Έλεγχος για σφάλματα σύνδεσης
    if ($mysqli->connect_errno) {
        echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με τον
διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli->error}");
        exit();
    }
    return $mysqli;
}

//Εκτέλεση Ερωτήματος
function dbq($query) {
    //
    $mysqli = dbconnect();
    if ( $result = $mysqli->query($query) ) {
        return $result;
        mysqli_close($mysqli);
    }else {
        echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με τον
διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli->error}");
    }
}

```

```

        exit();
    }
}

//Γεννήτρια μοναδικού αλφαριθμητικού κωδικού
function gencode($table, $num_of_chars) {
    $code = '';
    $characters = 'abcdefghijklmnopqrstuvwxyz0123456789';
    for ( $i = 0 ; $i < $num_of_chars; $i++ ) {
        $code .= $characters[rand(0, strlen($characters) - 1)];
    }
    $mysqli = dbconnect();
    $query = "SELECT code FROM {$table} WHERE code='{$code}'";
    if ( $mysqli->query($query) ) {
        $code = gencode($table, $num_of_chars);
    }else {
        return $code;
    }
}

//Μετατροπή Ημερών στα ελληνικά
function convertdays($day){
    switch($day){
        case "Mon": $greekday = 'Δευτέρα'; break;
        case "Tue": $greekday = 'Τρίτη'; break;
        case "Wed": $greekday = 'Τετάρτη'; break;
        case "Thu": $greekday = 'Πέμπτη'; break;
        case "Fri": $greekday = 'Παρασκευή'; break;
        case "Sat": $greekday = 'Σάββατο'; break;
        case "Sun": $greekday = 'Κυριακή'; break;
    }
    return $greekday;
}

//Ανάκτηση Πόντων Χρήστη
function getcredits($user_id){
    if ( getuserpermissions($user_id) > 90 ) {
        $credits = 100;
    }else {
        $result = dbq("SELECT credits FROM users WHERE id={$user_id}");
        $row = mysqli_fetch_array($result);
        $credits = intval($row['credits']);
    }
    return $credits;
}

//Εμφάνιση Πόντων
function showcredits($username) {
    $ret = '';
    $user_id = getuserid($username);
    $isadmin = getuserpermissions($user_id);
    if ( $isadmin > 90 ) {
        $ret .= 'Credits: <span style="font-size:20px;"
class="yellow">(&#8734;)</span>';
    }else {
        $ret .= '<a href="buycredits.php">Credits: <span class="yellow">(' .
getcredits($user_id) . ')</span></a>';
    }
    return $ret;
}

```

```

}

//Αφαίρεση Πόντων
function spendcredits($credits) {
    $title = 'Ενημέρωση!';
    $msg = '';
    if ($_SESSION['admin'] < 90) {
        $mysqli = dbconnect();
        $username = $_SESSION['user'];
        $result = $mysqli->query("SELECT * FROM users WHERE
`username`='{ $username}' LIMIT 1");
        $row = mysqli_fetch_array($result);
        $hascredits = $row['credits'];
        $credits = intval($credits);
        if ( $hascredits >= $credits ) {
            $query = "UPDATE users SET `credits`=`credits`-{$credits}
WHERE `username`='{ $username}' ";
            if ($mysqli->query($query)) {
                $msg .= 'Αφαιρέθηκαν ' . $credits . ' credits από τον
λογαριασμό σας!';
            }else{
                $title = 'Σφάλμα!';
                $msg .= 'Κάποιο σφάλμα προέκυψε, επικοινωνήστε με τον
διαχειριστή!';
            }
        }else {
            $msg .= 'Προειδοποίηση! Έγινε κράτηση αλλά όχι πληρωμή, δεν
διαθέτετε αρκετά credits';
        }
    }else {
        $mysqli = dbconnect();
        $amount = $credits * 5;
        $user_id = getuserid($_SESSION['user']);
        $query = "INSERT INTO `transactions` (`user_id`, `amount`) VALUES
( { $user_id}, { $amount} ) ";
        if ( $mysqli->query($query) ) {
            $title = 'Ενημέρωση!';
            $msg .= 'Έκδοση επιτυχής!';
        }else {
            $title = 'Σφάλμα!';
            $msg .= 'Κάποιο σφάλμα προέκυψε, επικοινωνήστε με τον
διαχειριστή!';
        }
    }
    $ret = notify($title, $msg);
    return $ret;
}

//Φόρμα Προσθήκης Ταινίας
function addmovie(){
    if (!isset($_POST['submit-addmovie']) && !isset($_POST['submit-genmovie']))
    {
        echo '
<div class="box">
    <h3>Προσθήκη Ταινίας
        ' . showhelp('Για να προσθέσετε μία ταινία συμπληρώστε τον κωδικό
IMDB <br>π.χ. "tt1877832" και πάτησε Generate.
            Αφού βρεθούν όλα τα στοιχεία αυτόματα μπορείτε να τα
επεξεργαστείτε και μετά να πατήσετε αποθήκευση!') .'
    
```



```

</h3>
<form action="'. $_SERVER['PHP_SELF'] .'

```

```

        <label>Πλοκή: </label>
        <textarea rows="4" cols="50" name="plot">'. $data['Plot']
    .</textarea><br>
        <input type="submit" name="submit-genmovie" value="Generate" />
        <input type="submit" name="submit-addmovie" value="Αποθήκευση"
    autofocus/>
    </form>
</div><br>';
}else {

    $mysqli = dbconnect();

    // Προετοιμασία Δεδομένων
    $title    = mysql_real_escape_string($_POST['title']);
    $year     = $_POST['year'];
    $runtime  = $_POST['runtime'];
    $genre    = $_POST['genre'];
    $url      = $_POST['poster-url'];
    $imdbID   = $_POST['imdbID'];
    $status   = $_POST['status'];
    $plot     = mysql_real_escape_string($_POST['plot']);

    $result = $mysqli->query("SELECT title from movies WHERE title =
'{$title}' LIMIT 1");
    if ($result->num_rows == 1) echo notify("Σφάλμα!", "Αυτή η ταινία
υπάρχει ήδη!");
    else {
        // Δημιουργία Εικόνας
        $filename = $imdbID.".jpg";
        $path = "images/". $filename;
        file_put_contents($path, file_get_contents($url));

        $sql = "INSERT INTO `movies` (`imdbID`, `title`, `year`,
`runtime`, `genre`, `plot`, `poster`, `status`)
VALUES ('{$imdbID}', '{$title}', '{$year}', '{$runtime}',
'{$genre}', '{$plot}', '{$path}', '{$status}')";

        if ($mysqli->query($sql)) {
            echo notify("Ενημέρωση!", "Ο κατάλογος των ταινιών ενημερώθηκε
επιτυχώς!");
        } else {
            echo notify("Σφάλμα!", "MySQL error no {$mysqli->errno} :
{$mysqli->error}");
            exit();
        }
    }
}
}

//Φόρμα Προσθήκης Προβολής
function addscreenings() {
    if ( !isset($_POST['submit-addscreening']) ) {
        $query1 = "SELECT `id`, `title` FROM MOVIES WHERE status = 2";
        $query2 = "SELECT `id` FROM `halls`";
        $result1 = dbq($query1);
        $result2 = dbq($query2);
        echo '<div class="box"><h3>Προσθήκη Προβολής
            . showhelp('Για να προσθέσετε μία προβολή
    συμπληρώστε τα στοιχεία και πάτησε αποθήκευση!') .'

```

```

        </h3>';
    echo '<form action="'. $SERVER['PHP_SELF']. '" method="post"
id="addscreening">';
    echo '<label>Τίτλος Ταινίας: </label><select name="movie_id"
value="0">';
    while($row = mysqli_fetch_array($result1)) {
        echo '<option value="'. $row['id'] . '">'. $row['title']
.'</option>';
    }
    echo '</select><br>';
    echo '<label>Αίθουσα: </label><select name="hall_id" value="0">';
    while($row = mysqli_fetch_array($result2)) {
        echo '<option value="'. $row['id'] . '">'. $row['id'] .'</option>';
    }
    echo '
</select><br>
<label>Ημερομηνία/ώρα: </label><input type="datetime-local"
name="datetime_start"><br><br>
<input type="submit" name="submit-addscreening" value="Αποθήκευση">
</form></div><br>
';
}else{
    $mysqli = dbconnect();
    $movie_id = $_POST['movie_id'];
    $movie = intval($movie_id);
    $hall_id = $_POST['hall_id'];
    $hall = intval($hall_id);
    $screening = $_POST['datetime_start'];
    $today_check = strtotime($screening);
    $screening = strtotime($screening);
    $screening = date("Y-m-d H-i-s", $screening);
    $result3 = $mysqli->query("SELECT `movie_id`, `hall_id`,
`datetime_start` from screenings
WHERE movie_id = '{$movie}' AND
hall_id = '{$hall}' AND
datetime_start = '{$screening}' LIMIT 1");
    if ($result3->num_rows == 1){
        $query = "UPDATE screenings SET cancelled = 0
WHERE movie_id = '{$movie}' AND
hall_id = '{$hall}' AND
datetime_start = '{$screening}'";
        if ( dbq($query) ){
            echo notify("Ενημέρωση!", "Η Προβολή προστέθηκε επιτυχώς");
        }
    }else{
        $sql = "INSERT INTO `screenings` (`movie_id`, `hall_id`,
`datetime_start`)
VALUES ('{$movie}', '{$hall}', '{$screening}')";
        if ($mysqli->query($sql)) {
            if ( $today_check < time() ){
                echo notify("Προειδοποίηση!", "Ο κατάλογος των προβολών
ενημερώθηκε αλλά η ημερομηνία της προβολής ανήκει στο παρελθόν!");
            }else{
                echo notify("Ενημέρωση!", "Ο κατάλογος των προβολών
ενημερώθηκε επιτυχώς!");
            }
        } else {
            echo notify("Σφάλμα!", "MySQL error no {$mysqli->errno} :
{$mysqli->error}");
        }
    }
}

```

```

        exit();
    }
}
}

//Φόρμα Προσθήκης Αίθουσας
function addhall() {
    if ( !isset($_POST['submit-addhall']) ) {
        $mysqli = dbconnect();
        $result = mysqli_query($mysqli, "SELECT count(*) as total from halls");
        $data = mysqli_fetch_assoc($result);
        $halls_now = intval($data['total']);
        $hall_next = $halls_now + 1 ;
        echo '
            <div class="box">
            <h3>Προσθήκη Αίθουσας '. showhelp("Για να προσθέσετε μία αίθουσα
            συμπληρώστε την χωρητικότητα και πατήστε αποθήκευση!") .'</h3>
            <form action=".'. $_SERVER['PHP_SELF'].'" method="post" id="addhall">
            <label>Νέα Αίθουσα: </label><input type="number" name="id"
            value=".'. $hall_next .' " readonly /><br>
            <label>Χωρητικότητα: </label><input type="number" name="capacity"
            min="0" max="1000" /><br><br>
            <input type="submit" name="submit-addhall" value="Αποθήκευση" />
            </form>
            </div><br>';
        }else{
            $mysqli = dbconnect();
            $id = $_POST['id'];
            $capacity = $_POST['capacity'];
            $query = "INSERT INTO `halls` (`id`, `capacity`)
                VALUES ('{$id}', '{$capacity}')";
            if ($mysqli->query($query)) {
                echo notify('Ενημέρωση!', 'Ο κατάλογος με τις αίθουσες
                ενημερώθηκε!');
            } else {
                echo notify('Σφάλμα!', "MySQL error no {$mysqli->errno} : {$mysqli-
                >error}");
            }
        }
    }
}

