

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ****ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ****ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ****Πτυχιακή Εργασία**

Τίτλος Πτυχιακής Εργασίας	Ολοκληρωμένη διαδικτυακή Εφαρμογή κινηματογραφικών προτάσεων και διαχείρισης ηλεκτρονικών εισιτηρίων Complete online Application for movie recommendations and e-ticket management
Όνοματεπώνυμο Φοιτητή	Κωνσταντίνος Ρίζος
Πατρώνυμο	Παναγιώτης
Αριθμός Μητρώου	Π17110
Επιβλέπων	Κωνσταντίνα Χρυσafiάδη, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης: Σεπτέμβριος 2024

Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς. Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την Επίκουρο Καθηγήτρια Κωνσταντίνα Χρυσafiάδη για την θερμή της υποστήριξη κατά την διάρκεια της πτυχιακής μου εργασίας. Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς για όλες τις γνώσεις και τα εφόδια που μου μετέδωσαν όλα αυτά τα χρόνια. Χωρίς αυτούς δεν θα είχα φτάσει στο επίπεδο όπου βρίσκομαι σήμερα. Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για όλη αυτή την ψυχολογική υποστήριξη αλλά και την ενθάρρυνση που μου μετέφεραν για να φτάσω στο τέλος των σπουδών μου και να με δουν να εξελίσσομαι ως προπτυχιακός φοιτητής.

Περίληψη

Στην σημερινή κοινωνία ο μέσος άνθρωπος έχει ιδιαίτερα βαρύ και φορτωμένο πρόγραμμα λόγω των γρήγορων ρυθμών ζωής αλλά και του αυξημένου ανταγωνισμού που επικρατεί στην αγορά εργασίας. Για αυτό τον λόγο αναζητεί συνέχεια τρόπους για να ξεφύγει από την ρουτίνα αλλά και την δύσκολη καθημερινότητα. Ο κάθε άνθρωπος το επιδιώκει αυτό με διαφορετική προσέγγιση: μερικοί επιλέγουν να αξιοποιήσουν τον ελεύθερο τους χρόνο κάνοντας κάποιο άθλημα ή απλά πηγαίνοντας γυμναστήριο, άλλοι βρίσκουν συντροφιά στις σελίδες ενός βιβλίου και αρκετοί περνούν ευχάριστα τον χρόνο τους παρακολουθώντας ταινίες και σειρές στην τηλεόραση. Τα τελευταία χρόνια έχει αναδειχθεί αρκετά η έβδομη τέχνη. Ο λόγος γίνεται για τον χώρο του κινηματογράφου. Παρατηρείται ότι όλο και περισσότερος κόσμος σπεύδει στους κινηματογράφους της περιοχής του για να παρακολουθήσει ταινίες. Η κινηματογραφική βιομηχανία έχει κάνει τεράστιο άλμα τα τελευταία χρόνια και αυτό οφείλεται στην ζήτηση του κοινού για νέα αλλά και ποιοτικά έργα. Η ποιότητα των ταινιών έχει ανέβει κατακόρυφα και οι προϋπολογισμοί αυτών αγγίζουν νούμερα που «ρίχνουν σαγόνια» .

Για αυτό τον λόγο, για την πτυχιακή μου εργασία, αποφάσισα να αναπτύξω μια web εφαρμογή που θα προτείνει στον χρήστη ταινίες ανάλογα με τις προτιμήσεις του αλλά θα του δίνει και την δυνατότητα να αγοράσει εισιτήριο για να τις παρακολουθήσει σε όποιον κινηματογράφο επιθυμεί ο ίδιος. Η εφαρμογή αυτή αναπτύχθηκε σε Python και πιο συγκεκριμένα χρησιμοποιεί το microframework του Flask. Το όνομά της είναι Flix και παρουσιάζει στον χρήστη όλες τις διαθέσιμες ταινίες που παίζονται στους κινηματογράφους την αντίστοιχη περίοδο. Η επιλογή τους γίνεται με βάση τις προτιμήσεις του χρήστη για τα είδη ταινιών τα οποία και έχει δηλώσει. Η αγορά εισιτηρίων γίνεται online και η εφαρμογή είναι προσβάσιμη από οποιοδήποτε browser.

Θεωρώ πως η εφαρμογή αυτή έχει προοπτική με περαιτέρω προσθήκες, κυρίως στον τομέα του design, να φανεί χρήσιμη σε αρκετούς χρήστες διαφόρων ηλικιακών ομάδων. Πέρα από την ενίσχυση του design, υπάρχουν κι άλλες ιδέες που θα μπορούσαν να υλοποιηθούν σε επόμενη φάση, οι οποίες θ' αναπτυχθούν επαρκώς στην κατάλληλη ενότητα.

Abstract

In today's society, the average person has an especially heavy and packed schedule due to the fast pace of life and increased competition in the job market. For this reason, people are constantly looking for ways to escape routine and the challenges of daily life. Each person pursues this with a different approach: some choose to spend their free time engaging in sports or simply going to the gym, others find companionship in the pages of a book, and many enjoy spending their time watching movies and series on television. In recent years, the seventh art has gained considerable popularity. We are referring to the realm of cinema. It is noticeable that more and more people are heading to their local cinemas to watch movies. The film industry has taken a huge leap in recent years, thanks to public demand for new and quality works. Movie quality has skyrocketed, and the budgets reach jaw-dropping figures.

For this reason, for my thesis, I decided to develop a web application that will recommend movies to users based on their preferences and also allow them to purchase tickets to watch them at a cinema of their choice. This application was developed in Python and specifically uses the Flask microframework. Its name is Flix, and it presents users with all the available movies currently playing in cinemas. Movie selection is based on the user's stated genre preferences. Ticket purchases are made online, and the application is accessible from any browser.

I believe this application has potential, and with further additions, primarily in the area of design, it could prove useful to many users across different age groups. Beyond design enhancements, there are other ideas that could be implemented in a future phase, which will be thoroughly developed in the appropriate section.

Πίνακας Περιεχομένων

Copyright ©	ii
Ευχαριστίες	iii
Περίληψη	iv
Abstract	iv
Πίνακας Περιεχομένων	vi
Κατάλογος Εικόνων	vii
Στόχοι	1
Βιβλιογραφική Ανασκόπηση.....	2
Web Εφαρμογές.....	2
Εξατομίκευση	2
Εξατομίκευση και Εμπειρία Χρήστη	4
Η Εξατομίκευση και η Τεχνητή Νοημοσύνη στις Εφαρμογές	4
Περιγραφή Προβλήματος	5
Διαφοροποίηση του Flix από Παρόμοιες Εφαρμογές.....	6
Εφαρμογή Village Cinemas	7
Εφαρμογή Option Cinemas.....	8
Σύντομη Περιγραφή της Λύσης και του Λογισμικού που Αναπτύχθηκε	10
Ανάλυση Απαιτήσεων.....	11
Απαιτήσεις Εκτέλεσης Προγράμματος / Χρήσης	11
Χρήστες – Ρόλοι και Λειτουργίες	12
Σχεδιασμός	15
Αρχιτεκτονική Συστήματος	15
Τεχνολογίες που Χρησιμοποιήθηκαν	16
Παρουσίαση Εργασιών Εφαρμογής.....	17
Περιγραφή Βάσης Δεδομένων	27
Σενάρια χρήσης του Συστήματος με Screenshots.....	29
Εγχειρίδιο Χρήστη.....	34
Οφέλη που αναμένουμε να έχουμε από την λύση της Πτυχιακής	42
Συμπεράσματα.....	43
Βιβλιογραφία	45

Κατάλογος Εικόνων

Εικόνα 1 Εφαρμογή Village Cinemas(1)	7
Εικόνα 2 Εφαρμογή Village Cinemas(2)	7
Εικόνα 3 Εφαρμογή Option Cinemas(1)	8
Εικόνα 4 Εφαρμογή Option Cinemas(2)	9
Εικόνα 5 Τρόπος Εγκατάστασης Πακέτων	11
Εικόνα 6 Use Case Diagram	13
Εικόνα 7 Class Diagram	13
Εικόνα 8 Μοντέλο Πελάτη-Διακομιστή	15
Εικόνα 9 Index.html(1).....	17
Εικόνα 10 Index.html(2).....	18
Εικόνα 11 Index.html(3).....	18
Εικόνα 12 Register.html.....	19
Εικόνα 13 Register Function	19
Εικόνα 14 Login.html	20
Εικόνα 15 Login Function	20
Εικόνα 16 Settings Function.....	21
Εικόνα 17 Movies.html(1)	22
Εικόνα 18 Movies.html(2).....	22
Εικόνα 19 Movies.html(3).....	23
Εικόνα 20 Movies.html(4).....	23
Εικόνα 21 Movies Function	24
Εικόνα 22 My_bookings.html.....	25
Εικόνα 23 My_bookings Function.....	25
Εικόνα 24 Logout Function.....	25
Εικόνα 25 Διάγραμμα Ροής Λειτουργιών	26
Εικόνα 26 Διάγραμμα Βάσης Δεδομένων	27
Εικόνα 27 Φόρμα Δημιουργίας Λογαριασμού(User1)	29
Εικόνα 28 Φόρμα Δημιουργίας Λογαριασμού(User2)	29
Εικόνα 29 Λίστα Ταινιών(User1).....	30
Εικόνα 30 Λίστα Ταινιών(User2).....	30
Εικόνα 31 Επιλεγμένη Ταινία(User1).....	31
Εικόνα 32 Επιλεγμένη Ταινία(User2).....	32
Εικόνα 33 Κράτηση(User1)	33
Εικόνα 34 Κράτηση(User2)	33
Εικόνα 35 Αρχική Οθόνη(1).....	34
Εικόνα 36 Αρχική Οθόνη(2).....	35
Εικόνα 37 Λίστα ταινιών	36
Εικόνα 38 Δημιουργία Λογαριασμού	37
Εικόνα 39 Είσοδος Χρήστη	37
Εικόνα 40 Καλωσόρισμα Χρήστη	37
Εικόνα 41 Επεξεργασία Προτιμήσεων Χρήστη	38
Εικόνα 42 Λίστα Ταινιών με Βάση τις Προτιμήσεις του Χρήστη.....	38
Εικόνα 43 Συμπλήρωση Στοιχείων(1)	39
Εικόνα 44 Συμπλήρωση Στοιχείων(2)	40
Εικόνα 45 Κρατήσεις Χρήστη	41
Εικόνα 46 Έξοδος.....	41

Στόχοι

Από την αρχή της εκπόνησης της εργασίας μου, είχα θέσει συγκεκριμένους στόχους. Τόσο για τον τρόπο υλοποίησής της, όσο και για το πως θα μεγιστοποιήσω την χρησιμότητα της εφαρμογής για τους διάφορους χρήστες που θα την χρησιμοποιούν. Παρακάτω προσδιορίζονται οι κύριοι στόχοι – άξονες, στους οποίους έχει βασιστεί ο σχεδιασμός και η ανάπτυξη του Flix.

- 1. Ανάπτυξη εύχρηστης εφαρμογής για αγορά εισιτηρίων κινηματογράφου:** Ο βασικός στόχος αυτής της διαδικτυακής εφαρμογής είναι η δημιουργία ενός «συστήματος» απλού και λειτουργικού, το οποίο θα δίνει την δυνατότητα στον χρήστη να αναζητά ταινίες ανάλογα με τις προτιμήσεις του καθώς και να αγοράζει εισιτήρια για τις παραπάνω για τις ώρες και τους κινηματογράφους που τον βολεύουν.
- 2. Φιλικό προς τον χρήστη περιβάλλον:** Η εφαρμογή θα απευθύνεται σε κάθε είδους χρήστη ανεξάρτητα από την εξοικείωση που θα έχει με την τεχνολογία και ανεξαρτήτου ηλικίας. Το περιβάλλον θα είναι απλό έως λιτό έτσι ώστε οι χρήστες να μην «μπερδεύονται» με τις περιπτές και αχρείαστες λειτουργίες.
- 3. Εξέλιξη χαρακτηριστικών που χρησιμοποιούνται από άλλες εφαρμογές αλλά και έμπνευση από αυτές:** Τέλος, είναι σημαντικό να σημειωθεί ότι επίσης βασικός στόχος είναι να υιοθετήσουμε χρήσιμες λειτουργίες από άλλες παρόμοιες εφαρμογές και να τις εξελίξουμε όσο το δυνατότερο γίνεται. Έτσι θα έχουμε ένα τελικό έργο που ως βάση θα έχει κάποια γνώριμα χαρακτηριστικά αλλά θα χρησιμοποιεί και νέα αξιοποιημένα στο έπακρο.
- 4. Αποδοτικότητα Εφαρμογής:** Ένας από τους βασικούς μου στόχους για το Flix είναι να προσφέρει ένα υψηλό επίπεδο αποδοτικότητας στους χρήστες. Η αλήθεια είναι ότι παρέχει κάποιες βασικές λειτουργίες. Το νόημα είναι να χρησιμοποιούνται αυτές οι λειτουργίες με τρόπο τέτοιο, ώστε να προσφέρουν τον μέγιστο βαθμό χρησιμότητας στους χρήστες της εφαρμογής.
- 5. Αποδοχή από το ευρύ κοινό:** Τέλος είναι πολύ σημαντικό για το Flix, να αγαπηθεί και αν ενστερνιστεί από τους χρήστες. Αυτό θα επιτευχθεί με την σωστή αξιοποίηση όλων των παραπάνω.