//Φόρμα Επεξεργασίας Ταινίας
function editmovie($id) {
    $whereid = $id;
    $mysqli = dbconnect();
    $result = dbq("SELECT * from movies WHERE `id` = ". $whereid);
    while($row = mysqli_fetch_array($result)) {
        $id = $row['id'];
        $title = $row['title'];
        $year = $row['year'];
        $posterpath = $row['poster'];
        $imdbID = $row['imdbID'];
        $genre = $row['genre'];
        $runtime = $row['runtime'];
        $plot = $row['plot'];
        $status = $row['status'];
        echo '
            <div id="editpopup" class="box animated zoomIn">

```

```

    <form action="'. $_SERVER['PHP_SELF'] .'

```

```

$ret = '';
$query = "SELECT DATE(datetime_start) as startingdate, cancelled FROM
`screenings` WHERE cancelled = 0 GROUP BY startingdate";
$result = dbq($query);
$ret .= '<div class="box"><h3>Διάλεξε Ημερομηνία</h3><ul
class="accordion">
    <form action="'. $_SERVER['PHP_SELF'] .'

```

```

    }
    return $ret;
}

//Φόρμα Επιλογής Προβολής αφού επιλέχθηκε ταινία
function respickdate($movie_id){
    $ret = '';
    $query = 'SELECT * FROM (
                SELECT id AS screen_id,
                       date(datetime_start) AS startingdate,
                       time(datetime_start) AS startingtime
                FROM screenings WHERE `movie_id`=' . $movie_id . '
            ) AS dates
            ORDER BY startingdate ASC, startingtime ASC';
    $result = dbq($query);
    $ret .= '<div class="box"><h3>Διάλεξε Προβολή</h3><br>';
    $ret .= '<form action="'. $_SERVER['PHP_SELF'] .' " method="post">';
    while ( $row = mysqli_fetch_array($result) ) {
        $screen_id = $row['screen_id'];
        $date = $row['startingdate'];
        $time = $row['startingtime'];
        $date = date("d-m-Y", strtotime($date));
        $time = date("H:i", strtotime($time));
        $show = $date . ', στις ' . $time;
        $ret .= '<label>' . $show . '</label><input type="radio" name="screen_id"
            value="'. $screen_id . '"><br><br>';
    }
    $ret .= '</select>';
    $ret .= '<input type="submit" name="submit-respickdate"
        value="Συνέχεια">';
    $ret .= '</form></div><br>';
    return $ret;
}

//Σύστημα Επιλογής Θέσης
function respickseat($screen) {
    $screen_id = $screen;
    $ret = '';
    $query1 = 'SELECT * FROM screenings WHERE `id`=' . $screen_id . '
LIMIT 1';
    $result1 = dbq($query1);
    $row = mysqli_fetch_array($result1);
    $movie_id = $row['movie_id'];
    $hall_id = $row['hall_id'];
    $datetime_start = $row['datetime_start'];
    $query2 = 'SELECT * FROM halls WHERE `id`=' . $hall_id . ' LIMIT 1';
    $result2 = dbq($query2);
    $row = mysqli_fetch_array($result2);
    $hall_capacity = $row['capacity'];
    $date = date("d-m-Y", strtotime($datetime_start));
    $time = date("H:i", strtotime($datetime_start));
    $query3 = 'SELECT * FROM reservations WHERE `screening_id`=' .
        $screen_id . ' ORDER BY `seat`';
    $result3 = dbq($query3);
    $taken_seat[] = array();
    while ( $row = mysqli_fetch_array($result3) ) {
        $seat = $row['seat'];
        $taken_seat[$seat] = $seat ;
    }
}

```

```

$seats_avail    = $hall_capacity - count($taken_seat) + 1;
$query4         = 'SELECT title FROM movies WHERE id='. $movie_id .' LIMIT
1';
$result4       = dbq($query4);
$row           = mysqli_fetch_array($result4);
$title        = $row['title'];
$ret          = '<div class="box"><h3>Επέλεξε Θέσεις</h3><br>';
$ret          = '<h3>Διαθέσιμες θέσεις <span class="yellow">'.
    $seats_avail . '</span> από '. $hall_capacity . '</h3><br><br>';
if ($seats_avail == 0) {
    $ret       = '<p>Λυπούμαστε, δεν υπάρχουν διαθέσιμες θέσεις για αυτή
τη προβολή.</p>';
} else {
    $ret       = '<form name="pickseat" action="'. $_SERVER['PHP_SELF']
.' " method="post" id="pickseat">';
    $ret       = '<input type="hidden" name="screen_id" value="'.
    $screen_id .' ">';
    $ret       = '<input type="hidden" name="title" value="'. $title
.' ">';
    $ret       = '<input type="hidden" name="startingdate" value="'. $date
.' ">';
    $ret       = '<input type="hidden" name="startingtime" value="'. $time
.' ">';
    $ret       = '<input type="hidden" name="hall" value="'. $hall_id
.' ">';
    $ret       = '<input type="submit" name="submit-respickseat"
value="Συνέχεια"><br><br>';
    $ret       = '<table id="seats">';
    $ret       = '<tr><th colspan="20">000NH</th></tr><tr>' ;
    for ($seat_num = 1 ; $seat_num <= $hall_capacity ; $seat_num++) {
        $flag = $seat_num - 1;
        $ret   = '<td ' ;
        if ($seat_num % 10 == 5) {
            $ret = 'class="hallway"';
        }
        $ret   = '><input type="checkbox" id="seat'. $seat_num .' "
name="seat[]" value="'. $seat_num .' "' ;
        if (isset( $taken_seat[$seat_num] )) {
            $ret = 'disabled=""';
        }
        $ret   = '</td><label for="seat'. $seat_num .' " alt="'. $seat_num .' "
title="'. $seat_num .' "></label></td>';
        if ( $seat_num % 20 == 0 ) {
            $ret = '</tr><tr>';
        }
    }
    $ret       = '</tr></table></form><br><br>';
}
$ret          = '</div><br>';
return $ret;
}

//Φόρμα επιβεβαίωσης κράτησης
function resconfirm() {
    $ret       = '';
    $user_id   = getuserid($_SESSION['user']);
    $screen_id = $_POST['screen_id'];
    $title     = $_POST['title'];
    $date      = $_POST['startingdate'];

```



```

$time      = $_POST['startingtime'];
$hall      = $_POST['hall'];
$seats     = $_POST['seat'];
$ret .= '
<div class="box">
  <h3>Επιβεβαίωση Κράτησης</h3>
  <form action="'. $_SERVER['PHP_SELF'] .'

```

```

}
$query = "INSERT INTO `reservations` (`code`, `screening_id`, `user_id`,
`seat`, `confirmed`, `paid`, `notes`)
VALUES ";
for ( $i=0 ; $i<$lastseat ; $i++) {
    $code      = gencode('reservations', 10);
    $codes[$i] = $code;
    $seat      = intval($seats[$i]);
    $query     .= "('{ $code}', { $screening_id}, { $user_id}, { $seat},
{ $confirmed}, { $paid}, '{ $notes}' )";
    if ( $i != $lastseat - 1 ) {
        $query .= ",";
    }
}
if ( $mysqli->query($query) ) {
    if ( $paid == 1 ) {
        $credits = $lastseat;
        echo spendcredits($credits);
    }
}
else{
    $title = 'Σφάλμα!';
    $msg   = "Κάποιο σφάλμα προέκυψε, επικοινωνήστε με τον
αχειριστή!<br>MySQL error no {$mysqli->errno} : {$mysqli->error}";
}
return notify($title, $msg);
}

```

*//Επεξεργασία Προβολής*

```

function editscreening($id) {
    $whereid = $id;
    $mysqli  = dbconnect();
    $query1  = "SELECT `id`, `title` FROM MOVIES WHERE status = 2";
    $query2  = "SELECT `id` FROM `halls`";
    $query3  = "SELECT * FROM (
                SELECT screenings.id AS screen_id,
                screenings.movie_id as hiddenmovieid,
                movies.title AS title,
                screenings.hall_id AS hall,
                DATE(datetime_start) AS startingdate,
                TIME(datetime_start) AS startingtime
                FROM `screenings`
                INNER JOIN movies ON
                screenings.movie_id = movies.id
            ) AS screens
            WHERE `screen_id` = ". $whereid;
    $result1 = dbq($query1);
    $result2 = dbq($query2);
    $result3 = dbq($query3);
    while($row = mysqli_fetch_array($result3)) {
        $edit_id      = $row['screen_id'];
        $edit_movieid = $row['hiddenmovieid'];
        $edit_title   = $row['title'];
        $edit_hall    = $row['hall'];
        $date         = $row['startingdate'];
        $time         = $row['startingtime'];
        $date         = date("Y-m-d", strtotime($date));
        $time         = date("H:i:s", strtotime($time));
        $edit_start   = $date.'T'.$time;
    }
}

```

```

echo      '<div id="editpopup" class="box animated zoomIn">
          <form action="'. $_SERVER['PHP_SELF'] .'';
}
echo      '</select><br>';
echo      '<label>Αίθουσα: </label><select name="hall_id"
value="0">';
while($row = mysqli_fetch_array($result2)) {
    echo      '<option value="'. $row['id'] .'';
}
echo      '
          </select><br>
          <label>Ημερομηνία/ώρα: </label><input type="datetime-local"
name="datetime_start" value="'. $edit_start .'

```

```

    }else{
        $id      = $_POST['id'. $id];
        $capacity = $_POST['capacity'. $id];
        $mysqli = dbconnect();
        $query = '
                UPDATE halls SET
                    `capacity` = '.$capacity.'
                WHERE `id`      = '.$id.'
            ';
        if ($mysqli->query($query)) {
            header("Location: edit.php");
        } else {
            echo notify('Σφάλμα!', "Κάποιο σφάλμα προέκυψε, επικοινωνήστε
            με τον διαχειριστή!<br> MySQL error no {$mysqli->errno} : {$mysqli-
            >error}");
            exit();
        }
    }
}
}
$ret .= '</div><br>';
return $ret;
}

```

*//Εμφάνιση Εικονιδίων Διαγραφής και Επεξεργασίας*

```

function showbuttons($table, $id) {
    $fromtable = $table;
    $whereid   = $id;
    $ret = '';
    if ( $_SESSION['admin'] > 95 ) {
        if ( !isset($_POST['submit-delete'.$table.$id]) ) {
            $ret .= '
                <td class="buttons">
                    <form class="buttons" action="'. $_SERVER['PHP_SELF']. "'
                    method="post">
                        <input type="hidden" name="fromtable'. $id.'"
                        value="'. $table.'" />
                        <input type="hidden" name="whereid'. $id.'" value="'.
                        $id.'" />
                        <input type="submit" class="imgdelete" name="submit-
                        delete'. $table.$id.'" value="" />
                    </form>
                </td>
            ';
        }else if ( $fromtable == 'screenings' ){
            $fromtable = $_POST['fromtable'.$id];
            $whereid   = $_POST['whereid'.$id];
            $query     = "UPDATE screenings SET cancelled=1 WHERE
            id={$whereid}";
            if ( dbq($query) ){
                echo notify('Ενημέρωση!', 'Η προβολή που επιλέξατε
                διαγράφηκε!');
            }
        }else {
            $fromtable = $_POST['fromtable'.$id];
            $whereid   = $_POST['whereid'.$id];
            $query     = 'DELETE FROM '.$fromtable.' WHERE id='.$whereid;
            $result = dbq($query);
            if ( $table == 'movies' ) {

```

```

        echo notify('Ενημέρωση!', 'Η ταινία που επιλέξατε
διαγράφηκε!');
    }
}
if ( $fromtable != 'halls' ) {
    if ( !isset($_POST['submit-edit'].$table.$id] ) ) {
        $ret .= '
            <td class="buttons">
                <form class="buttons" action="'. $_SERVER['PHP_SELF']. '"
method="post">
                    <input type="hidden" name="fromtable'. $id.'"
value="'. $table.'" />
                    <input type="hidden" name="whereid'. $id.'" value="'.
$id.'" />
                    <input type="submit" class="imgedit" name="submit-
edit'. $table.$id.'" value="" />
                </form>
            </td>
        ';
    }else {
        $table = $_POST['fromtable'].$id];
        $id = $_POST['whereid'].$id];
        if ( $table == 'movies' ) {
            $ret .= editmovie($id);
        }
        if ( $table == 'screenings' ) {
            $ret .= editscreening($id);
        }
    }
}
}
return $ret;
}

//Εμφάνιση Προβολών στην σελίδα προβολές
function showscreenings($movie_id){ //Προβολή Προβολών
    $ret = '';
    $query = "SELECT `id`, `movie_id`, `hall_id`, DATE(datetime_start) AS
startingdate, TIME(datetime_start) AS startingtime
FROM `screenings` WHERE movie_id='". $movie_id.'" ORDER BY
startingdate ASC, startingtime ASC ";
    $result = dbq($query);
    $current_date = null;
    $ret .= '<label for="'. $movie_id.'" class="labelh3"><h3>Εμφάνιση
Προβολών</h3></label>
        <input type="checkbox" id="'. $movie_id.'" class="toggle"
name="screenstoggle" value="" />';
    if ( $result->num_rows == 0 ) {
        $ret .= '<p>Διαθέσιμες προβολές θα είναι σύντομα διαθέσιμες</p>';
    }else{
        $ret .= '<dl class="togglehide animated bounceInUp">';
        while($row = mysqli_fetch_array($result)) {
            $hall = $row['hall_id'];
            $date = $row['startingdate'];
            $time = $row['startingtime'];
            $date = date("d-m-Y", strtotime($date));
            $time = date("H:i", strtotime($time));
            if ( $current_date != $date ){
                $current_date = $date;
            }
        }
    }
}

```

```

        $ret .= '
        <dt>
            <span>
                στις <span class="yellow">'. $date .'</span> στις
            </span>
        </dt>
        <dd>
            <span>
                στις <span class="yellow">'. $time .'</span> στην
                Αίθουσα <span class="yellow">'. $hall .'</span>
            </span>
        </dd>
        ';
    }
    else{
        $ret .= '
        <dd>
            <span>
                στις <span class="yellow">'. $time .'</span> στην
                Αίθουσα <span class="yellow">'. $hall .'</span>
            </span>
        </dd>
        ';
    }
}
$ret .= '</dl>';
}
return $ret;
}

//Εμφάνιση ταινιών σε προβολή για χρήστες
function globalscreenings(){ //Εμφάνιση box ταινιών σε προβολή
    $ret = '';
    $mysqli = dbconnect();
    $result = dbq("SELECT * from movies WHERE `status` = 2");
    while($row = mysqli_fetch_array($result)) {
        $id      = $row['id'];
        $title   = $row['title'];
        $poster  = $row['poster'];
        $imdbID  = $row['imdbID'];
        $genre   = $row['genre'];
        $runtime = $row['runtime'];
        $plot    = $row['plot'];
        $ret    .= '
            <div class="box">
                <div class="movietitle"><h2>'. $title .'</h2></div>
                <div class="poster">
                    

                </div>
                <div class="data">
                    <h3>Κατηγορία : </h3>
                    <p>'. $genre .'</p>
                    <h3>Διάρκεια : </h3>
                    <p>'. $runtime .'</p>
                    <h3>Υπόθεση : </h3>
                    <p class="plot">'. $plot .'</p>
                    '. showscreenings($id) .'
                </div>

```

```

        <div class="imdblink">
            <a href="http://www.imdb.com/title/'. $imdbID .'
target="_blank" >
                
            </a>
        </div><br>
    </div>
    <br>
    ';
}
return $ret;
}

//Εμφάνιση ταινιών προσεχώς για χρήστες
function globalsoon() { //Εμφάνιση box ταινιών προσεχώς
    $ret = '';
    $mysqli = dbconnect();
    $result = dbq("SELECT * from movies WHERE `status` = 1");
    while($row = mysqli_fetch_array($result)) {
        $id = $row['id'];
        $title = $row['title'];
        $poster = $row['poster'];
        $imdbID = $row['imdbID'];
        $genre = $row['genre'];
        $runtime = $row['runtime'];
        $plot = $row['plot'];
        $ret .= '
            <div class="box">
                <div class="movietitle"><h2>'. $title .'</h2></div>
                <div class="poster">
                    
                </div>
                <div class="data">
                    <h3>Κατηγορία : </h3>
                    <p>'. $genre .'</p>
                    <h3>Διάρκεια : </h3>
                    <p>'. $runtime .'</p>
                    <h3>Υπόθεση : </h3>
                    <p class="plot">'. $plot .'</p>
                </div>
                <div class="imdblink">
                    <a href="http://www.imdb.com/title/'. $imdbID .'
target="_blank" >
                        
                    </a>
                </div><br>
            </div>
            <br>
        ';
    }
    return $ret;
}

//Εμφάνιση πίνακα όλων των ταινιών στον διαχειριστή
function allmoviestable() {
    $ret = '';

```

```

$mysqli = dbconnect();
$result = dbq("SELECT * from movies");
$ret     .= '<table class="admintable animated fadeInLeft">';
$ret     .= '<tr class="adminth">';
$ret     .=
    '<th>Τίτλος</th><th>Χρονολογία</th><th>Διάρκεια</th><th>Είδος</th><th>Κατ
    άσταση</th><th>Έσοδα</th>';
$ret     .= '</tr>';
while($row = mysqli_fetch_array($result)) {
    $id     = $row['id'];
    $table  = 'movies';
    $title  = $row['title'];
    $year   = $row['year'];
    $runtime = $row['runtime'];
    $genre  = $row['genre'];
    $status = $row['status'];
    if ( $status == 0 ) {
        $status = "Αρχείο";
    }else if ( $status == 1 ) {
        $status = "Προσεχώς";
    }
    else{
        $status = "Προβάλλεται";
    }
    $ret .= '<tr class="adminrow">';
    $ret .= '    <td>'. $title .'</td>
        <td>'. $year .'</td>
        <td>'. $runtime .'</td>
        <td>'. $genre .'</td>
        <td>'. $status .'</td>
        <td>'. countmovieincome($id) .'</td>'. showbuttons($table,
        $id) ;
    $ret .= '</tr>';
}
$ret     .= '</table><br>';
return $ret;
}

//Εμφάνιση πίνακα των ταινιών που είναι στο αρχείο στον διαχειριστή
function archivemoviestable() {
    $ret     = '';
    $mysqli = dbconnect();
    $result = dbq("SELECT * from movies WHERE status = 0 ");
    $ret     .= '<table class="admintable animated fadeInLeft">';
    $ret     .= '<tr class="adminth">';
    $ret     .=
        '<th>Τίτλος</th><th>Χρονολογία</th><th>Διάρκεια</th><th>Είδος</th><th>Κατ
        άσταση</th><th>Έσοδα</th>';
    $ret     .= '</tr>';
    while($row = mysqli_fetch_array($result)) {
        $id     = $row['id'];
        $table  = 'movies';
        $title  = $row['title'];
        $year   = $row['year'];
        $runtime = $row['runtime'];
        $genre  = $row['genre'];
        $status = $row['status'];
        if ( $status == 0 ) {
            $status = "Αρχείο";

```



```

    }else if ( $status == 1 ) {
        $status = "Προσεχώς";
    }
    else{
        $status = "Προβάλλεται";
    }
    $ret .= '<tr class="adminrow">';
    $ret .= '    <td>'. $title .'</td>
            <td>'. $year .'</td>
            <td>'. $runtime .'</td>
            <td>'. $genre .'</td>
            <td>'. $status .'</td>
            <td>'. countmovieincome($id) .'</td>'. showbuttons($table,
$id) ;
    $ret .= '</tr>';
}
$ret .= '</table><br>';
return $ret;
}

```

*//Εμφάνιση πίνακα των ταινιών που θα είναι διαθέσιμες προσεχώς στον διαχειριστή*

```

function soonmoviestable() {
    $ret = '';
    $mysqli = dbconnect();
    $result = dbq("SELECT * from movies WHERE status = 1 ");
    $ret .= '<table class="admintable animated fadeInLeft">';
    $ret .= '<tr class="adminth">';
    $ret .=
        '<th>Τίτλος</th><th>Χρονολογία</th><th>Διάρκεια</th><th>Είδος</th><th>Κατ
άσταση</th><th>Έσοδα</th>';
    $ret .= '</tr>';
    while($row = mysqli_fetch_array($result)) {
        $id = $row['id'];
        $table = 'movies';
        $title = $row['title'];
        $year = $row['year'];
        $runtime = $row['runtime'];
        $genre = $row['genre'];
        $status = $row['status'];
        if ( $status == 0 ) {
            $status = "Αρχείο";
        }else if ( $status == 1 ) {
            $status = "Προσεχώς";
        }
        else{
            $status = "Προβάλλεται";
        }
        $ret .= '<tr class="adminrow">';
        $ret .= '    <td>'. $title .'</td>
                <td>'. $year .'</td>
                <td>'. $runtime .'</td>
                <td>'. $genre .'</td>
                <td>'. $status .'</td>
                <td>'. countmovieincome($id) .'</td>'. showbuttons($table,
$id) ;
        $ret .= '</tr>';
    }
    $ret .= '</table><br>';
    return $ret;
}

```