Βιβλιογραφική Ανασκόπηση

Web Εφαρμογές

Οι διαδικτυακές εφαρμογές είναι τύποι λογισμικού που βασίζονται σε μια αρχιτεκτονική πελάτη-διακομιστή, όπου η κεντρική λογική και τα δεδομένα αποθηκεύονται στον διακομιστή και προσπελάζονται από τον πελάτη μέσω ενός προγράμματος πλοήγησης, όπως είναι ο Chrome ή ο Firefox. Αυτές οι εφαρμογές δεν απαιτούν εγκατάσταση, καθώς λειτουργούν και ενημερώνονται αυτόματα, ενώ μπορούν να χρησιμοποιηθούν απλώς μέσω σύνδεσης στο διαδίκτυο, ανεξαρτήτως συσκευής ή λειτουργικού συστήματος. [12] Αυτή η χαρακτηριστική διαθεσιμότητα είναι ένα από τα βασικά πλεονεκτήματά τους. Επειδή είναι προσβάσιμες και εύχρηστες, είναι σημαντικό να είναι και επεκτάσιμες, ώστε να μπορούν να εξυπηρετούν μεγάλο αριθμό χρηστών και να διαχειρίζονται όγκο δεδομένων χωρίς να μειώνεται η απόδοσή τους. [13,14] Όσον αφορά το περιεχόμενό τους, διακρίνονται σε στατικές και δυναμικές: Οι στατικές πλατφόρμες προσφέρουν σταθερό περιεχόμενο, ενώ οι δυναμικές επιτρέπουν την τροποποίηση του περιεχομένου ανάλογα με τις ενέργειες των χρηστών. Αυτή η διάκριση οδηγεί σε διαφορετικές στρατηγικές ανάπτυξης, με τις δυναμικές πλατφόρμες να επιτρέπουν την ενσωμάτωση πιο περίπλοκων λειτουργιών, όπως η ταυτοποίηση χρήστη [12], που εισάγει την έννοια της ασφάλειας. Η ασφάλεια αποτελεί κρίσιμο στοιχείο για την ομαλή λειτουργία της εφαρμογής, καθώς η χρήση σύγχρονων τεχνολογιών ασφαλείας προστατεύει τα δεδομένα των χρηστών αλλά και των επιχειρήσεων, σε συνδυασμό με κατάλληλους μηχανισμούς αποθήκευσης αυτών των πληροφοριών. [13,14] Επίσης, αξίζει να δίνεται προσοχή στην απόδοση της κάθε πλατφόρμας. [12,13,14] Σύμφωνα με την προσέγγιση που υιοθετείται, ένα τέτοιο σύστημα απαιτεί και το σωστό σχεδιασμό για την ανάπτυξή του. Υπάρχουν διάφορες μεθοδολογίες για την περιγραφή αυτής της διαδικασίας, δηλαδή του κύκλου ζωής ενός λογισμικού, όπως το “Waterfall” μοντέλο και το “Scrum,” τα οποία έχουν σημαντικές και λιγότερο σημαντικές διαφορές. Συνήθως, ο κύκλος ζωής περιλαμβάνει τη συλλογή και ανάλυση των απαιτήσεων των χρηστών, τον καθορισμό σαφών στόχων, το σχεδιασμό των διεπαφών και την επιλογή της κατάλληλης αρχιτεκτονικής, την υλοποίηση, τις δοκιμές και, τελικά, τη συντήρηση ή την περαιτέρω εξέλιξη του λογισμικού. [15]

Εξατομίκευση

Η εξατομίκευση μιας ιστοσελίδας είναι η διαδικασία δημιουργίας εξατομικευμένων εμπειριών για τους επισκέπτες της. Αντί να προσφέρει μια γενική, ευρεία εμπειρία, η εξατομίκευση μιας ιστοσελίδας επιτρέπει στις εταιρείες να παρουσιάζουν μοναδικές εμπειρίες προσαρμοσμένες στις ανάγκες και τις επιθυμίες των επισκεπτών. Η εξατομίκευση δεν είναι καθόλου καινούργια έννοια για τις επιχειρήσεις. Οι σερβιτόροι συχνά χαιρετούν επαναλαμβανόμενους πελάτες με το όνομά τους και γνωρίζουν ακόμα και την “συνηθισμένη παραγγελία” τους. Οι καταστηματάρχες μπορούν να χρησιμοποιήσουν οπτικές ενδείξεις για να παρέχουν εξατομικευμένες εμπειρίες — ένα κουρασμένο ζευγάρι με τρία παιδιά πιθανότατα θέλει να πάρει αυτό που χρειάζεται και να φύγει γρήγορα, ενώ ένας μοναχικός πελάτης που περιηγείται αργά στους διαδρόμους μπορεί να είναι ανοιχτός για συζήτηση. Το νόημα είναι ότι στον πραγματικό κόσμο, η εξατομίκευση είναι εξαιρετικά συνηθισμένη.

Οι εταιρείες που προσφέρουν εξατομικευμένες εμπειρίες στους πελάτες τους υιοθετούν εργαλεία εξατομίκευσης και λογισμικό εξατομίκευσης για να βελτιστοποιήσουν την πορεία του χρήστη. Αυτά τα εργαλεία εξατομίκευσης συνδυάζουν τα συλλεγμένα δεδομένα μέσω μιας πλατφόρμας δεδομένων πελατών (CDP) μαζί με τεχνητή νοημοσύνη και μηχανική μάθηση για να αξιολογούν τη συμπεριφορά των πελατών σε πραγματικό χρόνο. Μερικά από τα δεδομένα των χρηστών που είναι ιδιαίτερα πολύτιμα για την παροχή εξατομικευμένης εμπειρίας περιλαμβάνουν:

- Ιστορικό περιήγησης
- Γεωγραφική τοποθεσία
- Δημογραφικά στοιχεία

- Προηγούμενες αγορές
- Δεδομένα πρώτου επιπέδου

Το λογισμικό εξατομίκευσης στη συνέχεια αναλύει αυτά τα δεδομένα και δημιουργεί μια εξατομικευμένη εμπειρία ιστοσελίδας που αξιοποιεί δυναμικό περιεχόμενο ώστε να ανταποκριθεί στις συγκεκριμένες ανάγκες του πελάτη. Αυτό μπορεί να γίνει είτε με εξατομίκευση βασισμένη σε κανόνες (όπου μια εξατομικευμένη εμπειρία ιστοσελίδας έχει σχεδιαστεί εκ των προτέρων) είτε με χρήση AI που αξιοποιεί αλγόριθμους για να δημιουργήσει νέα σημεία επαφής και να προβλέψει τη συμπεριφορά των χρηστών.

Αυτή η εξατομικευμένη εμπειρία ιστοσελίδας μπορεί να περιλαμβάνει:

- Παρουσίαση σχετικού περιεχομένου ιστοσελίδας, όπως μελέτες περίπτωσης και στοχευμένες σελίδες που έχουν σχεδιαστεί ειδικά για τις προτιμήσεις του πελάτη.
- Ανασχεδιασμό της εμπειρίας περιήγησης, ώστε οι επισκέπτες της ιστοσελίδας να έρχονται αντιμέτωποι με συγκεκριμένες προσκλήσεις σε ενέργειες και πιο φιλικές προς τον χρήστη λειτουργίες.
- Παροχή συστάσεων προϊόντων βασισμένων σε προηγούμενη συμπεριφορά πελατών και εμπειρίες παρόμοιων προφίλ χρηστών.

Το εξατομικευμένο μάρκετινγκ δεν περιορίζεται μόνο στην εμπειρία ιστού. Οι εκστρατείες ηλεκτρονικού μάρκετινγκ, οι σελίδες προορισμού και τα κοινωνικά δίκτυα μπορούν επίσης να προσαρμοστούν βάσει των ενεργειών που έχει κάνει ο χρήστης είτε στην ιστοσελίδα είτε μέσω δεδομένων που έχουν συλλεχθεί.

Ποια άλλη στρατηγική μάρκετινγκ θα μπορούσε να αποφέρει αύξηση 10-15% στα έσοδα μιας επιχείρησής; Η απάντηση, σύμφωνα με τη McKinsey [26], είναι η εξατομίκευση της ιστοσελίδας. Οι προσδοκίες των καταναλωτών έχουν μεταβληθεί στο σημείο που οι άνθρωποι αναμένουν μια προσωπική ψηφιακή εμπειρία που αντικατοπτρίζει το επίπεδο εξατομίκευσης που λαμβάνουν εκτός διαδικτύου. Γι' αυτό το 83% των καταναλωτών είναι επίσης πρόθυμοι να μοιραστούν τα δεδομένα τους [27], ώστε οι διαφημιστές και οι προγραμματιστές να τους προσφέρουν μια προσαρμοσμένη εμπειρία. Η λίστα με τα στατιστικά στοιχεία που αναδεικνύουν τα οφέλη της εξατομίκευσης ιστοσελίδων για τους διαφημιστές είναι μεγάλη, αλλά όλα καταλήγουν σε αυτό: η εξατομίκευση της ιστοσελίδας αυξάνει τα έσοδα και βελτιώνει τους δείκτες με τους παρακάτω τρόπους:

- Αύξηση ποσοστών μετατροπής: Δεδομένα από την Google έδειξαν ότι το 63% των χρηστών smartphone ήταν πιο πιθανό να αγοράσουν από μια εμπειρία ηλεκτρονικού εμπορίου μέσω κινητού ή εφαρμογής όταν η εμπειρία περιλάμβανε σχετικές συστάσεις προϊόντων.
- Δημιουργία αφοσίωσης: Η έρευνα "Twilio State of Personalization" διαπίστωσε ότι το 62% των επιχειρήσεων ανέφεραν αύξηση της αφοσίωσης των πελατών μετά την επένδυση σε εξατομικευμένες εμπειρίες.
- Αύξηση μέσης αξίας παραγγελίας (AOV): Η σύσταση προϊόντων, η μείωση των εμποδίων για μετατροπή και η χρήση δεδομένων πελατών για τη βελτίωση της εμπειρίας του πελάτη σημαίνουν όχι μόνο αύξηση των μετατροπών, αλλά και αύξηση της μέσης αξίας παραγγελίας κατά 10%.
- Μείωση ποσοστού αποχώρησης: Οι χρήστες είναι πολύ λιγότερο πιθανό να εγκαταλείψουν την ιστοσελίδα σας μετά από μια αλληλεπίδραση αν τους υποδέχεται σχετικό περιεχόμενο.

Συνολικά, η επένδυση στην εξατομίκευση ιστοσελίδων είναι επένδυση για παραμονή στον ανταγωνισμό: σύμφωνα με την Twilio, το 75% των ηγετών επιχειρήσεων δηλώνει πλέον ότι η εξατομίκευση είναι «απαραίτητη» [28]. Ωστόσο, η κούρσα για την εξατομίκευση, ή ακόμα και την υπερεξατομίκευση, δεν έχει κερδηθεί ακόμα. Παρόλο που οι διαφημιστές και οι διευθυντές δείχνουν μεγάλο ενδιαφέρον, οι πελάτες συχνά δεν εντυπωσιάζονται. Επιπλέον, μόνο το 9% των στελεχών σε πρόσφατη έρευνα σημείωσε ότι έχει επιτύχει πλήρη υλοποίηση των προσπαθειών εξατομίκευσής τους. Τα δεδομένα από όλες τις βασικές έρευνες, όπως αυτές των Google, McKinsey, και Twilio, καταδεικνύουν ότι οι πελάτες επιθυμούν την εξατομίκευση και είναι πρόθυμοι να σας δώσουν τα δεδομένα τους, αρκεί να τα διαχειρίζεστε με υπευθυνότητα [29]. Παρ' όλα αυτά, οι καταναλωτές αναφέρουν ότι δεν βλέπουν σημαντική βελτίωση.

Εξατομίκευση και Εμπειρία Χρήστη

Η εξατομίκευση και η ευφυΐα των σύγχρονων συστημάτων έχουν συμβάλει καθοριστικά στη βελτίωση της συνολικής εμπειρίας χρήστη (user experience) [16]. Τα συστήματα αυτά διαθέτουν τη δυνατότητα να διαμορφώνουν το περιβάλλον χρήσης με βάση τις προσωπικές προτιμήσεις και ανάγκες του χρήστη, καθιστώντας την αλληλεπίδραση με την εφαρμογή ή την πλατφόρμα πιο άμεση, προσιτή και αποδοτική [17]. Η πολυπλοκότητα μειώνεται αισθητά, καθώς διαδικασίες που ενδέχεται να είναι δύσκολες ή χρονοβόρες απλοποιούνται, επιτρέποντας καλύτερη ανταπόκριση στις εξατομικευμένες ανάγκες κάθε χρήστη [18]. Η εξατομίκευση δεν αποτελεί απλώς μια τεχνική, αλλά ενισχύει την αίσθηση ότι το σύστημα "καταλαβαίνει" τον χρήστη, δημιουργώντας μια πιο προσωπική και οικεία εμπειρία [19]. Έτσι, η συνολική ικανοποίηση των χρηστών αυξάνεται, καθώς νιώθουν ότι το σύστημα δεν είναι απλά ένα εργαλείο, αλλά ένας "συνεργάτης" που μπορεί να τους παρέχει αυτό που χρειάζονται με τον καλύτερο δυνατό τρόπο [20].

Η Εξατομίκευση και η Τεχνητή Νοημοσύνη στις Εφαρμογές

Η εξατομίκευση στις εφαρμογές και η Τεχνητή Νοημοσύνη (TN) αποτελούν δύο από τα πλέον αναδυόμενα και σημαντικά πεδία στη σύγχρονη τεχνολογία. Στην ψηφιακή εποχή, η εξατομίκευση παρέχει στους χρήστες μοναδικές εμπειρίες, προσαρμοσμένες στις ατομικές τους ανάγκες, προτιμήσεις και συμπεριφορές [21]. Η Τεχνητή Νοημοσύνη διαδραματίζει κρίσιμο ρόλο σε αυτή τη διαδικασία, καθώς επιτρέπει την ανάλυση μεγάλου όγκου δεδομένων και την αναγνώριση μοτίβων, με σκοπό να δημιουργήσει εξατομικευμένες προτάσεις και λύσεις. Ένα από τα κύρια πεδία εφαρμογής αυτής της τεχνολογίας είναι οι πλατφόρμες ηλεκτρονικού εμπορίου, όπου οι αλγόριθμοι TN προσφέρουν προτάσεις προϊόντων βάσει του ιστορικού αγορών και πλοήγησης του χρήστη [21].