```

}

//Εμφάνιση πίνακα των ταινιών που προβάλλονται τώρα στον διαχειριστή
function screenmoviestable() {
    $ret = '';
    $mysqli = dbconnect();
    $result = dbq("SELECT * from movies WHERE status = 2 ");
    $ret .= '<table class="admintable animated fadeInLeft">';
    $ret .= '<tr class="adminth">';
    $ret .=
        '<th>Τίτλος</th><th>Χρονολογία</th><th>Διάρκεια</th><th>Είδος</th><th>Κατ
        άσταση</th><th>Έσοδα</th>';
    $ret .= '</tr>';
    while($row = mysqli_fetch_array($result)) {
        $id = $row['id'];
        $table = 'movies';
        $title = $row['title'];
        $year = $row['year'];
        $runtime = $row['runtime'];
        $genre = $row['genre'];
        $status = $row['status'];
        if ( $status == 0 ) {
            $status = "Αρχείο";
        }else if ( $status == 1 ) {
            $status = "Προσεχώς";
        }
        else{
            $status = "Προβάλλεται";
        }
        $ret .= '<tr class="adminrow">';
        $ret .= '    <td>'. $title .'</td>
            <td>'. $year .'</td>
            <td>'. $runtime .'</td>
            <td>'. $genre .'</td>
            <td>'. $status .'</td>
            <td>'. countmovieincome($id) .'</td>'. showbuttons($table,
$id) ;
        $ret .= '</tr>';
    }
    $ret .= '</table><br>';
    return $ret;
}

//Συγκεντρωτικός Πίνακας Ταινιών με φίλτρα
function moviestable() {
    $ret = '
        <div class="box">
        <h3>Ταινίες
            '. showhelp('Στον παρακάτω πίνακα εμφανίζονται όλες οι ταινίες που
            έχουν προστεθεί στην βάση.<br>
            Μπορείτε να δείτε ορισμένα στοιχεία καθώς και τα έσοδα από κάθε
            ταινία, να τις επεξεργαστείτε ή να τις διαγράψετε!') .'
        </h3>
        <div class="tabs">
            <div class="tab">
                <input type="radio" id="tab-1" name="tab-group-2" checked>
                <label for="tab-1" class="tablabel">-. countmovies() .'-
            <br>Όλες</label>
            <div class="tabcontent">

```

```

        '. allmoviestable() .'
    </div>
</div>

    <div class="tab">
        <input type="radio" id="tab-2" name="tab-group-2">
        <label for="tab-2" class="tablabel">-. countarchive() .'-
<br>Αρχείο</label>
        <div class="tabcontent">
            '. archivemoviestable() .'
        </div>
    </div>

    <div class="tab">
        <input type="radio" id="tab-3" name="tab-group-2">
        <label for="tab-3" class="tablabel">-. countsoon() .'-
<br>Προσεχώς</label>
        <div class="tabcontent">
            '. soonmoviestable() .'
        </div>
    </div>

    <div class="tab">
        <input type="radio" id="tab-4" name="tab-group-2" >
        <label for="tab-4" class="tablabel">-. countscreens() .'-
<br>Προβάλλονται</label>
        <div class="tabcontent">
            '. screenmoviestable() .'
        </div>
    </div>

</div>
</div>
<br>';
return $ret;
}

//Ημερολόγιο Προβολών Διαχειριστή
function adminscreencalendar(){
    $ret = '';
    $query = "SELECT DATE(datetime_start) as startingdate, cancelled FROM
`screenings` WHERE cancelled=0 GROUP BY startingdate";
    $result = dbq($query);
    $ret .= '<div class="box">
        <h3>Προβολές
            '. showhelp('Στον παρακάτω πίνακα εμφανίζονται όλες οι
ενεργές προβολές και η πληρότητά τους,
                                μπορείτε να τις επεξεργαστείτε ή να τις
διαγράψετε!<br> Πατώντας πάνω στην πληρότητα μπορείτε δείτε αναλυτικά τις
                                κρατήσεις!') .'
        </h3>
        <ul class="accordion"><br>';
    while ( $row = mysqli_fetch_array($result) ) {
        $date = $row['startingdate'];
        $temp = date("Y-m-d", strtotime($date));
        $day = date("D", strtotime($date));
        $day = convertdays($day);
        $date = date("d.m.Y", strtotime($date));
        $fulldate = $date.' - '.$day;
    }
}

```

```

        $ret      .= '<li>
                    <label class="accordtabs" for="'. $date .'>'.
$fulldate .' ('. countdailyscreens($temp) .')</label>
                    <input type="checkbox" id="'. $date .'>
                    <div class="accordcont animated fadeInLeft">
                        '. screeningstable($date) .'
                    </div>
                </li>';
    }
    $ret      .= '</ul></div><br>';
    return $ret;
}

//Πίνακας Προβολών Διαχειριστή
function screeningstable($date) {
    $ret      = '';
    $date     = date("Y-m-d", strtotime($date));
    $query    = "SELECT * FROM
                (
                    SELECT screenings.id as id, screenings.movie_id as
hiddenid, movies.title, movies.runtime, screenings.hall_id,
                        DATE(screenings.datetime_start) AS startingdate,
                    TIME(screenings.datetime_start) AS startingtime, cancelled
                FROM screenings
                INNER JOIN movies ON screenings.movie_id = movies.id
                ) AS screens
                WHERE startingdate = '{$date}' AND cancelled = 0
                ORDER BY hall_id ASC, startingdate ASC, startingtime ASC,
hiddenid ASC";
    $result   = dbq($query);
    $ret      .= '<table class="admintable">';
    $ret      .= '<tr class="adminth">';
    $ret      .=
        '<th>Τίτλος</th><th>Αίθουσα</th><th>Ημερομηνία</th><th>Ωρα</th><th>Πληρότ
ητα</th><th>Διάρκεια</th>';
    $ret      .= '</tr>';
    while($row = mysqli_fetch_array($result)) {
        $id    = $row['id'];
        $table = 'screenings';
        $title = $row['title'];
        $hall  = $row['hall_id'];
        $date  = $row['startingdate'];
        $time  = $row['startingtime'];
        $date  = date("d-m-Y", strtotime($date));
        $time  = date("H:i", strtotime($time));
        $runtime = $row['runtime'];
        $ret   .= '<tr class="adminrow">';
        $ret   .= ' <td>'. $title      . '</td>
                    <td>'. $hall      . '</td>
                    <td>'. $date      . '</td>
                    <td>'. $time      . '</td>
                    <td><a href="availability.php?screen_id='. $id .' "
target="_blank">'. countavailability($id) . '</a></td>
                    <td>'. $runtime  . '</td>
                    '. showbuttons($table, $id) ;
        $ret   .= '</tr>';
    }
    $ret      .= '</table><br>';
    return $ret;
}

```

```

}

//Πίνακας Αιθουσών Διαχειριστή
function hallstable() {
    $ret = '';
    $query = "SELECT * FROM halls";
    $result = dbq($query);
    $ret .= '<div class="box">';
    $ret .= '<h3>Αιθουσες</h3><div class="income">';
    $ret .= '<table class="admintable">';
    $ret .= '<tr class="adminth">';
    $ret .= '<th>Αίθουσα</th><th>Χωρητικότητα</th>';
    $ret .= '</tr>';
    while( $row = mysqli_fetch_array($result) ) {
        $id = $row['id'];
        $table = 'halls';
        $capacity = $row['capacity'];
        $ret .= '<tr class="adminrow">';
        $ret .= '<td>'. $id . '</td><td>'. $capacity . '</td>'.
        showbuttons($table, $id) ;
        $ret .= '</tr>';
    }
    $ret .= '</table></div></div><br>';
    return $ret;
}

//Συνάρτηση Παραγωγής Λίστας Μηνών για Πιστωτική Κάρτα
function monthsdropdown() {
    $ret = '<select name="ccmonth">';
    $ret .= '<option value="1" selected>1</option>';
    for ( $month = 2 ; $month <= 12 ; $month++ ) {
        $ret .= '<option value="'. $month .' ">'. $month . '</option>';
    }
    $ret .= '</select>';
    return $ret;
}

//Συνάρτηση Παραγωγής Λίστας Χρόνων για Πιστωτική Κάρτα
function yearsdropdown() {
    $ret = '';
    $ret = '<select name="ccyear">';
    $temp1 = getdate();
    $year = intval($temp1['year']);
    $temp2 = $year + 1;
    $temp3 = $year + 20;
    $ret .= '<option value="'. $year ." " selected>'. $year . '</option>';
    for ( $year = $temp2 ; $year <= $temp3 ; $year++ ) {
        $ret .= '<option value="'. $year .' ">'. $year . '</option>';
    }
    $ret .= '</select>';
    return $ret;
}

//Συνάρτηση Εύρεσης id χρήστη μέσω username
function getuserid($username) {
    $query = "SELECT * from users WHERE username LIKE '{$username}' LIMIT 1";
    $result = dbq($query);
    $row = mysqli_fetch_array($result);
    $id = $row['id'];
}

```

```

return $id;
}

//Ημερομηνίες που έχει κάνει κράτηση ένας χρήστης
function userresdates($user_id) {
    $query = " SELECT startingdate FROM
        (
            SELECT
                DATE(screenings.datetime_start) AS startingdate,
                TIME(screenings.datetime_start) AS startingtime
            FROM reservations
            INNER JOIN screenings
                ON reservations.screening_id = screenings.id
            INNER JOIN movies
                ON screenings.movie_id = movies.id
            WHERE user_id = {$user_id}
        ) as res
        GROUP BY startingdate
        ORDER BY startingdate DESC, startingtime DESC";
    $result = dbq($query);
    return $result;
}

//Ημερομηνίες που έχει κάνει κράτηση ένας χρήστης ανά ημερομηνία
function getuserres($user_id, $date) {
    $date = date("Y-m-d", strtotime($date));
    $query = " SELECT
        reservations.id AS res_id,
        reservations.code AS res_code,
        reservations.screening_id AS screen_id,
        reservations.user_id AS user_id,
        reservations.seat AS res_seat,
        reservations.confirmed AS res_conf,
        reservations.paid AS res_paid,
        screenings.id AS SCREEN_ID,
        screenings.movie_id AS movie_id,
        DATE(screenings.datetime_start) AS startingdate,
        TIME(screenings.datetime_start) AS startingtime,
        screenings.hall_id AS hall_id,
        movies.title AS movie_title,
        movies.id AS MOVIE_ID,
        movies.poster AS poster,
        movies.runtime AS runtime
    FROM reservations
    INNER JOIN screenings
        ON reservations.screening_id = screenings.id
    INNER JOIN movies
        ON screenings.movie_id = movies.id
    WHERE reservations.user_id = {$user_id} AND
        reservations.confirmed = 1 AND DATE(screenings.datetime_start) =
        '{$date}'
        ORDER BY startingdate DESC, startingtime DESC
    ";
    $result = dbq($query);
    return $result;
}

//Εμφάνιση κρατήσεων χρήστη
function showuserres($user_id, $date) {

```

```

$ret = '';
$result = getuserres($user_id, $date);
while ( $row = mysqli_fetch_array($result) ){
    $res_id      = $row['res_id'];
    $screen_id   = $row['screen_id'];
    $poster      = $row['poster'];
    $title       = $row['movie_title'];
    $time        = $row['startingtime'];
    $time        = date("H:i", strtotime($time));
    $seat        = $row['res_seat'];
    $hall        = $row['hall_id'];
    $runtime     = $row['runtime'];
    $res_paid    = $row['res_paid'];
    if ( $res_paid == 1 ) {
        $ispaid   = 'Εξοφλήθη';
        $paidimage = 'images/paid.png';
    }else {
        $ispaid   = 'Αναμένεται Πληρωμή';
        $paidimage = 'images/notpaid.png';
    }
    $ret        .= '<div class="minibox animated fadeInLeft">';
    $ret        .= '<div class="miniposter"></div>';
    $ret        .= '<div class="minidata">
        <div class="title">'. $title .'</div>
        <div class="time">'. $time .'</div>
        <div class="seat"> Θέση: '. $seat .'</div>
        <div class="hall"> Αίθουσα: '. $hall .'</div>
        <div class="runtime"> Διάρκεια: '. $runtime
    . '</div>';
    $ret        .= '</div>
        <div class="paidcheck"></div></div>';
}
return $ret;
}

//Ημερολόγιο Κρατήσεων Χρήστη
function userrescal($user_id) {
    $ret = '';
    $ret .= '<ul class="accordion animated fadeInLeft"><br>';
    $dates = userresdates($user_id);
    while ( $row = mysqli_fetch_array($dates) ) {
        $date      = $row['startingdate'];
        $day       = date("D", strtotime($date));
        $day       = convertdays($day);
        $date      = date("d.m.Y", strtotime($date));
        $fulldate  = $date.' - '.$day;
        $ret       .= '<li><label class="accordtabs" for="'. $date .' ">'.
    $fulldate .'</label>';
        $ret       .= '<input type="checkbox" name="a" id="'. $date .' "><div
class="accordcont animated fadeInLeft">';
        $ret       .= showuserres($user_id, $date);
        $ret       .= '</div></li>';
    }
    $ret .= '</ul>';
    return $ret;
}
}

```

```

//Συνάρτηση Ανάκτησης Συναλλαγών Χρήστη
function gettransactions($user_id) {
    $query = " SELECT user_id, amount, DATE(datetime) AS date, TIME(datetime)
        AS time FROM transactions
            WHERE user_id = {$user_id}
            ORDER BY date DESC, time DESC";
    $result = dbq($query);
    return $result;
}

//Εμφάνιση Συναλλαγών Χρήστη
function showtransactions($user_id) {
    $ret = '';
    $ret .= '<table class="admintable animated fadeInLeft"><tr
        class="adminth"><th>Ποσό</th><th>Ημερομηνία</th><th>Ωρα</th></tr>';
    $result = gettransactions($user_id);
    while ( $row = mysqli_fetch_array($result) ) {
        $amount = $row['amount'];
        $amount .= ' €';
        $date = $row['date'];
        $date = date("d/m/Y", strtotime($date));
        $time = $row['time'];
        $time = date("H:m:s", strtotime($time));
        $ret .= '<tr class="adminrow">';
        $ret .= '<td>'. $amount .'</td><td>'. $date .'</td><td>'. $time
            .'</td>';
        $ret .= '</tr>';
    }
    $ret .= '</table>';
    return $ret;
}

//Φόρμα πληρωμής κράτησης
function showrespayform($user_id){
    $ret = '';
    $username = getusername($user_id);
    $i = 0;
    $query = " SELECT
        reservations.id AS res_id,
        reservations.code AS res_code,
        reservations.screening_id AS screen_id,
        reservations.user_id AS user_id,
        reservations.seat AS res_seat,
        reservations.confirmed AS res_conf,
        reservations.paid AS res_paid,
        screenings.id AS SCREEN_ID,
        screenings.movie_id AS movie_id,
        DATE(screenings.datetime_start) AS startingdate,
        TIME(screenings.datetime_start) AS startingtime,
        screenings.hall_id AS hall_id,
        movies.title AS movie_title,
        movies.id AS MOVIE_ID,
        movies.runtime AS runtime
    FROM reservations
    INNER JOIN screenings
        ON reservations.screening_id = screenings.id
    INNER JOIN movies
        ON screenings.movie_id = movies.id

```



```

        WHERE reservations.user_id = {$user_id} AND
        reservations.confirmed = 1 AND reservations.paid = 0
        ORDER BY startingdate DESC, startingtime DESC";
    $result = dbq($query);
    $ret    .= '<div class="animated fadeInLeft"><form action="'.
$_SERVER['PHP_SELF']. '" method="post">
        <input type="hidden" name="user_id" value="'. $user_id .'">
        <input type="hidden" name="username" value="'. $username .'">
        <input type="submit" name="submit-cancelres" value="Ακύρωση
Κρατήσεων">
        <input type="submit" name="submit-payres" value="Πληρωμή
Κρατήσεων">
        <ul class="buttonlist">;
while ( $row = mysqli_fetch_array($result) ){
    $res_id    = $row['res_id'];
    $title     = $row['movie_title'];
    $date      = $row['startingdate'];
    $date      = date("d/m/Y", strtotime($date));
    $time      = $row['startingtime'];
    $time      = date("H:i", strtotime($time));
    $seat      = 'Θέση: ';
    $seat      .= $row['res_seat'];
    $hall      = 'Αίθουσα: ';
    $hall      .= $row['hall_id'];
    $showdata  = $title.'<br>'.$date.' '.$time.'<br>'.$hall.'<br>'.$seat;
    $ret       .= '<li class="button">
                <div class="buttonlist-cont">
                    <input type="checkbox" id="button-'. $i .' "
name="tickets[]" value="'. $res_id .' "
                    <label for="button-'. $i .' ">'. $showdata
.'</label>
                    <div class="ischecked"></div>
                </div>
            </li>';

    $i++;
}
$ret    .= '</ul></div></form>';
return $ret;
}

//Εμφάνιση Προφίλ Χρήστη
function showprofile($username) {
    $user_id    = getuserid($username);
    $ret        = '';
    $ret        .= '<div class="box">';
    $ret        .= '<h3>Προφίλ: '. $username . ' - '. showcredits($username)
.'</h3>';
    $ret        .= '<div class="tabs">
                <div class="tab">
                    <input type="radio" id="tab-1" name="tab-group-1"
checked>
                    <label for="tab-1"
class="tablabel">Ιστορικό<br>Κρατήσεων</label>
                    <div class="tabcontent">
                        '. userrescal($user_id) .'
                    </div>
                </div>
                <div class="tab">
                    <input type="radio" id="tab-2" name="tab-group-1">

```

```

        <label for="tab-2"
class="tablabel">Ιστορικό<br>Συναλλαγών</label>
        <div class="tabcontent">
            '. showtransactions($user_id) .'
        </div>
    </div>
    <div class="tab">
        <input type="radio" id="tab-3" name="tab-group-1">
        <label for="tab-3"
class="tablabel">Εξόφληση<br>Ακύρωση</label>
        <div class="tabcontent">
            '. showrespayform($user_id) .'
        </div>
    </div>';
if ( $_SESSION['user'] == $username ) {
    $ret .= ' <div class="tab">
        <input type="radio" id="tab-4" name="tab-group-1">
        <label for="tab-4"
class="tablabel">Αλλαγή<br>Κωδικού</label>
        <div class="tabcontent">
            '. changepassword($user_id) .'
        </div>
    </div> ';
}else if ( $_SESSION['admin'] > 95 ){
    $ret .= ' <div class="tab">
        <input type="radio" id="tab-5" name="tab-group-1">
        <label for="tab-5"
class="tablabel">Αλλαγή<br>Δικαιωμάτων</label>
        <div class="tabcontent">
            '. changepermissions($user_id) .'
        </div>
    </div> ';
    $ret .= ' <div class="tab">
        <input type="radio" id="tab-6" name="tab-group-1">
        <label for="tab-6" class="tablabel">Έσοδα<br>από ' .
$username .'</label>
        <div class="tabcontent">
            '. countuserincome($user_id)[0] .'
        </div>
    </div> ';
}
if ( $_SESSION['admin'] > 90 ) {
    $ret .= ' <div class="tab">
        <input type="radio" id="tab-7" name="tab-group-1">
        <label for="tab-7" class="tablabel">Στοιχεία<br>του
'. $username .'</label>
        <div class="tabcontent">
            '. getuserinfo($user_id) .'
        </div>
    </div> ';
}
$ret .= '</div>
</div>
<br>';
return $ret;
}

//Εμφάνιση Προσωπικών Στοιχείων Χρήστη
function getuserinfo($user_id) {

```

```

$ret      = '';
$query    = "SELECT * from users WHERE id={$user_id} LIMIT 1";
$result   = dbq($query);
$row      = mysqli_fetch_array($result);
$code     = $row['code'];
$firstname = $row['fname'];
$lastname = $row['lname'];
$birthdate = $row['birthdate'];
$birthdate = date("d/m/Y", strtotime($birthdate));
$clubcard = $row['clubcard'];
$ret      .= '<div class="income animated fadeInLeft">
              <form action="printclubcard.php" method="post">
              <input type="hidden" name="code" value="'. $code .' " />
              <input type="hidden" name="firstname" value="'. $firstname
              .'" />
              <input type="hidden" name="lastname" value="'. $lastname
              .'" />
              <input type="submit" name="submit-printclubcard"
value="Έκδοση Κάρτας Club" /></form><br><br>
              <table class="admintable">
              <tr class="adminrow">
              <td><label>Όνομα:</label></td>
              <td><span class="align-left">'. $firstname
              .'\</span></td>
              </tr>
              <tr class="adminrow">
              <td><label>Επίθετο:</label></td>
              <td><span class="align-left">'. $lastname
              .'\</span></td>
              </tr>
              <tr class="adminrow">
              <td><label>Ημερομηνία Γέννησης:</label></td>
              <td><span class="align-left">'. $birthdate
              .'\</span></td>
              </tr>
              <tr class="adminrow">
              <td><label>Εκδόσεις Κάρτας:</label></td>
              <td><span class="align-left">'. $clubcard
              .'\</span></td>
              </tr>
              </table>
              </div>';
return $ret;
}

//Εύρεση ονόματος χρήστη μέσω id
function getusername($user_id) {
    $query    = "SELECT * from users WHERE id={$user_id} LIMIT 1";
    $result   = dbq($query);
    $row      = mysqli_fetch_array($result);
    $username = $row['username'];
    return $username;
}

//Φόρμα Αλλαγής κωδικού
function changepassword($user_id){
    $ret      = '';
    $ret      .= '<form action="'. $_SERVER['PHP_SELF'] .' " method="post"
class="animated fadeInLeft">

```

```

        <input type="hidden" name="user_id" value="'. $user_id .'

```

```

        <label>Κωδικός Εισιτηρίου: </label><input type="text"
name="code" /><br>
        <input type="submit" name="submit-searchticket"
value="Αναζήτηση" /><br>
        </form></div><br>';
return $ret;
}