Ένα ακόμα σημαντικό πεδίο εφαρμογής της εξατομίκευσης με TN είναι η εκπαίδευση, όπου έξυπνα συστήματα μάθησης προσαρμόζουν το υλικό στον ρυθμό και τις ανάγκες κάθε μαθητή. Οι εκπαιδευτικές πλατφόρμες χρησιμοποιούν μηχανική μάθηση και ανάλυση συμπεριφοράς για να παρέχουν προσαρμοσμένες προτάσεις μάθησης, ενισχύοντας την αποτελεσματικότητα της διαδικασίας μάθησης [22]. Αυτή η χρήση της TN δημιουργεί ένα πιο προσωπικό και αλληλεπιδραστικό εκπαιδευτικό περιβάλλον, το οποίο ευνοεί την εξέλιξη των μαθητών. Επιπλέον, η εξατομίκευση στις εφαρμογές υγείας προσφέρει σημαντικά οφέλη, καθώς οι πλατφόρμες TN μπορούν να προσαρμόσουν προγράμματα υγείας και ευεξίας στις ατομικές ανάγκες των ασθενών [23]. Εφαρμογές όπως οι ψηφιακοί βοηθοί υγείας χρησιμοποιούν δεδομένα που συλλέγονται από φορητές συσκευές και αισθητήρες για να παρέχουν εξατομικευμένες συμβουλές υγείας, δίνοντας έτσι τη δυνατότητα στους χρήστες να παρακολουθούν την υγεία τους πιο αποτελεσματικά.

Τέλος, η τεχνητή νοημοσύνη και η εξατομίκευση έχουν ανεβάσει επίπεδο τις εφαρμογές των κινηματογράφων προσφέροντας μια ιδανική εμπειρία. Μέσω αυτών των εφαρμογών οι χρήστες μπορούν να βρουν ταινίες που ταιριάζει στα γούστα τους αλλά και να αγοράσουν τα εισιτήριά τους ηλεκτρονικά. Αυτές οι εφαρμογές αναπροσαρμόζονται συνεχώς, λαμβάνοντας υπόψη τις προτιμήσεις

κάθε χρήστη, προσφέροντας εξατομίκευση σε πραγματικό χρόνο [24]. Επίσης αναλύουν δεδομένα του χρήστη όπως αγαπημένες κατηγορίες ταινιών, αγαπημένους ηθοποιούς, προτείνοντας συνεχώς νέα πράγματα για την μεγιστοποίηση ψυχαγωγίας του χρήστη [25].

Περιγραφή Προβλήματος

Σε αυτή την ενότητα, θα υπάρξει αναλυτική περιγραφή του προβλήματος που καλείται να αντιμετωπίσει η εφαρμογή μου, καθώς θα αναλυθεί και η φιλοσοφία του Flix.

Η καθημερινότητα των ανθρώπων γίνεται συνεχώς και πιο απαιτητική τα τελευταία χρόνια, εξαιτίας των πολλών υποχρεώσεων και του αυξημένου όγκου εργασίας. Είναι κυρίως οι απαιτήσεις της σύγχρονης εποχής αυτές, που τους ωθούν να έχουν καθημερινότητες που δημιουργούν μια επαναλαμβανόμενη ρουτίνα. Υπάρχουν άτομα που για να ξεφύγουν από αυτή την «μάστιγα», ασχολούνται με διάφορα «χόμπι». Από αυτά τα χόμπι αποσπούν ψυχική ικανοποίηση. Μία από αυτές τις ψυχαγωγικές ασχολίες είναι ο κινηματογράφος. Ο χρόνος όμως είναι περιορισμένος, συνεπώς δεν προλαβαίνουν όλοι να αγοράσουν το εισιτήριό τους εγκαίρως. Έτσι λοιπόν θα μπορούσαν όλοι να επωφεληθούν από την χρήση μίας διαδικτυακής εφαρμογής η οποία εξυπηρετεί ακριβώς αυτό τον σκοπό, την αγορά εισιτηρίων online. Ουσιαστικά το Flix δίνει την δυνατότητα στον χρήστη να αγοράσει το εισιτήριο για την ταινία που ο ίδιος επιθυμεί να παρακολουθήσει, εύκολα και γρήγορα από όπου και αν βρίσκεται. Έτσι αποφεύγονται και οι μεγάλες ουρές που δημιουργούνται έξω από τα σινεμά για την αγορά εισιτηρίων. Είναι πρακτικό από όλες τις απόψεις.

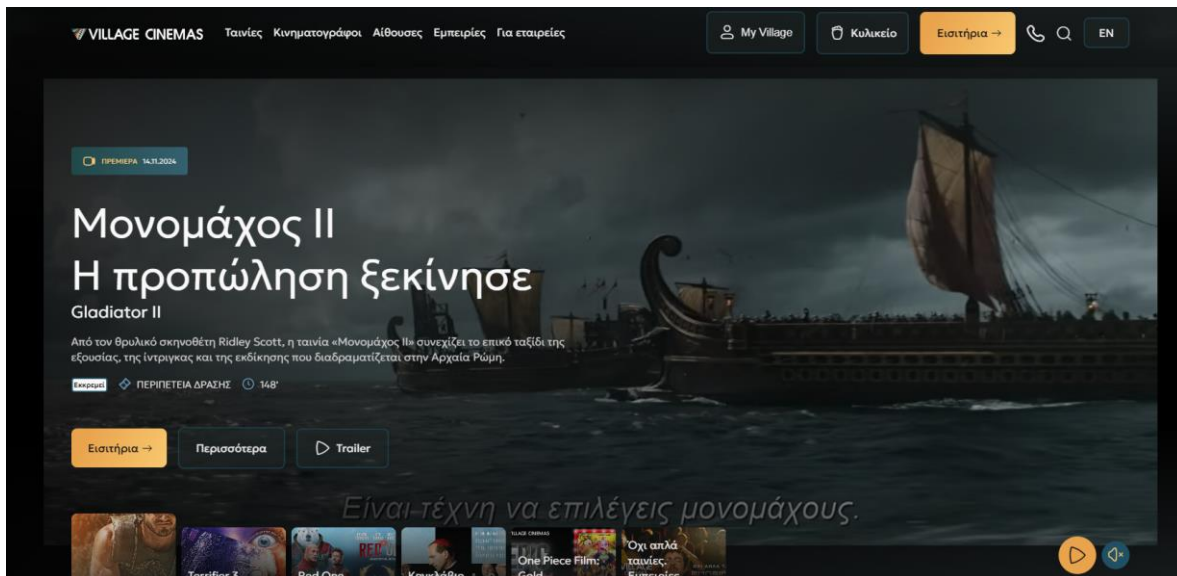
Είναι ευρέως γνωστό ότι υπάρχουν πολλές παρόμοιες εφαρμογές. Ωστόσο δεν υπάρχει καμία εφαρμογή που να λειτουργεί με βασικό γνώμονα τις προτιμήσεις του χρήστη. Οι περισσότερες επικεντρώνονται στον αισθητικό τομέα και παραμελούν το γεγονός ότι οι χρήστες αναζητούν φιλικές και με απλές λειτουργίες εφαρμογές. Έτσι σημείωσα τα παράπονα του κόσμου και τις αδυναμίες των άλλων εφαρμογών και προσπάθησα να φτιάξω μία νέα, όπου θα καλύπτει όλα τα παραπάνω κενά και θα παρουσιάζει ένα καινοτόμο περιβάλλον.

Διαφοροποίηση του Flix από Παρόμοιες Εφαρμογές

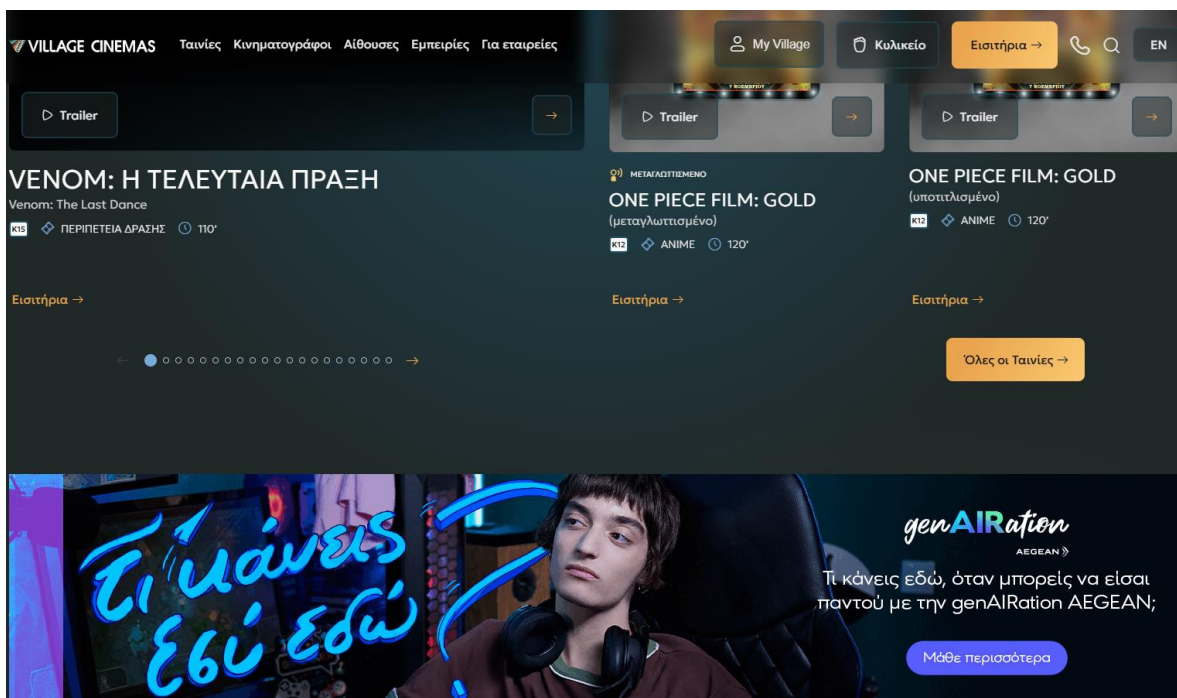
Ίσως αναρωτηθείτε για ποιο λόγο ανέλαβα την υλοποίηση του συγκεκριμένου project, από την στιγμή που υπάρχουν ήδη αρκετές παρόμοιες εφαρμογές. Είναι αλήθεια πως οι ήδη υπάρχουσες εφαρμογές προσφέρουν αμεσότητα, αναγνωρίζονται για το όμορφο καθώς και εύχρηστο Design και User Interface τους και σίγουρα βρίσκονται σε ένα πιο προχωρημένο στάδιο στο «development». Προφανώς, το Flix είναι αυτή την στιγμή, μία εφαρμογή που μπορεί να χρησιμοποιηθεί για να βοηθήσει πραγματικά τον χρήστη της, αλλά πιστεύω ότι με τις ιδέες μου για μελλοντικές επεκτάσεις, οι οποίες θα αναλυθούν σε επόμενη ενότητα και βασίζονται σε προαναφερόμενους άξονες, θα ανέβει το επίπεδο και θα προσφέρει έναν μοναδικό συνδυασμό ευκολίας πλοήγησης – εξατομίκευσης. Με τα παραπάνω ως δεδομένα, θα αναφέρω συγκεκριμένα σημεία διαφοροποίησης του Flix σε σχέση με αντίστοιχες εφαρμογές.

- 1. Εξατομίκευση:** Όπως ήδη γνωρίζουμε υπάρχουν ήδη αρκετές εφαρμογές που χρησιμοποιούνται για την αγορά εισιτηρίων για διάφορους κινηματογράφους (Village, Ster κτλ.). Αυτό όμως που λείπει από αυτές τις εφαρμογές είναι ένα πολύ βασικό χαρακτηριστικό, η εξατομίκευση. Με λίγα λόγια, ενώ όλες οι εφαρμογές δίνουν την δυνατότητα στον χρήστη να αγοράσει ένα εισιτήριο για μια συγκεκριμένη ταινία, καμία δεν του προτείνει ταινίες, ανάλογα με τις προτιμήσεις του. Εδώ έρχεται το Flix να καλύψει αυτή την «ανάγκη» με το πρωτόκολλο εξατομίκευσης που χρησιμοποιεί. Οι χρήστες δηλώνουν τα είδη ταινιών που προτιμούν και η εφαρμογή τους προτείνει ταινίες με βάση τις παραπάνω.
- 2. Απλή Διεπαφή Χρήστη(UI):** Μια απλή και άμεση διεπαφή χρήστη προσφέρει εύκολη προσπέλαση και περιήγηση στην εφαρμογή. Πιστεύω πως είναι πολύ σημαντικό αυτό το χαρακτηριστικό, αφού η εφαρμογή στοχεύει στην διευκόλυνση του χρήστη και όχι στην περαιτέρω ταλαιπωρία του. Έχω φροντίσει ιδιαίτερα να είναι τακτοποιημένο το «στήσιμο» του Flix για να είναι ξεκάθαρο στον χρήστη πως θα πραγματοποιεί την κάθε του ενέργεια. Παρατήρησα πως διάφορες παρόμοιες εφαρμογές δεν έχουν καταφέρει να πετύχουν αυτό τον στόχο, κυρίως επειδή έχουν προσπαθήσει να εφαρμόσουν διάφορες λειτουργίες, που κάνουν την περιήγηση πιο «χαοτική». Θα αναφερθούν παραδείγματα με screenshots παρακάτω.
- 3. Έμπνευση από την κλασική εφαρμογή αγοράς εισιτηρίων για κινηματογράφο:** Το Flix συνδυάζει ιδανικά τα χρήσιμα χαρακτηριστικά της κλασικής εφαρμογής που χρησιμοποιείται για την αγορά εισιτηρίων σε κινηματογράφους (πχ Village), με τις σύγχρονες δυνατότητες που προσφέρουν οι εξελιγμένοι browsers. Για παράδειγμα μπορεί ο χρήστης να ενδιαφέρεται για την αγορά εισιτηρίου αλλά να μην έχει καταλήξει ποια ταινία επιθυμεί να παρακολουθήσει. Έτσι, απλά δηλώνει τις προτιμήσεις του και το Flix αναλαμβάνει να του προτείνει τις αντίστοιχες ταινίες.
- 4. Απουσία Διαφημίσεων:** Στις περισσότερες εφαρμογές για αγορά εισιτηρίων online υπάρχουν third party διαφημίσεις. Με λίγα λόγια αυτές οι ιστοσελίδες πληρώνονται για να διαφημίζουν προϊόντα και υπηρεσίες άλλων εταιρειών. Αυτές οι διαφημίσεις αρκετές φορές κουράζουν τον χρήστη και του αποσπούν την προσοχή. Το Flix είναι Ad Free επειδή βασικός σκοπός του είναι να εξυπηρετήσει τον χρήστη με συγκεκριμένο τρόπο και να του κάνει την πλοήγηση εύκολη.

Εφαρμογή Village Cinemas



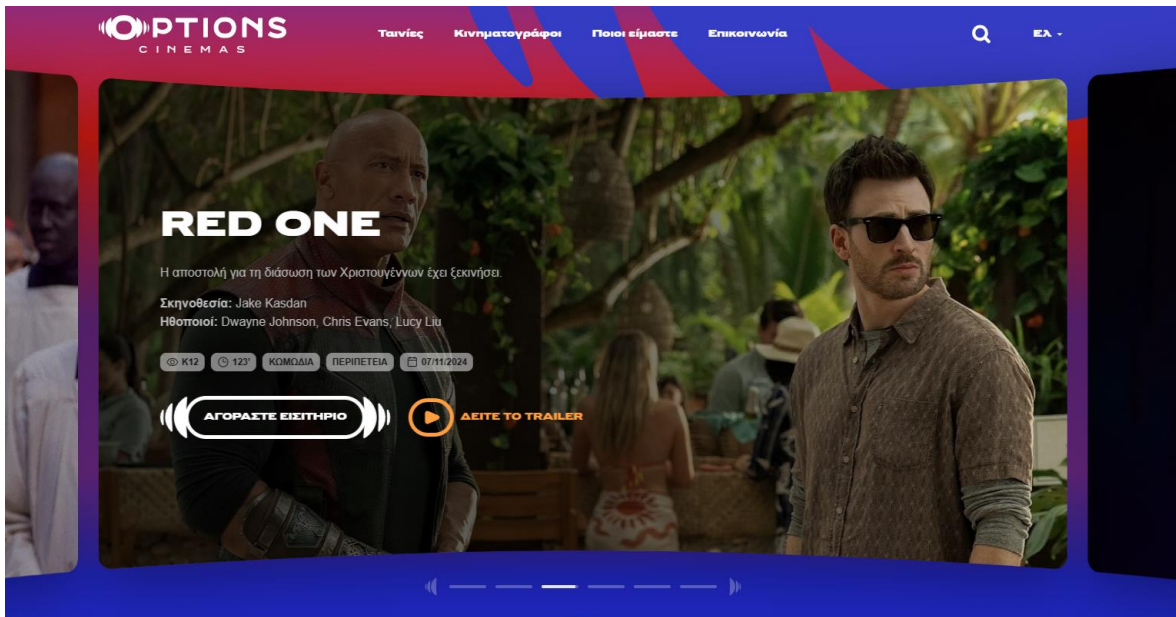
Εικόνα 1 Εφαρμογή Village Cinemas(1)



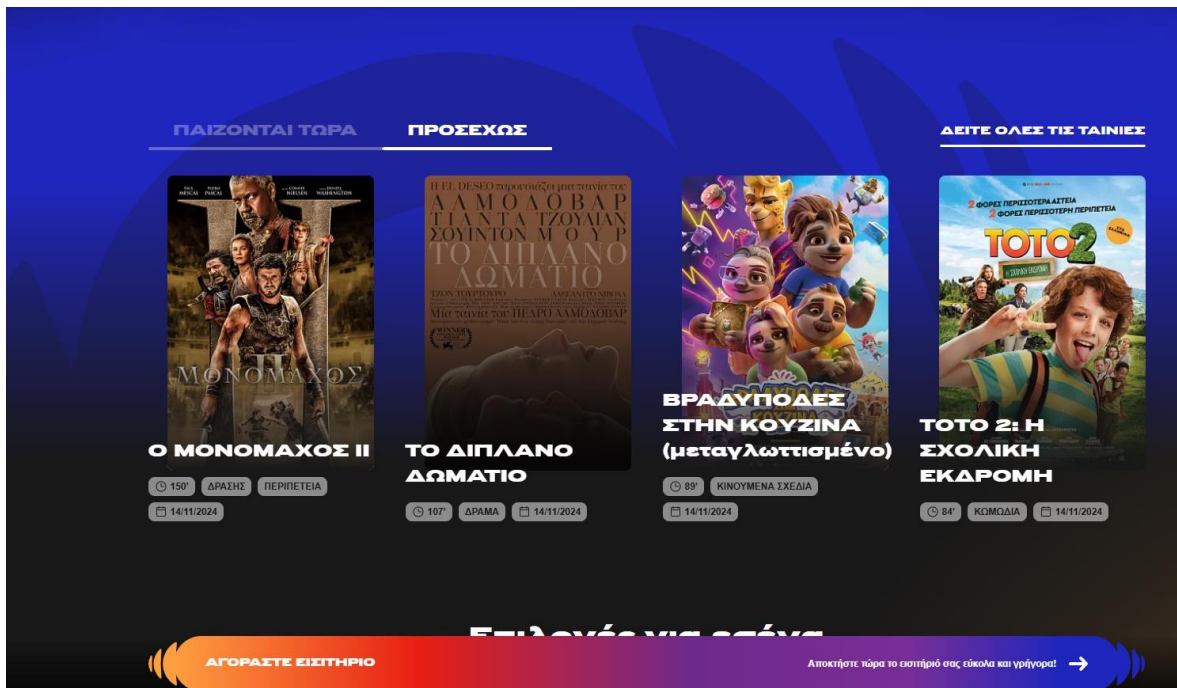
Εικόνα 2 Εφαρμογή Village Cinemas(2)

Στην παραπάνω διαδικτυακή εφαρμογή των Village Cinemas να μην υπάρχει όμορφο UI, αλλά παρατηρούμε πάρα πολλές λειτουργίες που ίσως κουράσουν και μπερδέψουν τον χρήστη. Παρατηρούμε επιλογές όπως «εμπειρίες», «για εταιρείες» κτλ. Αρκετές φορές οι χρήστες δεν έχουν την υπομονή αλλά ούτε τον διαθέσιμο χρόνο για να περιηγούνται σε μια εφαρμογή μέχρι να βρουν αυτό που αναζητούν. Με λίγα λόγια «αγανακτούν» και μπορεί στο τέλος η επιθυμία τους για την αγορά ενός εισιτηρίου να αποτελεί πλέον παρελθόν. Συνεπώς έχω φροντίσει το Flix ώστε να είναι λιτό και περιεκτικό ταυτόχρονα, δηλαδή να παρέχει τις βασικές λειτουργίες που θα χρειαστεί ο μέσος χρήστης. Ένα χαρακτηριστικό που εκνευρίζει αρκετό κόσμο είναι η ύπαρξη περιπτώσεων και μη σχετικών με το θέμα, διαφημίσεων. Για παράδειγμα στην τελευταία εικόνα βλέπουμε μια διαφήμιση που αναφέρεται στην αεροπορική εταιρεία της «Aegean» που δεν έχει καμία σχέση με τον χώρο του κινηματογράφου. Το Flix είναι μία «Adsfree» εφαρμογή που κύριο μέλημά της είναι να λύσει τα χέρια του χρήστη και όχι να τον «βομβαρδίζει» με μη σχετικά θέματα.

Εφαρμογή Option Cinemas



Εικόνα 3 Εφαρμογή Option Cinemas(1)



Εικόνα 4 Εφαρμογή Option Cinemas(2)

Στην παραπάνω διαδικτυακή εφαρμογή των Option Cinemas δεν υπάρχει καθόλου το χαρακτηριστικό της εξατομίκευσης, δηλαδή να προτείνει ταινίες στον χρήστη ανάλογα με τις προτιμήσεις του. Το Flix φροντίζει για το παραπάνω πρόβλημα με τον καλύτερο δυνατό τρόπο. Εξασφαλίζει ότι θα δοθεί η ευκαιρία στον χρήστη να διαλέξει ανάμεσα από μία γκάμα ταινιών που πληρούν τα γούστα του. Έτσι δεν χρειάζεται να περιηγούνται σε κατηγορίες ταινιών που δεν τους ενδιαφέρουν.

Σύντομη Περιγραφή της Λύσης και του Λογισμικού που Αναπτύχθηκε

Στις προηγούμενες ενότητες έχει αναλυθεί επαρκώς το πρόβλημα που έχω επιλέξει να λύσω με την ανάπτυξη του Flix. Σε αυτό το κεφάλαιο όμως, θα ξεκινήσουμε να ασχολούμαστε με την λύση που προσφέρεται.

Η συγκεκριμένη διαδικτυακή εφαρμογή είναι σχεδιασμένη έτσι ώστε να διευκολύνει την αλληλεπίδραση των χρηστών της με μια ψυχαγωγική δραστηριότητα και πιο συγκεκριμένα τον κινηματογράφο. Πλέον η αγορά εισιτηρίου online γίνεται «παιχνιδάκι». Απευθύνεται σε άτομα διαφόρων ηλικιών, ανεξαρτήτως εξοικείωσης με την τεχνολογία. Με λίγα λόγια, μπορεί να φανεί χρήσιμο σε οποιονδήποτε μπορεί να χρησιμοποιήσει έναν απλό browser. Ωστόσο, όπως έχει αναφερθεί και παραπάνω, το Flix είναι πιο αποδοτικό όταν οι χρήστες αξιοποιούν στο έπακρο τον εξατομικευμένο χαρακτήρα του.

Σε αυτή την παράγραφο, θα υπάρξει μια σύντομη ανάλυση για το λογισμικό που έχει αναπτυχθεί. Εξαιρώντας προς το παρόν την οθόνη Login/Register, η διαδικτυακή εφαρμογή είναι σχεδιασμένη με τέτοιο τρόπο ώστε να επιτρέπει στον χρήστη να βρίσκεται στο σημείο που ο ίδιος επιθυμεί με το πάτημα ενός «κλικ». Αυτό οφείλεται στην ενσωμάτωση μίας γραμμής που βρίσκεται στο πάνω μέρος της οθόνης (top navigation bar) η οποία διαχωρίζει τις κύριες λειτουργίες της εφαρμογής. Αρχικά ο χρήστης καλείται να δημιουργήσει έναν λογαριασμό (αν δεν έχει ήδη) στον οποίο δηλώνει το username, το password αλλά και τις προτιμήσεις του στα είδη ταινιών. Η εφαρμογή τον «καλωσορίζει» και στην συνέχεια του δίνει 2 επιλογές. Μπορεί να δει τις κρατήσεις του πατώντας το κουμπί «My bookings» ή να κλείσει εισιτήριο μέσω της επιλογής «Movies». Να σημειωθεί ότι μόλις πατηθεί η επιλογή «Movies» εμφανίζονται στον χρήστη ταινίες που ανήκουν στα είδη ο οποίος έχει δηλώσει στις προτιμήσεις του. Ο ίδιος επιλέγει ποια ταινία θέλει να παρακολουθήσει, σε ποιον κινηματογράφο, την ώρα αλλά και τις θέσεις που θέλει να κάτσει. Τέλος βάζει τα στοιχεία της κάρτας για να γίνει η online αγορά. Περισσότερες λεπτομέρειες θα δοθούν σε παρακάτω ενότητα.

Ανάλυση Απαιτήσεων

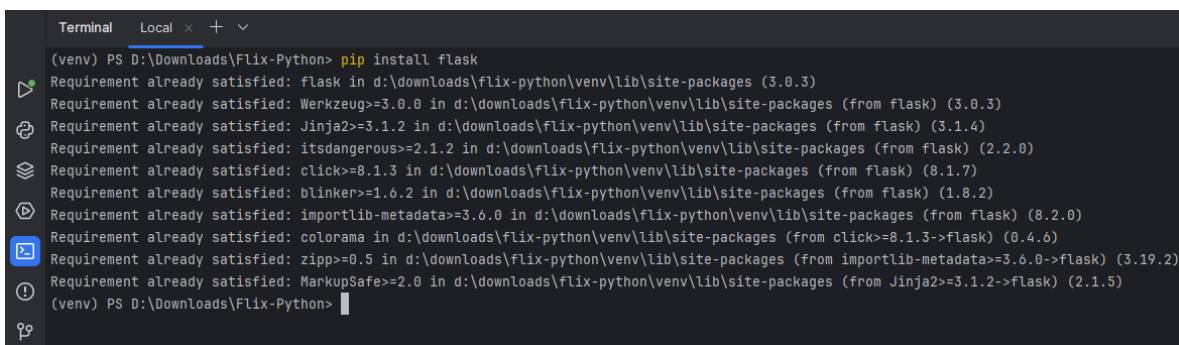
Σε αυτή την ενότητα, θα επισημάνω τις απαιτήσεις της εφαρμογής ως προς την εγκατάσταση και την χρήση της καθώς και τα απαραίτητα εργαλεία που χρειάζεται για να τρέξει αυτή η εφαρμογή σε περιβάλλον Python. Αρχικά να σημειωθεί ότι η εφαρμογή αυτή εκτός από localhost μπορεί να τρέξει και κανονικά σε οποιοδήποτε Online Server που είναι υπεύθυνος για το host διαδικτυακών εφαρμογών. Στην παρούσα φάση όμως θα δούμε πως μπορεί κάποιος να το τρέξει από τον προσωπικό του υπολογιστή(localhost).

Απαιτήσεις Εκτέλεσης Προγράμματος / Χρήσης

Αρχικά είναι απαραίτητη η εγκατάσταση ενός Python IDE(Integrated Development Environment). Εγώ προσωπικά χρησιμοποιώ το PyCharm Community της JetBrains. Το interpreter που χρησιμοποιήσα ήταν 3.9 έκδοση οπότε είναι απαραίτητη η εγκατάσταση αυτού ή ενός νεότερου μέσω του PyCharm. Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας κατά κύριο το microframework της Python, το Flask.