//Εμφάνιση Αποτελεσμάτων αναζήτησης κράτησης
function showsearchres($search) {
    $ret    = '';
    $i      = 0;
    $query  = " SELECT
                reservations.id AS res_id,
                reservations.code AS res_code,
                reservations.screening_id AS screen_id,
                reservations.user_id AS user_id,
                reservations.seat AS res_seat,
                reservations.confirmed AS res_conf,
                reservations.paid AS res_paid,
                reservations.notes AS notes,
                screenings.id AS SCREEN_ID,
                screenings.movie_id AS movie_id,
                DATE(screenings.datetime_start) AS startingdate,
                TIME(screenings.datetime_start) AS startingtime,
                screenings.hall_id AS hall_id,
                movies.title AS movie_title,
                movies.id AS MOVIE_ID,
                movies.runtime AS runtime
            FROM reservations
            INNER JOIN screenings
                ON reservations.screening_id = screenings.id
            INNER JOIN movies
                ON screenings.movie_id = movies.id
            WHERE reservations.notes LIKE '%{$search}%' AND
            reservations.confirmed = 1 AND reservations.paid = 0
            ORDER BY startingdate DESC, startingtime DESC";
    $result = dbq($query);
    if ( $result->num_rows == 0 ) {
        echo notify('Προειδοποίηση!', 'Δεν βρέθηκαν κρατήσεις!');
    }else {
        $user_id = getuserid($_SESSION['user']);
        $ret     .= '<div class="box"><h3>Αποτελέσματα Αναζήτησης "'. $search
        .'"</h3>
                <form action="'. $_SERVER['PHP_SELF']. '" method="post">
                <input type="hidden" name="user_id" value="'. $user_id .' ">
                <input type="submit" name="submit-cancelres" value="Ακύρωση
Κρατήσεων">
                <input type="submit" name="submit-payres" value="Πληρωμή
Κρατήσεων">
                <ul class="buttonlist">';
        while ( $row = mysqli_fetch_array($result) ){
            $res_id      = $row['res_id'];
            $title       = $row['movie_title'];
            $date        = $row['startingdate'];
            $date        = date("d/m/Y", strtotime($date));
            $time        = $row['startingtime'];
            $time        = date("H:i", strtotime($time));
            $seat        = 'Θέση: ';

```

```

        $seat      .= $row['res_seat'];
        $hall      = 'Αίθουσα: ';
        $hall      .= $row['hall_id'];
        $notes     = $row['notes'];
        $showdata  = $title.'  
'.$date.'
        '.$time.'  
'.$hall.'  
'.$seat.'  
'.$notes;
        $ret       .= ' <li class="button">
                        <div class="buttonlist-cont">
                            <input type="checkbox" id="button2-'. $i
        .'" name="tickets[]" value="'. $res_id .'">
                            <label for="button2-'. $i .' ">'. $showdata
        .'/></label>
                        <div class="ischecked"></div>
                        </div>
                    </li>';

        $i++;
    }
    $ret  .= '</ul></form></div><br>';
}
return $ret;
}

//Εμφάνιση Αποτελεσμάτων Ελέγχου εγκυρότητας εισιτηρίου
function showsearchticket($code){
    $ret = '';
    $query = "SELECT * FROM reservations WHERE code='{$code}' AND
    confirmed=1";
    $result = dbq($query);
    if ( $result->num_rows == 0 ) {
        $ret .= '<p>Άκυρο Εισιτήριο!!!</p>';
    }else {
        $row = mysqli_fetch_array($result);
        $paid = $row['paid'];
        if ( $paid == 1 ) {
            $ret .= '<p>Εγκυρο Εισιτήριο!</p>';
        }else {
            $ret .= '<p>Εγκυρο Εισιτήριο αλλά εκκρεμεί πληρωμή</p>';
        }
    }
    return $ret;
}

//Αναζήτηση χρήστη μέσω μοναδικού αναγνωριστικού
function searchuserbycode($code) {
    $ret = '';
    $mysqli = dbconnect();
    $query = "SELECT * FROM users WHERE code='{$code}'";
    $result = dbq($query);
    return $result;
}

//Φόρμα Αναζήτησης χρήστη μέσω μοναδικού αναγνωριστικού
function showsearchuserbycode() {
    $ret = '';
    $ret .= '<div class="box">
            <h3>
                Εύρεση βάση κωδικού
                '. showhelp('Αναζήτηση χρήστη μέσω του μοναδικού
                αναγνωριστικού!') .'

```

```

        </h3>
        <form action="'. $_SERVER['PHP_SELF']. '" method="post">
        <label>Αναγνωριστικό Χρήστη</label><input type="text"
name="code" /><br>
        <input type="submit" name="submit-searchbycode"
value="Αναζήτηση" />
        </form>
    </div><br>';
return $ret;
}

//Αναζήτηση Χρήστη μέσω ονόματος χρήστη
function searchuserbyusername($username){
    $ret = '';
    $ret = '';
    $mysqli = dbconnect();
    $query = "SELECT * FROM users WHERE username LIKE '%{$username}%'";
    $result = dbq($query);
    return $result;
}

//Εμφάνιση Φόρμας Αναζήτησης χρήστη μέσω ονόματος χρήστη
function showsearchuserbyusername() {
    $ret = '';
    $ret .= '<div class="box">
        <h3>
            Εύρεση βάση ονόματος χρήστη
            '. showhelp('Αναζήτηση χρήστη μέσω του ονόματος χρήστη που
χρησιμοποίησε κατά την εγγραφή του!') .'
        </h3>
        <form action="'. $_SERVER['PHP_SELF']. '" method="post">
        <label>Όνομα Χρήστη</label><input type="text" name="username"
/><br>
        <input type="submit" name="submit-searchbyusername"
value="Αναζήτηση" />
        </form>
    </div><br>';
return $ret;
}

//Εμφάνιση Ειδοποίησης από την εφαρμογή
function notify($title, $msg){
    $ret = '';

    if ( $title == 'Σφάλμα!' ) { $class = 'error'; }
    else if ( $title == 'Ενημέρωση!' ) { $class = 'report'; }
    else { $class = 'warning'; }

    $ret .= '
        <div class="notification" class="'. $class .'>
        <div id="notifttitle" class="'. $class .'>
            <h4>'. $title .'</h4>
        </div>
        <div id="notifmsg" class="'. $class .'>
            <p>'. $msg .'</p>
        </div>
    </div>';
return $ret;
}

```

```

//Εμφάνιση βοήθειας
function showhelp($msg) {
    $ret = '';
    $ret .= '<div class="help">
        <div class="icon"><span?</span></div>
        <div class="bubble animated fadeInUp"><span class="helpmsg">'.
        $msg .'</span></div>
        </div>';
    return $ret;
}

//Καταμέτρηση Ταινιών
function countmovies(){
    $query = "SELECT COUNT(*) AS total FROM movies";
    $result = dbq($query);
    $row = mysqli_fetch_array($result);
    $total = $row['total'];
    return $total;
}

//Καταμέτρηση Ταινιών στο αρχείο
function countarchive(){
    $query = "SELECT COUNT(*) AS total FROM movies WHERE status=0";
    $result = dbq($query);
    $row = mysqli_fetch_array($result);
    $total = $row['total'];
    return $total;
}

//Καταμέτρηση Ταινιών προσεχώς
function countsoon(){
    $query = "SELECT COUNT(*) AS total FROM movies WHERE status=1";
    $result = dbq($query);
    $row = mysqli_fetch_array($result);
    $total = $row['total'];
    return $total;
}

//Καταμέτρηση Ταινιών σε προβολή
function countscreens(){
    $query = "SELECT COUNT(*) AS total FROM movies WHERE status=2";
    $result = dbq($query);
    $row = mysqli_fetch_array($result);
    $total = intval($row['total']);
    return $total;
}

//Καταμέτρηση Εσόδων Ανά Ταινία
function countmovieincome($movie_id){
    $query = "SELECT COUNT(res_code) AS total FROM (
        SELECT
            reservations.code AS res_code,
            reservations.confirmed AS res_conf,
            reservations.paid AS res_paid,
            screenings.movie_id AS movie_id
        FROM reservations
        INNER JOIN screenings
            ON reservations.screening_id = screenings.id
    )";
}

```



```

        INNER JOIN movies
            ON screenings.movie_id = movies.id
        WHERE reservations.confirmed = 1 AND reservations.paid = 1
        AND movie_id={ $movie_id } as tickets";
$result = dbq($query);
$row    = mysqli_fetch_array($result);
$total  = $row['total'];
$income = $total*5;
$income .= ' €';
return $income;
}

//Καταμέτρηση Εσόδων από χρήστη
function countuserincome($user_id) {
    $ret      = '';
    $sumtotal = 0;
    $query1   = " SELECT * FROM
        (SELECT
            user_id, amount, DATE(datetime) AS date
        FROM transactions
        WHERE user_id={ $user_id }
        GROUP BY date) AS tr
        ORDER BY date DESC";
    $result1  = dbq($query1);
    if ( $result1->num_rows == 0 ) {
        $ret  .= '<p>Δεν βρέθηκαν έσοδα από τον χρήστη '.
            getusername($user_id).'!</p>';
    }else {
        $i = 0;
        while ( $row = mysqli_fetch_array($result1) ){
            $dates[$i] = $row['date'];
            $i++;
        }
        $ret  .= '<div class="income animated fadeInLeft"><table
class="admintable">';
        $sumtotal = 0;
        $trs      = '';
        foreach ( $dates as $date ){
            $query2   = " SELECT
                user_id, amount, DATE(datetime) AS date
            FROM transactions
            WHERE user_id={ $user_id } AND
DATE(datetime)='{ $date }'";
            $result2  = dbq($query2);
            $dailysum = 0;
            while ( $row2 = mysqli_fetch_array($result2) ) {
                $amount = $row2['amount'];
                $dailysum += $amount;
            }
            $showdate  = date("d.m.Y", strtotime($date));
            $trs      .= ' <tr class="adminrow">
                <td>'. $showdate .'</td><td>'. $dailysum
            .'€</td>
                </tr>';
            $sumtotal += $dailysum;
        }
        $ret  .= '<tr class="adminrow">
            <th class="adminth">ΣΥΝΟΛΟ</th><th class="adminth">'.
            $sumtotal .'€</th>

```

```

        </tr>';
    $ret    .= $trs;
    $ret    .= '</table></div>';
}
return array($ret, $sumtotal);
}

//Καταμέτρηση Κρατήσεων ανά ημερομηνία
function countdailyscreens($date){
    $query = "SELECT COUNT(id) AS total FROM screenings WHERE
        DATE(datetime_start)='{ $date}' AND cancelled=0";
    $result = dbq($query);
    $row    = mysqli_fetch_array($result);
    $total  = $row['total'];
    return $total;
}

//Εμφάνιση Πληρότητας Αίθουσας
function countavailability($screen_id){
    $query1 = "SELECT hall_id FROM screenings WHERE id={ $screen_id}";
    $result1 = dbq($query1);
    $row1    = mysqli_fetch_array($result1);
    $hall_id = $row1['hall_id'];

    $query2 = "SELECT capacity FROM halls WHERE id={ $hall_id}";
    $result2 = dbq($query2);
    $row2    = mysqli_fetch_array($result2);
    $total   = $row2['capacity'];

    $query3 = "SELECT COUNT(id) AS total FROM reservations WHERE
        screening_id='{ $screen_id}'";
    $result3 = dbq($query3);
    $row3    = mysqli_fetch_array($result3);
    $taken   = $row3['total'];