Το Flask είναι ένα microframework που βασίζεται στην Python για την ανάπτυξη εφαρμογών ιστού. Συνεπώς συνδυάζει Python με Html, CSS κτλ. Ταξινομείται ως microframework, καθότι δεν βασίζεται σε εξωτερικές βιβλιοθήκες για την εκτέλεση εργασιών. Επιπλέον, είναι ευέλικτο, ανεξάρτητο και γενικότερα προσφέρει περισσότερο έλεγχο στη διαδικασία ανάπτυξης. Εστιάζει στην απλότητα και το μινιμαλισμό και είναι κατάλληλο για single-page applications, καθώς και έρχεται με ένα ενσωματωμένο debugger που προσφέρει εικονικό debugging (εντοπισμό σφαλμάτων). Αρκετές γνωστές εταιρείες όπως οι Uber, Reddit, Netflix, Mozilla, Airbnb και Lyft χρησιμοποιούν το Flask.

Συνεπώς χρησιμοποιήθηκαν κάποια «πακέτα» που θα αναλύσουμε παρακάτω. Η εγκατάσταση ενός πακέτου γίνεται χρησιμοποιώντας την γραμμή εντολών του PyCharm γράφοντας την εντολή «pip install (όνομα πακέτου)», πχ pip install flask. Για να είναι δυνατή η εκτέλεση του προγράμματος πρέπει να γίνει η εγκατάσταση όλων των παρακάτω πακέτων: flash, flask-sqlalchemy, flask-login, flask-wtf και wtforms.



```
Terminal Local x + v
(venv) PS D:\Downloads\Flix-Python> pip install flask
Requirement already satisfied: flask in d:\downloads\flix-python\venv\lib\site-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in d:\downloads\flix-python\venv\lib\site-packages (from flask) (3.0.3)
Requirement already satisfied: Jinja2>=3.1.2 in d:\downloads\flix-python\venv\lib\site-packages (from flask) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in d:\downloads\flix-python\venv\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in d:\downloads\flix-python\venv\lib\site-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in d:\downloads\flix-python\venv\lib\site-packages (from flask) (1.8.2)
Requirement already satisfied: importlib-metadata>=3.6.0 in d:\downloads\flix-python\venv\lib\site-packages (from flask) (8.2.0)
Requirement already satisfied: colorama in d:\downloads\flix-python\venv\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: zipp>=0.5 in d:\downloads\flix-python\venv\lib\site-packages (from importlib-metadata>=3.6.0->flask) (3.19.2)
Requirement already satisfied: MarkupSafe>=2.0 in d:\downloads\flix-python\venv\lib\site-packages (from Jinja2>=3.1.2->flask) (2.1.5)
(venv) PS D:\Downloads\Flix-Python>
```

Εικόνα 5 Τρόπος Εγκατάστασης Πακέτων

Χρήστες – Ρόλοι και Λειτουργίες

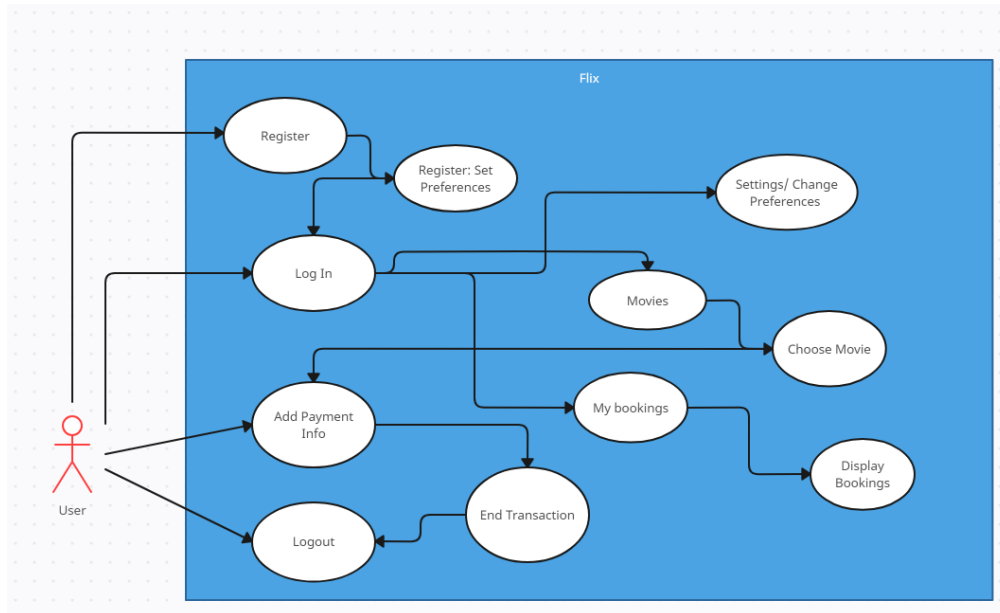
Αρχικά, το Flix εφόσον είναι μια διαδικτυακή εφαρμογή(ιστοσελίδα), μπορεί να τρέξει σε οποιοδήποτε browser. Για την πτυχιακή εργασία επέλεξα να το εκτελέσω μέσω του PyCharm ως localhost(τοπικά). Αν αποφασιστεί στο μέλλον, ότι αυτή η ιστοσελίδα θα «ανέβει» κανονικά στο διαδίκτυο είναι απαραίτητη η εύρεση ενός domain name και ενός παρόχου που θα είναι υπεύθυνος για το hosting του site. Με αυτό τον τρόπο το Flix θα είναι διαθέσιμο προς όλους.

Υπάρχει στη ουσία μόνο ένας τύπος χρήστη που χρησιμοποιεί την εφαρμογή, αυτός που θα μπει για να αγοράσει το εισιτήριο ή τα εισιτήρια για μία συγκεκριμένη ταινία. Εκτός από αυτόν, δεν υπάρχει κάποιος άλλος χρήστης-ρόλος(πχ administrator) που να έχει οριστεί. Επομένως, όλες οι λειτουργίες της εφαρμογής βρίσκονται στην διάθεση των τυπικών χρηστών της. Παρακάτω αναφέρονται όλες οι λειτουργίες που μπορεί να επιτελέσει ο χρήστης της εφαρμογής, ενώ θα αναλυθούν και πιο εκτενώς σε επόμενες ενότητες. Τέλος, υπάρχει ένα απλό Use Case Diagram και ένα Class Diagram για την πληρέστερη κατανόηση των λειτουργιών.

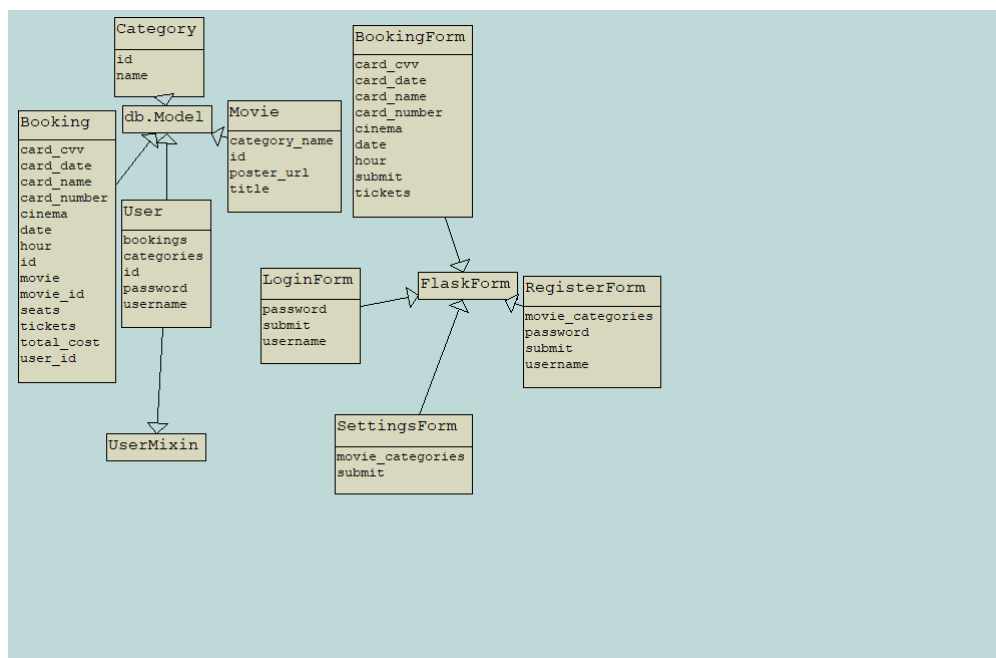
Οι λειτουργίες είναι οι ακόλουθες:

- **Login/Register:** Στην αρχική οθόνη το πρώτο πράγμα που παρατηρεί κανείς στο navigation bar είναι οι 2 επιλογές του Login και του Register. Αν ο χρήστης έχει ήδη λογαριασμό μπορεί απλά να πατήσει την επιλογή του Login και αφού εισάγει τα στοιχεία του σωστά(username και password), θα του εμφανιστούν κάποιες νέες επιλογές που θα αναλυθούν παρακάτω. Διαφορετικά, αν ο χρήστης δεν έχει λογαριασμό πρέπει να πατήσει το κουμπί Register και να εισάγει ένα username και ένα κωδικό που ο ίδιος επιθυμεί. Να σημειωθεί ότι ο κωδικός πρέπει να είναι τουλάχιστον 8 χαρακτήρες. Επίσης κατά την φάση του Register ο χρήστης δηλώνει τις προτιμήσεις του στις ταινίες.
- **Get Started:** Είναι ένα κουμπί στην αρχική οθόνη που μεταφέρει τον χρήστη κατευθείαν στην φόρμα του Register.
- **See more:** Είναι μία επιλογή στην αρχική οθόνη η οποία δείχνει στον χρήστη όλες τις διαθέσιμες ταινίες.
- **Subscribe:** Στο κάτω μέρος της αρχικής οθόνης ο χρήστης μπορεί να κάνει subscribe με το e-mail του για να μπει στο newsletter και να ενημερώνεται για νέες ταινίες αλλά και για τις ανακοινώσεις του κάθε κινηματογράφου.
- **Settings:** Η εφαρμογή με αυτή την επιλογή δίνει στον χρήστη την δυνατότητα να επεξεργαστεί τις προτιμήσεις του, δηλαδή να προσθέσει ή να καταργήσει ένα είδος ταινιών.
- **Movies:** Εδώ ο χρήστης θα βρει όλες τις διαθέσιμες ταινίες που παίζονται αυτό τον καιρό στους κινηματογράφους και ανήκουν στις κατηγορίες που τον ενδιαφέρουν. Αφού επιλέξει μία από αυτές, οδηγείται σε μία νέα φόρμα όπου πρέπει να συμπληρώσει κάποια πεδία. Αρχικά διαλέγει την ημερομηνία, την ώρα προβολής και τον κινηματογράφο στον οποίο θέλει να παρακολουθήσει την ταινία. Στην συνέχεια επιλέγει τον αριθμό των εισιτηρίων που θέλει και διαλέγει την θέση/θέσεις που επιθυμεί. Τέλος βάζει τα στοιχεία της χρεωστικής/πιστωτικής του κάρτας ώστε να πραγματοποιηθεί η online συναλλαγή.
- **My bookings:** Διαλέγοντας αυτή την λειτουργία ο χρήστης μπορεί να δει όλες τις κρατήσεις για ταινίες που έχει ήδη πραγματοποιήσει.

- Log out: Χρησιμοποιείται για έξοδο από τον λογαριασμό του χρήστη και επιστροφή στην αρχική οθόνη.



Εικόνα 6 Use Case Diagram



Εικόνα 7 Class Diagram

Παραπάνω παρατηρούμε 2 διαγράμματα:

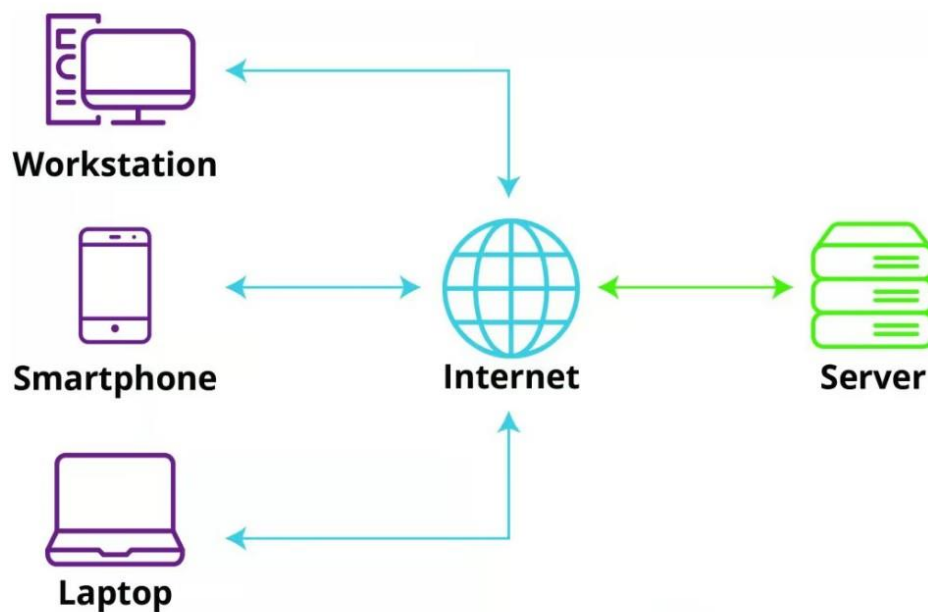
- Use Case Diagram: Εδώ παρουσιάζονται οι διάφορες λειτουργίες, πως συνδέονται μεταξύ τους καθώς και πως αλληλεπιδρά ο χρήστης με αυτές(παρουσιάζονται αναλυτικά σε επόμενη ενότητα).
- Class Diagram: Εδώ παρουσιάζονται οι κλάσεις του flask καθώς και οι συνδέσεις των tables της βάσης δεδομένων θα οποία θα αναλυθούν εκτενέστερα στην επόμενη ενότητα.