    $show    = $taken.'/'. $total;
    return $show;
}

//Εύρεση Τρεχόντων Δικαιωμάτων Χρήστη
function getuserpermissions($user_id) {
    $query = "SELECT * from users WHERE id={ $user_id}";
    $result = dbq($query);
    $row    = mysqli_fetch_array($result);
    $isAdmin = $row['isAdmin'];
    return $isAdmin;
}

//Εμφάνιση Στατιστικών Με φίλτρα
function clubstats() {
    $ret = '';
    $ret .= '<div class="box">
        <h3>Εσοδα</h3>
        <div class="tabs">

            <div class="tab">
                <input type="radio" id="tab-1" name="tab-group-1"
checked>

```

```

        <label for="tab-1"
class="tablabel">Ανά<br>Ημερομηνία</label>
        <div class="tabcontent">
            '. showformincomedates() .'
        </div>
    </div>

    <div class="tab">
        <input type="radio" id="tab-2" name="tab-group-1" >
        <label for="tab-2"
class="tablabel">Ανά<br>Ηλικία</label>
        <div class="tabcontent">
            '. userincomeages() .'
        </div>
    </div>

    <div class="tab">

        <input type="radio" id="tab-3" name="tab-group-1" >
        <label for="tab-3"
class="tablabel">Ανά<br>Ημέρα</label>
        <div class="tabcontent">
            '. incomebyday() . '
        </div>
    </div>

    <div class="tab">

        <input type="radio" id="tab-4" name="tab-group-1" >
        <label for="tab-4"
class="tablabel">Ανά<br>Ωρα</label>
        <div class="tabcontent">
            '. incomebyhours() . '
        </div>
    </div>
</div>';
return $ret;
}

//Φόρμα Αναζήτησης Εσόδων σε διάστημα ημερομηνιών
function showformincomedates(){
    $ret = '';
    $ret .= '<form class="animated fadeInLeft" action="'. $_SERVER['PHP_SELF'].
        '" method="post">
        <label>Από</label><input type="date" name="date1"><br>
        <label>Έως</label><input type="date" name="date2"><br>
        <input type="submit" name="submit-searchincome"
        value="Αναζήτηση" />
        </form>';
    return $ret;
}

//Υπολογισμός και εμφάνιση στατιστικών ανά ηλικία
function userincomeages() {
    $ret = '';
    $query1 = "SELECT id, birthdate FROM users";
    $result1 = dbq($query1);
    $c18_25 = 0;

```

```

$c26_33 = 0;
$c34_40 = 0;
$c41_60 = 0;
$c61 = 0;
$a18_25 = 0;
$a26_33 = 0;
$a34_40 = 0;
$a41_60 = 0;
$a61 = 0;
$users = $result1->num_rows;
while ( $row = mysqli_fetch_array($result1) ){
    $birthdate = $row['birthdate'];
    $user_id = $row['id'];
    $birthdate = new DateTime($birthdate);
    $age = $birthdate->diff(new DateTime);
    $age = intval($age->y);
    switch($age){
        case $age>=18 && $age<=25 : $c18_25++;
    $a18_25+=countuserincome($user_id)[1]; break;
        case $age>=26 && $age<=33 : $c26_33++;
    $a26_33+=countuserincome($user_id)[1]; break;
        case $age>=34 && $age<=40 : $c34_40++;
    $a34_40+=countuserincome($user_id)[1]; break;
        case $age>=41 && $age<=60 : $c41_60++;
    $a41_60+=countuserincome($user_id)[1]; break;
        case $age>=61 : $c61++; $a61
    +=countuserincome($user_id)[1]; break;
    }
}
$rate18_25 = round($c18_25 / $users * 100, 1);
$rate26_33 = round($c26_33 / $users * 100, 1);
$rate34_40 = round($c34_40 / $users * 100, 1);
$rate41_60 = round($c41_60 / $users * 100, 1);
$rate61 = round($c61 / $users * 100, 1);
$query2 = "SELECT ";
$totalam = $a18_25 + $a26_33 + $a34_40 + $a41_60 + $a61;
$ret .= ' <table class="admintable animated fadeInLeft">
    <tr class="adminth">
        <th>18-25</th>
        <th>26-33</th>
        <th>34-40</th>
        <th>41-60</th>
        <th>61+</th>
        <th>Σύνολο</th>
    </tr>
    <tr class="adminrow">
        <td>'. $c18_25 .'

```

```

        <td> -                </td>
    </tr>
    <tr class="adminrow">
        <td>'. $a18_25      .'€</td>
        <td>'. $a26_33      .'€</td>
        <td>'. $a34_40      .'€</td>
        <td>'. $a41_60      .'€</td>
        <td>'. $a61         .'€</td>
        <td>'. $totalam     .'€</td>
    </tr>
</table>';
return $ret;
}

//Υπολογισμός και εμφάνιση στατιστικών ανά ώρα
function incomebyhours(){
    $ret = '';
    $query = " SELECT
                reservations.screening_id AS screen_id,
                reservations.confirmed AS res_conf,
                reservations.paid AS res_paid,
                TIME(screenings.datetime_start) AS startingtime
            FROM reservations
            INNER JOIN screenings
            ON reservations.screening_id = screenings.id
            WHERE reservations.confirmed = 1 AND reservations.paid = 1";

    $count0 = 0;
    $count1 = 0;
    $count2 = 0;
    $count3 = 0;
    $count4 = 0;
    $count5 = 0;
    $count6 = 0;
    $count7 = 0;
    $result = dbq($query);
    while ( $row = mysqli_fetch_array($result) ) {
        $time = $row['startingtime'];
        $time = strtotime($time);
        switch ($time) {
            case $time >= strtotime('13:00:00') && $time <=
                strtotime('16:30:00') : $count0++; break;
            case $time >= strtotime('16:31:00') && $time <=
                strtotime('18:30:00') : $count1++; break;
            case $time >= strtotime('18:31:00') && $time <=
                strtotime('19:30:00') : $count2++; break;
            case $time >= strtotime('19:31:00') && $time <=
                strtotime('20:30:00') : $count3++; break;
            case $time >= strtotime('20:31:00') && $time <=
                strtotime('21:30:00') : $count4++; break;
            case $time >= strtotime('21:31:00') && $time <=
                strtotime('22:30:00') : $count5++; break;
            case $time >= strtotime('22:31:00') || $time <=
                strtotime('04:00:00') : $count6++; break;
        }
    }
    $total = $count0 + $count1 + $count2 + $count3 + $count4 + $count5 +
    $count6;
    $rate0 = round( $count0 / $total * 100, 1);
    $rate1 = round( $count1 / $total * 100, 1);

```

```

$rate2 = round( $count2 / $total * 100, 1);
$rate3 = round( $count3 / $total * 100, 1);
$rate4 = round( $count4 / $total * 100, 1);
$rate5 = round( $count5 / $total * 100, 1);
$rate6 = round( $count6 / $total * 100, 1);
$ret    .= ' <table class="admintable animated fadeInLeft">
        <tr class="adminth" id="hours">
            <th>13:00-16:30 </th>
            <th>16:31-18:30 </th>
            <th>18:31-19:30 </th>
            <th>19:31-20:30 </th>
            <th>20:31-21:30 </th>
            <th>21:31-22:30 </th>
            <th>22:31+      </th>
            <th>Σύνολο      </th>
        </tr>
        <tr class="adminrow">
            <td>'. $count0 .'</td>
            <td>'. $count1 .'</td>
            <td>'. $count2 .'</td>
            <td>'. $count3 .'</td>
            <td>'. $count4 .'</td>
            <td>'. $count5 .'</td>
            <td>'. $count6 .'</td>
            <td>'. $total  .'</td>
        </tr>
        <tr class="adminrow">
            <td>'. $rate0 .'%</td>
            <td>'. $rate1 .'%</td>
            <td>'. $rate2 .'%</td>
            <td>'. $rate3 .'%</td>
            <td>'. $rate4 .'%</td>
            <td>'. $rate5 .'%</td>
            <td>'. $rate6 .'%</td>
            <td>    -      </td>
        </tr>
        <tr class="adminrow">
            <td>'. $count0 * 5 .'€</td>
            <td>'. $count1 * 5 .'€</td>
            <td>'. $count2 * 5 .'€</td>
            <td>'. $count3 * 5 .'€</td>
            <td>'. $count4 * 5 .'€</td>
            <td>'. $count5 * 5 .'€</td>
            <td>'. $count6 * 5 .'€</td>
            <td>'. $total  * 5 .'€</td>
        </tr>
    </table>';
return $ret;
}

//Υπολογισμός και εμφάνιση στατιστικών ανά ημέρα της εβδομάδας
function incomebyday(){
    $ret    = '';
    $query  = " SELECT
                reservations.screening_id AS screen_id,
                reservations.confirmed AS res_conf,
                reservations.paid AS res_paid,
                DATE(screenings.datetime_start) AS startingdate
            FROM reservations

```

```

INNER JOIN screenings
  ON reservations.screening_id = screenings.id
WHERE reservations.confirmed = 1 AND reservations.paid = 1";
$count0 = 0;
$count1 = 0;
$count2 = 0;
$count3 = 0;
$count4 = 0;
$count5 = 0;
$count6 = 0;
$count7 = 0;
$result = dbq($query);
while ( $row = mysqli_fetch_array($result) ) {
  $date = $row['startingdate'];
  $day = date("D", strtotime($date));
  switch($day){
    case "Mon": $count0++; break;
    case "Tue": $count1++; break;
    case "Wed": $count2++; break;
    case "Thu": $count3++; break;
    case "Fri": $count4++; break;
    case "Sat": $count5++; break;
    case "Sun": $count6++; break;
  }
}
$total = $count0 + $count1 + $count2 + $count3 + $count4 + $count5 +
  $count6;
$rate0 = round( $count0 / $total * 100, 1);
$rate1 = round( $count1 / $total * 100, 1);
$rate2 = round( $count2 / $total * 100, 1);
$rate3 = round( $count3 / $total * 100, 1);
$rate4 = round( $count4 / $total * 100, 1);
$rate5 = round( $count5 / $total * 100, 1);
$rate6 = round( $count6 / $total * 100, 1);
$ret .= ' <table class="admintable animated fadeInLeft">
  <tr class="adminth">
    <th>Δευτέρα </th>
    <th>Τρίτη </th>
    <th>Τετάρτη </th>
    <th>Πέμπτη </th>
    <th>Παρασκευή </th>
    <th>Σάββατο </th>
    <th>Κυριακή </th>
    <th>Σύνολο </th>
  </tr>
  <tr class="adminrow">
    <td>'. $count0 .'</td>
    <td>'. $count1 .'</td>
    <td>'. $count2 .'</td>
    <td>'. $count3 .'</td>
    <td>'. $count4 .'</td>
    <td>'. $count5 .'</td>
    <td>'. $count6 .'</td>
    <td>'. $total .'</td>
  </tr>
  <tr class="adminrow">
    <td>'. $rate0 .'</td>
    <td>'. $rate1 .'</td>
    <td>'. $rate2 .'</td>

```