Σχεδιασμός

Στην παρούσα ενότητα θα υπάρξει ανάλυση ολόκληρου του σχεδιασμού της εφαρμογής. Αρχικά θα υπάρξει αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν, στην γλώσσα προγραμματισμού που γράφτηκε ο κώδικας, το IDE, και στα πακέτα που έχουν εφαρμοστεί στην ανάπτυξη του Flix. Έπειτα, θα αναλυθεί η αρχιτεκτονική του συστήματος, δηλαδή θα υπάρξει αναφορά και ανάλυση στις συναρτήσεις(functions), τις κλάσεις και τα packages που έχουν χρησιμοποιηθεί, αλλά και ο τρόπος με τον οποίο επικοινωνούν μεταξύ τους. Τέλος, θα έχει σειρά η ανάλυση της βάσης δεδομένων, μία οπτική αναπαράσταση με διάγραμμα, και ο σχολιασμός για τις περιπτώσεις που χρειάζεται η βάση δεδομένων να επικοινωνεί με την εφαρμογή.

Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική της εφαρμογής, η οποία έχει αναπτυχθεί με το microframework Flask, επιβεβαιώνει το μοντέλο πελάτη – διακομιστή. Η εφαρμογή μπορεί κάλλιστα να τρέξει σε έναν Server, όπου και βρίσκεται αυτός αρκεί να οριστεί και ένα domain name. Το front-end μέρος της εφαρμογής μέρος της εφαρμογής αξιοποιεί τις δυνατότητες του Flask για την δημιουργία δυναμικών και διαδραστικών σελίδων, με την χρήση HTML, CSS και Javascript. Στο back-end, για την διαχείριση των δεδομένων χρησιμοποιείται η SQLite. Το flask καθοδηγείται με την μέθοδο το routing. Με λίγα λόγια ορίζεται κάθε φορά τι θα τρέχει στο αντίστοιχο URL. Για παράδειγμα όταν ξεκινάει η εφαρμογή η αρχική σελίδα είναι στο “@app.route('/')”. Με αυτό τον τρόπο γίνεται και το routing για τις υπόλοιπες σελίδες ώστε να γνωρίζει το πρόγραμμα τι πρέπει να εμφανίσει κάθε φορά.



Εικόνα 8 Μοντέλο Πελάτη-Διακομιστή

Τεχνολογίες που Χρησιμοποιήθηκαν

Αυτή η υποενότητα αναφέρει τις τεχνολογίες που επέλεξα να χρησιμοποιήσω για την υλοποίηση της εφαρμογής. Αρχικά πρέπει να γίνει λόγος στις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν. Όπως ανέφερα και σε άλλη ενότητα, αρχικά ήμουν σε δίλημμα για το ποια γλώσσα προγραμματισμού ήταν ιδανική για αυτό το project. Τελικά αποφάσισα πως η χρήση της Python θα ήταν για μένα η πιο σωστή επιλογή καθώς προσφέρει μεγάλη ευελιξία και προσφέρει αμέτρητες δυνατότητες. Προφανώς δεν αγνοώ τα οφέλη που προσφέρουν και άλλες γλώσσες προγραμματισμού όπως είναι η C#, με την οποία ήμουν αρκετά εξοικειωμένος καθώς την είχα χρησιμοποιήσει σε αρκετά projects στο παρελθόν. Η python συνδυάστηκε με το microframework που ονομάζεται flask για να μπορέσω να στήσω την εφαρμογή σε web περιβάλλον. Συνεπώς χρησιμοποιήθηκαν και οι τυπικές γλώσσες σήμανσης για ιστοσελίδες HTML και CSS.

Όσον αφορά το περιβάλλον ανάπτυξης IDE, είναι το PyCharm, το επίσημο IDE της JetBrains για ανάπτυξη εφαρμογών Python. Ο interpreter που χρησιμοποιήθηκε ήταν της έκδοσης 3.9 και τα πακέτα ήταν τα εξής: flash, flask-sqlalchemy, flask-login, flask-wtf και wtforms. Τις λειτουργίες του καθενός τις έχω αναφέρει σε προηγούμενη ενότητα αλλά θα αναλυθούν και πιο σχολαστικά παρακάτω.

Ως βάση δεδομένων χρησιμοποίησα την SQLite Database, μία ιδιαίτερα εύχρηστη και απλή βάση, που μπορεί να δημιουργηθεί απευθείας από την ίδια την εφαρμογή, είναι ουσιαστικά μέρος του app.

Παρακάτω θα αναλύσουμε τα πακέτα που χρησιμοποιήθηκαν(πρέπει να γίνει η εγκατάσταση όλων όπως αναφέραμε παραπάνω) καθώς και την χρήση του καθενός:

- Flask: Αναλύσαμε την χρησιμότητα του παραπάνω.
- Flask-sqlalchemy: Είναι μια επέκταση του Flask που παρέχει υποστήριξη σε μία εφαρμογή. Ρυθμίζει κοινά αντικείμενα και μοτίβα όπως λειτουργίες που συνδέονται με αιτήματα Ιστού, μοντέλα και browsers.
- Flask-login: Διαχειρίζεται τις λειτουργίες login, logout και απομνημονεύει τις περιόδους σύνδεσης(sessions) των χρηστών για εκτεταμένες χρονικές περιόδους.
- Flask-wtf: Είναι συνδυασμός Flask και WTForms. Είναι υπεύθυνο για το CSRF(Cross Site Request Forgery) για την μεταφόρτωση αρχείων και για το reCAPTCHA.
- Wtforms: Είναι μια ευέλικτη βιβλιοθήκη επικύρωσης και εφαρμογής φορμών για web development σε Python. Μπορεί να λειτουργήσει με οποιοδήποτε πλαίσιο web και template επιλεχτεί. Υποστηρίζει επικύρωση δεδομένων, προστασία CSRF, διεθνοποίηση (I18N) και πολλά άλλα. Υπάρχουν διάφορες βιβλιοθήκες που παρέχουν στενότερη σύνδεση με δημοφιλή frameworks.

Όλα τα παραπάνω συνδυάστηκαν αρμονικά για να προκύψει η τελική web εφαρμογή, το Flix.

Παρουσίαση Εργασιών Εφαρμογής

Σε αυτή την υποενότητα, θα αναφερθώ στις κλάσεις και τα functions της εφαρμογής. Επίσης θα αναλύσω το πώς και για ποιο λόγο επικοινωνούν τα παραπάνω μεταξύ τους. Δεν θα σταθώ μόνο εκεί όμως, αφού θα γίνει αναφορά και στην αρχιτεκτονική του UI. Πριν ξεκινήσω, επισημαίνω ότι η περιγραφή θα είναι γραμμική, δηλαδή θα εξηγήω τα πράγματα όπως είναι η ροή της εφαρμογής και θα εξηγήω τόσο τις λειτουργίες όσο και το UI.

Ο χρήστης ανοίγοντας την εφαρμογή θα συναντήσει την αρχική σελίδα η οποία είναι η index.html . Εκεί θα παρατηρήσει διάφορες λειτουργίες. Αρχικά υπάρχει ένα κουμπί «Get Started», όπου το πάτημά του, οδηγεί τον χρήστη στην φόρμα δημιουργίας λογαριασμού(register.html) . Στην συνέχεια υπάρχουν κάποια γραφικά και κάποια σχόλια χρηστών τα οποία έχουν ενσωματωθεί με HTML στην index.html , καθώς και ένα κουμπί «See More» που δείχνει όλες τις διαθέσιμες ταινίες. Στην ουσία όταν πατιέται το κουμπί «See More» γίνεται η σύνδεση της index.html με την movies.html η οποία έχει ένα βρόγχο(for) και προβάλλει δυναμικά τις ταινίες και τις αφίσες που χρειάζεται κάθε φορά, χρησιμοποιώντας τα id των ταινιών. Στο κάτω μέρος της αρχικής οθόνης ο χρήστης μπορεί να εισάγει το e-mail του ώστε να εγγραφεί στο newsletter. Οι 2 βασικές λειτουργίες όμως της αρχικής οθόνης είναι οι επιλογές register και login. Εκεί χρησιμοποιούνται τα 2 functions, η login() και η register(). Η register() στην ουσία δημιουργεί στον χρήστη έναν λογαριασμό.

```
1 <!DOCTYPE html>
2
3 <head>
4   <meta charset="UTF-8">
5   <title>Welcome</title>
6   <link rel="stylesheet" href="/static/style.css">
7   <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
8 </head>
9 <body>
10
11   <header>
12     <nav class="left-nav">
13       <a href="{{ url_for('index') }}">
14         
15       </a>
16       {% if logged_in %}
17       <a href="{{ url_for('movies') }}">Movies</a>
18       <a href="{{ url_for('my_bookings') }}">My Bookings</a>
19       {% else %}
20
21       {% endif %}
22     </nav>
23     <nav class="right-nav">
24       {% if logged_in %}
25       <span class="welcome-message">Welcome {{ current_user.username }}</span>
26       <a href="{{ url_for('settings') }}">Settings</a>
27       <a href="{{ url_for('logout') }}">Logout</a>
28       {% else %}
29       <a href="{{ url_for('login') }}">Login</a>
30       <a href="{{ url_for('register') }}">Register</a>
31       {% endif %}
32     </nav>
33   </header>
34
35   <main>
36
37   <section class="hero">
38     <div class="hero-text">
39       <h1>Discover Your Next Favorite Movie</h1>
40       <p>Find personalized movie recommendations, stay updated on the latest releases, and never miss a great film again. Sign up now and start exploring!</p>
41       <a href="{{ url_for('register') }}" class="btn-primary">Get Started</a>
42     </div>
43   </section>
44 </body>
```

Εικόνα 9 Index.html(1)


```

44 <section class="features">
45 <h2>Why Choose Us?</h2>
46 <div class="feature-item">
47 <span class="emoji">🏠</span>
48 <h3>Personalized Recommendations</h3>
49 <p>Log in, select your preferences, and we'll suggest movies tailored just for you.</p>
50 </div>
51 <div class="feature-item">
52 <span class="emoji">🎫</span>
53 <h3>Don't Waste Time, Buy your Tickets Online</h3>
54 <p>Buy as many tickets as you want, for you and your friends.</p>
55 </div>
56 <div class="feature-item">
57 <span class="emoji">🌟</span>
58 <h3>Stay Updated on New Releases</h3>
59 <p>Be the first to know about the latest movies hitting the screens and streaming platforms.</p>
60 </div>
61 </section>
62
63
64 <section class="testimonials">
65 <h2>What Our Users Say</h2>
66 <div class="testimonial">
67 <p>"This platform helped me find the perfect movie for my weekend!" - Markos Nikolaou</p>
68 </div>
69 <div class="testimonial">
70 <p>"I discovered incredible films I wouldn't have found otherwise!" - Ariadni Charpantidou</p>
71 </div>
72 <div class="testimonial">
73 <p>"The personalized suggestions are spot on!" - Ioannis Papapetrou</p>
74 </div>
75 </section>
76
77
78 <section class="popular-movies">
79 <h2>Popular Right Now</h2>

```

Εικόνα 10 Index.html(2)

```

83 <h3>The Conjuring</h3>
84 </div>
85 <div class="movie">
86 
87 <h3>Inception</h3>
88 </div>
89 <div class="movie">
90 
91 <h3>La La Land</h3>
92 </div>
93 </div>
94 <a href="{{ url_for('movies') }}" class="btn-secondary">See More</a>
95 </section>
96
97 <section class="newsletter">
98 <div class="newsletter-text">
99 <h2>Stay Updated</h2>
100 <p>Subscribe to our newsletter to get updates on new releases, exclusive offers, and more.</p>
101 </div>
102 <form method="POST" class="newsletter-form" id="newsletterForm">
103 <input type="email" name="email" placeholder="Enter your email" required>
104 <button type="submit" class="btn-primary">Subscribe</button>
105 </form>
106 </section>
107
108 <script>
109 document.getElementById('newsletterForm').addEventListener('submit', function(event) {
110 event.preventDefault();
111 alert('Thank you for subscribing to our newsletter!');
112 });
113 </script>
114 </main>
115
116
117 <Footer>

```

Εικόνα 11 Index.html(3)

Η φόρμα εγγραφής είναι η register.html. Αφού γίνει η απαραίτητη διασφάλιση ότι τα δεδομένα που εισάγει ο χρήστης είναι ορθά και τηρούν τις προδιαγραφές, γίνονται «write» στην βάση δεδομένων user.db και πιο συγκεκριμένα στο table «user»(username,password) και στο table «category»(προτιμήσεις ταινιών).

```

34 <main>
35 <h1>Create your Flix account</h1>
36
37 {% with messages = get_flashed_messages(with_categories=true) %}
38 {% if messages %}
39 <div class="flash-messages">
40   {% for category, message in messages %}
41     <div class="alert alert-{{ category }}">
42       {{ message }}
43     </div>
44   {% endfor %}
45 </div>
46 {% endif %}
47 {% endwith %}
48
49 <form method="POST" action="{{ url_for('register') }}">
50   {{ form.hidden_tag() }}
51   <div>
52     {{ form.username.label }}
53     <small class="input-disclaimer">Minimum 4 characters, maximum 10 characters</small>
54     {{ form.username() }}
55   </div>
56   <div>
57     {{ form.password.label }}
58     <small class="input-disclaimer">Minimum 8 characters, maximum 16 characters</small>
59     {{ form.password() }}
60   </div>
61   <div>
62     <label for="movie_categories">I am interested in...</label>
63     <select id="movie_categories" name="movie_categories" class="form-control" multiple="multiple">
64       <option value="horror">👹 Horror</option>
65       <option value="comedy">😄 Comedy</option>
66       <option value="romance">💕 Romance</option>
67       <option value="action">🔪 Action</option>
68       <option value="drama">🎬 Drama</option>
69       <option value="sci-fi">🚀 Sci-Fi</option>
70       <option value="fantasy">🧙 Fantasy</option>
71       <option value="documentary">📽 Documentary</option>

```

Εικόνα 12 Register.html

```

@app.route( rule: '/register', methods=['GET', 'POST'])
def register():
    if current_user.is_authenticated:
        return redirect(url_for('index'))
    form = RegisterForm()
    if form.validate_on_submit():

        existing_user = User.query.filter_by(username=form.username.data).first()
        if existing_user:
            flash( message: 'Username already taken. Please choose a different one.', category: 'error')
            return render_template( template_name_or_list: 'register.html', form=form)

        new_user = User(username=form.username.data, password=form.password.data)
        selected_categories = form.movie_categories.data
        categories = Category.query.filter(Category.name.in_(selected_categories)).all()
        new_user.categories = categories
        db.session.add(new_user)
        db.session.commit()
        flash('You have created a new account, you can now login!')
        return redirect(url_for('login'))
    return render_template( template_name_or_list: 'register.html', form=form)

```

Εικόνα 13 Register Function

Όταν τελειώσει αυτή η διαδικασία, ο χρήστης μεταβιβάζεται στην φόρμα login(login.html) όπου και εισάγει το username και το password. Γίνεται η διασταύρωση στοιχείων στο table user και αν αυτά είναι σωστά και συμπίπτουν, το πρόγραμμα καλωσορίζει τον χρήστη και τον μεταφέρει στην αρχική οθόνη στην οποία πλέον έχουν προστεθεί περισσότερες επιλογές.

```

6 <link rel="stylesheet" href="/static/style.css">
7 <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
8 </head>
9 <body>
10 <header>
11 <nav class="left-nav">
12 <a href="{{ url_for('index') }}">
13 
14 </a>
15 {% if logged_in %}
16 <a href="{{ url_for('movies') }}">Movies</a>
17 <a href="{{ url_for('my_bookings') }}">My Bookings</a>
18 {% else %}
19
20 {% endif %}
21 </nav>
22 <nav class="right-nav">
23 {% if logged_in %}
24 <span class="welcome-message">Welcome {{ current_user.username }}</span>
25 <a href="{{ url_for('settings') }}">Settings</a>
26 <a href="{{ url_for('logout') }}">Logout</a>
27 {% else %}
28 <a href="{{ url_for('login') }}">Login</a>
29 <a href="{{ url_for('register') }}">Register</a>
30 {% endif %}
31 </nav>
32 </header>
33 <main>
34 <h1>Login to your Flix account</h1>
35 {% for message in get_flashed_messages() %}
36 <div>{{ message }}</div>
37 {% endfor %}
38 <form method="POST" action="{{ url_for('login') }}">
39 {{ form.hidden_tag() }}
40 <div>{{ form.username.label }} {{ form.username() }}</div>
41 <div>{{ form.password.label }} {{ form.password() }}</div>
42 <div>{{ form.submit() }}</div>
43 </form>
44 </main>

```

Εικόνα 14 Login.html

```

142 @app.route(rule='/login', methods=['GET', 'POST'])
143 def login():
144     if current_user.is_authenticated:
145         return redirect(url_for('index'))
146     form = LoginForm()
147     if form.validate_on_submit():
148         user = User.query.filter_by(username=form.username.data).first()
149         if user and user.password == form.password.data:
150             login_user(user, remember=True) # Κρατάει Logged In τον χρήστη
151             return redirect(url_for('index'))
152         else:
153             flash('Invalid username or password')
154     return render_template(template_name_or_list='login.html', form=form)

```

Εικόνα 15 Login Function

Αυτό γίνεται με τον έλεγχο της συνθήκης "if logged_in" η οποία βρίσκεται στην index.html . Αν αυτή η συνθήκη αυτή είναι true, τότε το πρόγραμμα καταλαβαίνει ότι ο χρήστης έχει κάνει login επιτυχώς στην φόρμα και εμφανίζονται στο πάνω μέρος της οθόνης κάποιες επιπλέον επιλογές: το «Movies», το «My bookings», το «Settings» και το «Logout».

Αφού κάνει login ο χρήστης έχει την δυνατότητα να επιλέξει την λειτουργία «settings» και να επεξεργαστεί τις προτιμήσεις ταινιών που έχει δηλώσει κατά την δημιουργία του λογαριασμού του. Δηλαδή μπορεί να προσθέσει κατηγορίες αλλά και να διαγράψει. Για αυτή την λειτουργία χρησιμοποιείται το function «settings» στο app.py . Στην ουσία αυτή η συνάρτηση αλλάζει το table «category» ανάλογα με την επεξεργασία που κάνει ο χρήστης στις προτιμήσεις του. Η σύνδεση γίνεται με το table «user categories» και με την χρήση του primary key «id» διασταυρώνονται για να γίνουν οι αλλαγές στον σωστό χρήστη.

```

184 @app.route(rule='/settings', methods=['GET', 'POST'])
185 @login_required
186 def settings():
187     form = SettingsForm()
188     if request.method == 'GET':
189         form.movie_categories.data = [category.name for category in current_user.categories]
190     if form.validate_on_submit():
191         selected_categories = form.movie_categories.data
192         categories = Category.query.filter(Category.name.in_(selected_categories)).all()
193         current_user.categories = categories
194         db.session.commit()
195         flash('Your settings have been updated!')
196         logging.debug(f'User accessing settings: username={current_user.username}, existing_categories={selected_categories}')
197         return redirect(url_for('index'))
198     selected_categories = [category.name for category in current_user.categories]
199     return render_template(template_name_or_list='settings.html', form=form, selected_categories=selected_categories)
200

```

Εικόνα 16 Settings Function

Μία από τις βασικές λειτουργίες της εφαρμογής είναι η επιλογή «Movies». Με το πάτημα αυτού του κουμπιού στην ουσία εμφανίζονται στον χρήστη όλες οι διαθέσιμες ταινίες που ανήκουν στις αγαπημένες κατηγορίες που έχει δηλώσει ο ίδιος. Για αυτή την ενέργεια συνεργάζονται πολλές λειτουργίες ταυτόχρονα και σχεδόν όλα τα tables επικοινωνούν μεταξύ τους. Αρχικά χρησιμοποιείται το function «movies» στο app.py που αναζητά στο table user τον συγκεκριμένο χρήστη, το τελευταίο table με την σειρά του επικοινωνεί με τα table user_categories και category και έτσι βρίσκονται οι προτιμήσεις του εκάστοτε χρήστη. Μέσω της movies.html αυτές οι ταινίες των κατηγοριών που αναφέραμε παραπάνω προβάλλονται στην οθόνη δυναμικά. Ο χρήστης έχει την δυνατότητα να πατήσει μία από αυτές και θα μεταφερθεί στην επόμενη οθόνη όπου πρέπει να ακολουθήσει κάποια βήματα για να τελειώσει την αγορά. Οι επιλογές που έχει είναι η ημερομηνία που επιθυμεί, ο κινηματογράφος, η ώρα, ο αριθμός εισιτηρίων αλλά και οι θέσεις. Στην συνέχεια βάζει τα στοιχεία της κάρτας του ώστε να ολοκληρωθεί η ηλεκτρονική συναλλαγή. Όλα τα παραπάνω βρίσκονται στην movie_detail.html . Σε συνεργασία με το function movie_detail γίνονται όλες οι απαραίτητες διαδικασίες ώστε να διαπεραιωθούν τα παραπάνω. Αρχικά όλες οι πληροφορίες της κράτησης αποθηκεύονται στο table booking. Όσο εισάγει ο χρήστης τα στοιχεία της κάρτας γίνεται και η απαραίτητη διασφάλιση ότι τα δεδομένα που εισάγονται πληρούν τις προϋποθέσεις.

```

10 <nav class="left-nav">
11   <a href="{{ url_for('index') }}">
12     
13   </a>
14   {% if logged_in %}
15     <a href="{{ url_for('movies') }}">Movies</a>
16     <a href="{{ url_for('my_bookings') }}">My Bookings</a>
17   {% endif %}
18 </nav>
19 <nav class="right-nav">
20   {% if logged_in %}
21     <span class="welcome-message">Welcome {{ current_user.username }}</span>
22     <a href="{{ url_for('settings') }}">Settings</a>
23     <a href="{{ url_for('logout') }}">Logout</a>
24   {% else %}
25     <a href="{{ url_for('login') }}">Login</a>
26     <a href="{{ url_for('register') }}">Register</a>
27   {% endif %}
28 </nav>
29 </header>
30 <main>
31   <h1>Our movie recommendations for you</h1>
32   <section class="movies-list">
33     {% for movie in movies %}
34       <div class="movie">
35         {% if logged_in %}
36           <a href="{{ url_for('movie_detail', movie_id=movie.id) }}">
37             
38             <h3>{{ movie.title }}</h3>
39             <p>Book a ticket now!</p>
40           </a>
41         {% else %}
42           <div class="movie-not-logged-in">
43             
44             <h3>{{ movie.title }}</h3>
45             <p>Please <a href="{{ url_for('register') }}">register</a> or <a href="{{ url_for('login') }}">login</a> to book tickets.</p>
46           </div>
47         {% endif %}

```

Εικόνα 17 Movies.html(1)

```

35 <h3>Booking Details</h3>
36 <form method="POST" action="{{ url_for('movie_detail', movie_id=movie.id) }}">
37   {{ form.hidden_tag() }}
38   <div>
39     {{ form.date.label }} {{ form.date(id="dateField") }}
40   </div>
41   <div>
42     {{ form.hour.label }} {{ form.hour(id="hourField") }}
43   </div>
44   <div>
45     {{ form.cinema.label }} {{ form.cinema(id="cinemaField") }}
46   </div>
47   <div>
48     {{ form.tickets.label }}
49     <div class="ticket-controls">
50       <button type="button" id="decrease">-</button>
51       {{ form.tickets(class="ticket-input") }}
52       <button type="button" id="increase">+</button>
53     </div>
54   </div>
55   <div class="seatmap-container">
56     <br>
57     <h3>Select Seats</h3>
58     <div id="seatMap" class="seatmap">
59     </div>
60   </div>
61   <input type="hidden" name="selectedSeats" id="selectedSeats">
62   <br>
63   <h3>Payment Details</h3>
64   <div>
65     {{ form.card_number.label }} {{ form.card_number() }}
66   </div>
67   <div>
68     {{ form.card_date.label }} {{ form.card_date() }}
69   </div>
70   <div>
71     {{ form.card_cvv.label }} {{ form.card_cvv() }}

```

Εικόνα 18 Movies.html(2)

```

47 <script>
48 document.addEventListener('DOMContentLoaded', function() {
49   const ticketInput = document.querySelector('.ticket-input');
50   const increaseBtn = document.getElementById('increase');
51   const decreaseBtn = document.getElementById('decrease');
52   const totalCostEl = document.getElementById('totalCost');
53   const seatMapContainer = document.getElementById('seatMap');
54   const dateField = document.getElementById('dateField');
55   const hourField = document.getElementById('hourField');
56   const cinemaField = document.getElementById('cinemaField');
57   const selectedSeatsInput = document.getElementById('selectedSeats');
58   const ticketPrice = 10;
59   let selectedSeats = [];
60   const rows = ['A', 'B', 'C', 'D', 'E'];
61   const seatsPerRow = 10;
62   const maxSeats = rows.length * seatsPerRow;
63
64   const today = new Date().toISOString().split('T')[0];
65   dateField.setAttribute('min', today);
66
67   function updateTotalCost() {
68     const numberOfTickets = parseInt(ticketInput.value) || 0;
69     totalCostEl.textContent = numberOfTickets * ticketPrice;
70   }
71
72   increaseBtn.addEventListener('click', function() {
73     ticketInput.value = parseInt(ticketInput.value) || 0;
74     ticketInput.value = Math.min(maxSeats, Math.max(0, parseInt(ticketInput.value) + 1));
75     updateTotalCost();
76     updateSeatSelection();
77   });
78
79   decreaseBtn.addEventListener('click', function() {
80     ticketInput.value = parseInt(ticketInput.value) || 0;
81     ticketInput.value = Math.max(0, parseInt(ticketInput.value) - 1);
82     updateTotalCost();
83     updateSeatSelection();
84   });
85
86   ticketInput.addEventListener('input', function() {
87     updateTotalCost();
88     updateSeatSelection();
89   });
90
91   function createSeatMap() {
92     seatMapContainer.innerHTML = '';
93     rows.forEach(row => {
94       const rowLabel = document.createElement('div');
95       rowLabel.classList.add('seat-row-label');
96       rowLabel.textContent = row;
97       seatMapContainer.appendChild(rowLabel);
98
99       for (let i = 1; i <= seatsPerRow; i++) {
100        const seat = document.createElement('div');
101        seat.classList.add('seat');
102        seat.textContent = Math.random() > 0.8 ? 'X' : ' ';
103        if (seat.textContent === 'X') {
104          seat.classList.add('occupied');
105        } else {
106          seat.addEventListener('click', () => toggleSeatSelection(seat, row + i));
107        }
108        seatMapContainer.appendChild(seat);
109      }
110    });
111  }
112
113  function toggleSeatSelection(seatElement, seatNumber) {
114    const numberOfTickets = parseInt(ticketInput.value) || 0;
115    if (seatElement.classList.contains('selected')) {
116      seatElement.classList.remove('selected');
117      selectedSeats = selectedSeats.filter(seat => seat !== seatNumber);
118    } else if (selectedSeats.length < numberOfTickets) {
119      seatElement.classList.add('selected');
120      selectedSeats.push(seatNumber);
121    }
122    selectedSeatsInput.value = selectedSeats.join(',');