```
        <td>' . $rate3 . '%</td>
        <td>' . $rate4 . '%</td>
        <td>' . $rate5 . '%</td>
        <td>' . $rate6 . '%</td>
        <td>    -          </td>
    </tr>
    <tr class="adminrow">
        <td>' . $count0 * 5 . '€</td>
        <td>' . $count1 * 5 . '€</td>
        <td>' . $count2 * 5 . '€</td>
        <td>' . $count3 * 5 . '€</td>
        <td>' . $count4 * 5 . '€</td>
        <td>' . $count5 * 5 . '€</td>
        <td>' . $count6 * 5 . '€</td>
        <td>' . $total  * 5 . '€</td>
    </tr>
</table>';
return $ret;
}
?>
```



## ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα πτυχιακή εργασία αναλύθηκε και υλοποιήθηκε το πρόβλημα δημιουργίας μια διαδικτυακής εφαρμογής η οποία θα επέτρεπε σε μία επιχείρηση πολύ-κινηματογράφου να προσφέρει τις υπηρεσίες της μέσω της ιστοσελίδας της. Δημιουργήθηκε μία δυναμική ιστοσελίδα με PHP, MySQL, HTML, CSS και JavaScript η οποία θα περιελάμβανε εργαλεία που θα επέτρεπαν την παροχή υπηρεσιών αγοράς και κράτησης εισιτηρίων.

Πιο συγκεκριμένα αναπτύχθηκαν εργαλεία PHP που επιτρέπουν την δυναμική μεταφορά δεδομένων από τον ιστότοπο σε μία βάση δεδομένων MySQL στην οποία έχουν διαφορετική πρόσβαση τρία επίπεδα χρηστών. Ο «Διαχειριστής» που η αρμοδιότητα του είναι να προσθέτει πληροφορίες για τις ταινίες και τις προβολές, ο «Υπάλληλος» ο οποίος είναι σε θέση να εξυπηρετεί τους πελάτες της επιχείρησης και το «Μέλος» που έχει δικαίωμα να γίνει οποιοσδήποτε πελάτης ώστε να απολαμβάνει τα οφέλη της εφαρμογής. Ωστόσο οι απλοί επισκέπτες περιορίζονται στην απλή άντληση πληροφοριών που αφορούν ημερομηνίες και ώρες προβολών.

Εξακριβώνοντας την ταυτότητα του οι χρήστες αποκτούν πρόσβαση στο μενού πλοήγησης για το επίπεδο χρήστη που τους έχει αντιστοιχηθεί από τον «Διαχειριστή» και μέσω αυτού είναι σε θέση να εκμεταλλευτούν τα εργαλεία που υλοποιήθηκαν ανάλογα με τις αρμοδιότητες και τα ενδιαφέροντα του κάθε επιπέδου. Η σχεδίαση έγινε με τέτοιο τρόπο ώστε η πλοήγηση στον ιστότοπο να είναι εύχρηστη και λειτουργική για κάθε χρήστη, επιτρέποντας την μεταφορά ανάμεσα στα εργαλεία μέσω συνδέσμων. Τα εργαλεία υλοποιήθηκαν μέσω διαδραστικών φορμών που έχουν ως έξοδο μηνύματα ειδοποίησης ανάλογα με το αποτέλεσμα του αιτήματος ενώ δεν παραλήφθηκαν, σε διακριτικό σημείο, βοηθητικά μηνύματα οδηγίων.

Όλα τα παραπάνω επιτεύχθηκαν μέσω γνώσεων που αποκτήθηκαν από το πρόγραμμα σπουδών, οι οποίες συνδιάστηκαν με γνώσεις από έρευνα βιβλιογραφίας και διαδικτύου. Αποτέλεσμα του παραπάνω συνδιασμού ήταν η υλοποίηση μίας ολοκληρωμένης εφαρμογής διαχείρισης και προσφοράς υπηρεσιών που επιτρέπει σε έναν πολύ-κινηματογράφο να εισέλθει στον χώρο του διαδικτυακού εμπορίου. Σε ένα τέτοιο έργο όμως τα όρια των δυνατοτήτων δεν είναι εμφανή, για αυτό τον λόγο η ανάπτυξη του κώδικα έγινε με τρόπο τέτοιο ώστε η πιθανή συμπλήρωση λειτουργιών και η συντήρηση να είναι εφικτή.

Μελλοντική προσθήκη στην εφαρμογή θα μπορούσε να είναι ένα σύστημα βαθμολογίας από τους χρήστες για την κάθε ταινία. Μεταβένοντας με τον αντίστοιχο σύνδεσμο στην ξεχωριστή σελίδα για κάθε ταινία θα μπορούσε να υπάρχει η βαθμολογία της στο IMDb μαζί με την βαθμολογία που έχουν δώσει οι χρήστες της ιστοσελίδας. Επίσης θα μπορούσε να υπάρχει κάτω από τα στοιχεία της ταινίας ένας χώρος σχολίων ώστε να ανταλλάζουν πληροφορίες μεταξύ τους οι χρήστες.

Επίσης μία ενδιαφέρουσα λειτουργία θα ήταν η εκτύπωση της κάρτας και των διαδικτυακών εισιτηρίων να ήταν εφικτά να γίνουν από τον ίδιο εφόσον έχει γίνει η ταυτοποίηση των στοιχείων με την επιχείρηση. Έτσι το «Μέλος» θα μπορούσε εξοφλώντας μία κράτηση ή αγοράζοντας εισιτήρια, είτε να εκτυπώσει το εισιτήριο του από το σπίτι, είτε να υπάρχει ένα είδος αυτόματου πωλητή που θα σαρώνει την κάρτα μέλους και θα εκτυπώνει τα πληρωμένα εισιτήρια. Στο σύστημα που υλοποιήθηκε υπάρχει

διαθέσιμη λειτουργία ελέγχου εγκυρότητας ενός εισιτηρίου άρα είναι εφικτή η ύπαρξη μίας τέτοιας δυνατότητας.

Ομοίως ενδιαφέρουσα θα ήταν μία λειτουργία που θα επέτρεπε στον οποιονδήποτε «Επισκέπτη» να συνδεθεί ζωντανά και δωρεάν σε μία υπηρεσία της εφαρμογής που θα αφορούσε ανταλλαγή άμεσων μηνυμάτων με έναν «Υπάλληλο» ώστε να ζητήσει περισσότερες πληροφορίες, οδηγίες στην χρήση του ιστότοπου ή ακόμα και να προβεί σε κράτηση θέσεων για μία προβολή.

Ακόμη εφόσον οι υπάλληλοι είναι ήδη μέλοι θα μπορούσαν να εκτυπώσουν και αυτοί μία κάρτα όπως τα μέλη η οποία θα ήταν απαραίτητη για την είσοδό τους στον ιστότοπο. Με την προϋπόθεση ότι αυτές οι κάρτες θα έμεναν στο κατάστημα, θα μπορούσαν να δοθούν δυνατότητες ελέγχου στον «Διαχειριστή» όσον αφορά τις ώρες προσέλευσης των υπαλλήλων για εργασία. Θα μπορούσε ακόμη να μην είναι δυνατή η είσοδος των υπαλλήλων στην εφαρμογή αν δεν είναι ημέρα και ώρα απασχόλησης στο κατάστημα.

Τέλος, η σημαντικότερη ίσως συμπληρωματική λειτουργία που είναι εφικτή είναι η ανάπτυξη και συμπλήρωση κώδικα της εφαρμογής, η οποία θα είσαγε ίσως ένα παραπάνω επίπεδο χρήστη από τον διαχειριστή και εργαλεία τέτοια ώστε τα περιεχόμενα και η εμφάνιση της εφαρμογής να ελέγχονται από τον ιστότοπό της. Έτσι θα ήταν δυνατή η χρήση της από οποιοδήποτε κατάστημα πολύ-κινηματογράφου, επιτρέποντας την προσθήκη άλλου λογότυπου, την επεξεργασία της αρχικής σελίδας και της σελίδας «Επικοινωνία», τον έλεγχο του διαφημιστικού χώρου, ακόμη και τον έλεγχο της ολοκληρωτικής εμφάνισης της ιστοσελίδας.

Αναμφίβολα λοιπόν τα όρια των γλωσσών προγραμματισμού που αφορούν την ανάπτυξη διαδικτυακών εφαρμογών περιορίζονται μόνο από την φαντασία του προγραμματιστή αλλά οι δυνατότητές τους παραμένουν απεριόριστες και ξεφεύγουν από τα όρια αυτής της πτυχιακής εργασίας. Με την απόκτηση και την αξιοποίηση γνώσεων, είτε ακαδημαϊκά, είτε ερευνητικά, όλα είναι δυνατά.

## ΚΕΦΑΛΑΙΟ 7 ΒΙΒΛΙΟΓΡΑΦΙΑ

### 7.1 Βιβλία

#### 7.1.1 Ελληνική Βιβλιογραφία

- Melonie, Julie C., (2008). Μάθετε PHP, MySQL και Apache Όλα σε ένα. 4<sup>η</sup> έκδοση. Εκδόσεις: Γκιούρδας Μ.
- Luke Welling & Laura Thomson, Μετάφραση: Γκλάβα Μαρία (2006). Ανάπτυξη Web Εφαρμογών με PHP και MySQL. Τρίτη Έκδοση. Εκδόσεις: Γκιούρδας Μ.
- Lemay Laura, Colburn Rafe, (2011). Πλήρες Εγχειρίδιο της HTML5 & CSS, 6<sup>η</sup> Έκδοση. Εκδόσεις: Χ. ΓΚΙΟΥΡΔΑ & ΣΙΑ ΕΕ.

#### 7.1.2 Ξένη Βιβλιογραφία

- K. Tatroe, P. MacIntyre, R. Lerdorf, (2013). Programing PHP. 3<sup>rd</sup> Edition. Εκδόσεις: Ο'Reilly
- Paul DeBois, (2002). MySQL Cookbook Εκδόσεις: Ο'Reilly
- David Sawyer McFarland, (2012). CSS3, The missing manual . 3<sup>rd</sup> Edition. Εκδόσεις: Ο'Reilly

### 7.2 Πηγές στο Διαδίκτυο

- <http://www.w3schools.com/html/>
- <http://www.w3schools.com/css/>
- <http://www.w3schools.com/php/>
- <http://www.w3schools.com/js/>
- [http://www.w3schools.com/php/php\\_mysql\\_intro.asp](http://www.w3schools.com/php/php_mysql_intro.asp)
- <http://php.net/manual/en/>
- <http://dev.mysql.com/doc/>
- [http://www.phpmyadmin.net/home\\_page/docs.php](http://www.phpmyadmin.net/home_page/docs.php)

#### Ξένος Κώδικας

Τύπος: Κώδικας CSS3 - Κίνηση

Πηγή: <http://daneden.github.io/animate.css/>

Δικαιώματα: Licensed under the MIT license  
<http://opensource.org/licenses/MIT>

Τύπος: Ενσωμάτωση Γραμματοσειρών OpenSans και Roboto

Πηγή: <https://www.google.com/fonts>

Δικαιώματα: Licensed under the SIL Open Font License 1.1  
[http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&id=OFL\\_web](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL_web)