```

Εικόνα 19 Movies.html(3)

```

125   ticketInput.addEventListener('input', function() {
126     updateTotalCost();
127     updateSeatSelection();
128   });
129
130   function createSeatMap() {
131     seatMapContainer.innerHTML = '';
132     rows.forEach(row => {
133       const rowLabel = document.createElement('div');
134       rowLabel.classList.add('seat-row-label');
135       rowLabel.textContent = row;
136       seatMapContainer.appendChild(rowLabel);
137
138       for (let i = 1; i <= seatsPerRow; i++) {
139        const seat = document.createElement('div');
140        seat.classList.add('seat');
141        seat.textContent = Math.random() > 0.8 ? 'X' : ' ';
142        if (seat.textContent === 'X') {
143          seat.classList.add('occupied');
144        } else {
145          seat.addEventListener('click', () => toggleSeatSelection(seat, row + i));
146        }
147        seatMapContainer.appendChild(seat);
148      }
149    });
150  }
151
152   function toggleSeatSelection(seatElement, seatNumber) {
153     const numberOfTickets = parseInt(ticketInput.value) || 0;
154     if (seatElement.classList.contains('selected')) {
155       seatElement.classList.remove('selected');
156       selectedSeats = selectedSeats.filter(seat => seat !== seatNumber);
157     } else if (selectedSeats.length < numberOfTickets) {
158       seatElement.classList.add('selected');
159       selectedSeats.push(seatNumber);
160     }
161     selectedSeatsInput.value = selectedSeats.join(',');

```

Εικόνα 20 Movies.html(4)

```

203 @app.route('/movies')
204 def movies():
205     if current_user.is_authenticated:
206         preferred_categories = [category.name for category in current_user.categories]
207         movies = Movie.query.filter(Movie.category_name.in_(preferred_categories)).all()
208     else:
209         movies = Movie.query.all()
210     return render_template(template_name_or_list='movies.html', movies=movies)
211
212 @app.route(rule='/movie/<int:movie_id>', methods=['GET', 'POST'])
213 def movie_detail(movie_id):
214     movie = Movie.query.get_or_404(movie_id)
215     form = BookingForm()
216     total_cost = None
217     if form.validate_on_submit():
218         tickets = int(form.tickets.data)
219         total_cost = tickets * 10
220         new_booking = Booking(
221             user_id=current_user.id,
222             movie_id=movie.id,
223             date=form.date.data.strftime('%Y-%m-%d'),
224             hour=form.hour.data,
225             cinema=form.cinema.data,
226             tickets=tickets,
227             seats=request.form['selectedSeats'],
228             card_number=form.card_number.data,
229             card_date=form.card_date.data,
230             card_cvv=form.card_cvv.data,
231             card_name=form.card_name.data,
232             total_cost=total_cost
233         )
234         db.session.add(new_booking)
235         db.session.commit()
236         flash('Your tickets have been booked!')
237         return redirect(url_for('my_bookings'))
238     return render_template(template_name_or_list='movie_detail.html', movie=movie, form=form, total_cost=total_cost)

```

Εικόνα 21 Movies Function

Αφού γίνει η αγορά του εισιτηρίου, ο χρήστης μεταφέρεται στο παράρτημα «My bookings» το οποίο του δείχνει τις κρατήσεις που έχει πραγματοποιήσει καθώς και ένα QR για καθεμία από αυτές ώστε να εισέλθει στον αντίστοιχο κινηματογράφο και στην αντίστοιχη αίθουσα. Εκτός από το QR φαίνονται και άλλες λεπτομέρειες για την κάθε κράτηση όπως η ταινία, ο κινηματογράφος, οι θέσεις κτλ . Όλα τα παραπάνω υπάρχουν στην my_bookings.html . Στην ουσία εδώ χρησιμοποιούνται τα table user και booking για να διασταυρωθούν και να απεικονιστούν οι κρατήσεις κάθε χρήστη. Αυτό κάνει και στην ουσία το function my_bookings, κάνει προσπέλαση του table booking.

```

8 <body>
9 <header>
10 <nav class="left-nav">
11 <a href="{{ url_for('index') }}">
12 
13 </a>
14 <a href="{{ url_for('movies') }}">Movies</a>
15 <a href="{{ url_for('my_bookings') }}">My Bookings</a>
16 </nav>
17 <nav class="right-nav">
18 {% if logged_in %}
19 <span class="welcome-message">Welcome {{ current_user.username }}</span>
20 <a href="{{ url_for('settings') }}">Settings</a>
21 <a href="{{ url_for('logout') }}">Logout</a>
22 {% else %}
23 <a href="{{ url_for('login') }}">Login</a>
24 <a href="{{ url_for('register') }}">Register</a>
25 {% endif %}
26 </nav>
27 </header>
28 <main>
29 <h1>My Bookings</h1>
30 <section class="bookings-list">
31 {% for booking in bookings %}
32 <div class="booking">
33 
34 <div>
35 <h3>{{ booking.movie.title }}</h3>
36 <p>Date: {{ booking.date }}</p>
37 <p>Hour: {{ booking.hour }}</p>
38 <p>Cinema: {{ cinema_names[booking.cinema] }}</p>
39 <p>Tickets: {{ booking.tickets }}</p>
40 <p>Seats: {{ booking.seats }}</p>
41 <p>Total Cost: {{ booking.total_cost }} EUR</p>
42 </div>
43 <div class="qr-code">
44 
45 </div>
46 </div>
47 {% endfor %}

```

Εικόνα 22 My_bookings.html

```

240 @app.route('/my_bookings')
241 @login_required
242 def my_bookings():
243     bookings = Booking.query.filter_by(user_id=current_user.id).all()
244     return render_template(template_name_or_list='my_bookings.html', bookings=bookings, cinema_names=CINEMA_NAMES)

```

Εικόνα 23 My_bookings Function

Η τελευταία και η πιο απλή λειτουργία είναι αυτή του «logout» που στην ουσία αποσυνδέει τον χρήστη από τον λογαριασμό του. Εδώ χρησιμοποιείται η συνάρτηση «logout» που εκτός από την αποσύνδεση από τον λογαριασμό του, αναλαμβάνει να μεταφέρει τον χρήστη στην αρχική οθόνη με τις βασικές λειτουργίες.

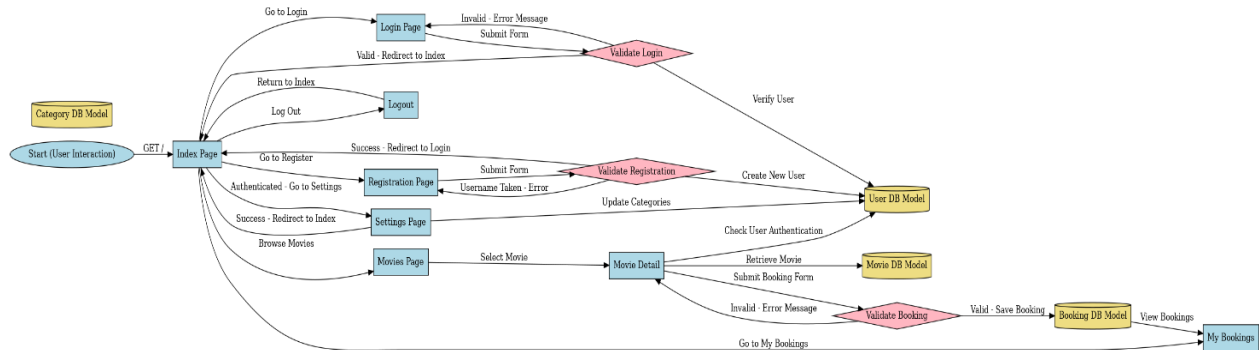
```

178 @app.route('/Logout')
179 @login_required
180 def logout():
181     logout_user()
182     return redirect(url_for('index'))

```

Εικόνα 24 Logout Function

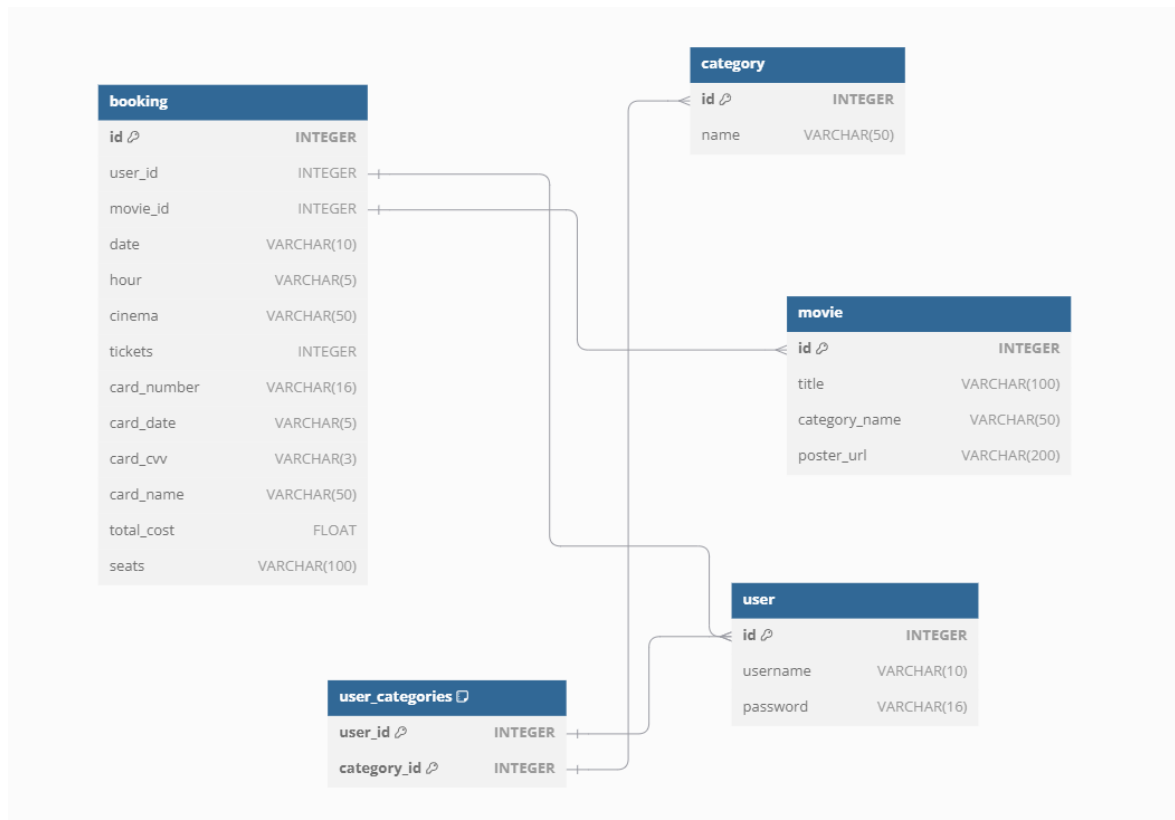
Όλες οι εργασίες που αναλύονται παραπάνω αναπαρίστανται αναλυτικά στο παρακάτω διάγραμμα ροής. Το οβάλ σχήμα δηλώνει αρχή και τέλος. Το ορθογώνιο παραλληλόγραμμα αναπαριστά μια διαδικασία. Ο ρόμβος παρουσιάζει μια συνθήκη και έχει παραπάνω από μια εξόδους. Τέλος τα σχήματα με το κίτρινο χρώμα δείχνουν τα tables της βάσης δεδομένων



Εικόνα 25 Διάγραμμα Ροής Λειτουργιών

Περιγραφή Βάσης Δεδομένων

Φτάνοντας στο τελευταίο κομμάτι της ανάλυσης του σχεδιασμού της εφαρμογής, ήρθε η ώρα να περιγράψω την δομή της βάσης δεδομένων που χρησιμοποίησα. Ο τρόπος με τον οποίο αλληλεπιδρά με το app έχει αναλυθεί παραπάνω. Παρακάτω θα παρουσιάσω αρχικά την βάση με οπτικό τρόπο και μετά θα υπάρξει περαιτέρω ανάλυση.



Εικόνα 26 Διάγραμμα Βάσης Δεδομένων

Όπως φαίνεται στο παραπάνω διάγραμμα, η βάση δεδομένων αποτελείται από 5 tables. Το booking, το category, το user, το user_categories και το movie. Εμβαθύνοντας θα εξηγήσω τα στοιχεία του κάθε table.

Table booking:

- Id: Το id(primary key) κάθε κράτησης.

- User_id: Το id του χρήστη της κράτησης που γίνεται reference στο user table και επικοινωνούν μεταξύ τους.
- Movie_id: Το id της ταινίας της κράτησης που γίνεται reference στο movie table και επικοινωνεί μαζί του για τον προσδιορισμό της ταινίας.
- Date: Η ημερομηνία της κράτησης.
- Hour: Η ώρα της κράτησης.
- Cinema: Ο επιλεγμένος κινηματογράφος για την κράτηση.
- Tickets: Ο αριθμός των εισιτηρίων που επιθυμεί ο χρήστης.
- Card_Number: Ο 16ψήφιος αριθμός της πιστωτικής κάρτας.
- Card_Date: Η ημερομηνία λήξης της κάρτας.
- Card_cvv: Ο τριψήφιος αριθμός ασφαλείας της κάρτας.
- Card_Name: Το όνομα κατόχου της κάρτας.
- Total_cost: Το συνολικό κόστος μετά την αγορά εισιτηρίου/εισιτηρίων.
- Seats: Οι επιλεγμένες θέσεις του χρήστη.

Table category:

- Id: Το id της κατηγορίας(primary key) ταινιών που γίνεται reference στο user_categories για την αντιστοίχιση των στοιχείων.
- Name: Το όνομα κατηγορίας ταινιών.

Table User:

- Id: Το id του χρήστη(primary key) που επικοινωνεί με το πεδίο user_id στο table booking.
- Username: Το username του χρήστη.
- Password: Ο κωδικός του χρήστη.

Table user categories:

- User_id: Το id του χρήστη(pk) το οποίο επικοινωνεί με το πεδίο id στο table user.
- Category_id: Το id της κατηγορίας ταινιών που επικοινωνεί με το πεδίο id στο table category.

Table movie:

- Id: Το id της ταινίας(pk) που επικοινωνεί με το πεδίο movie_id στο table booking.
- Title: Ο τίτλος της ταινίας.
- Category_name: Η κατηγορία στην οποία ανήκει η ταινία.
- Poster_url: Το path για την εικόνα της ταινίας στο πρόγραμμα.

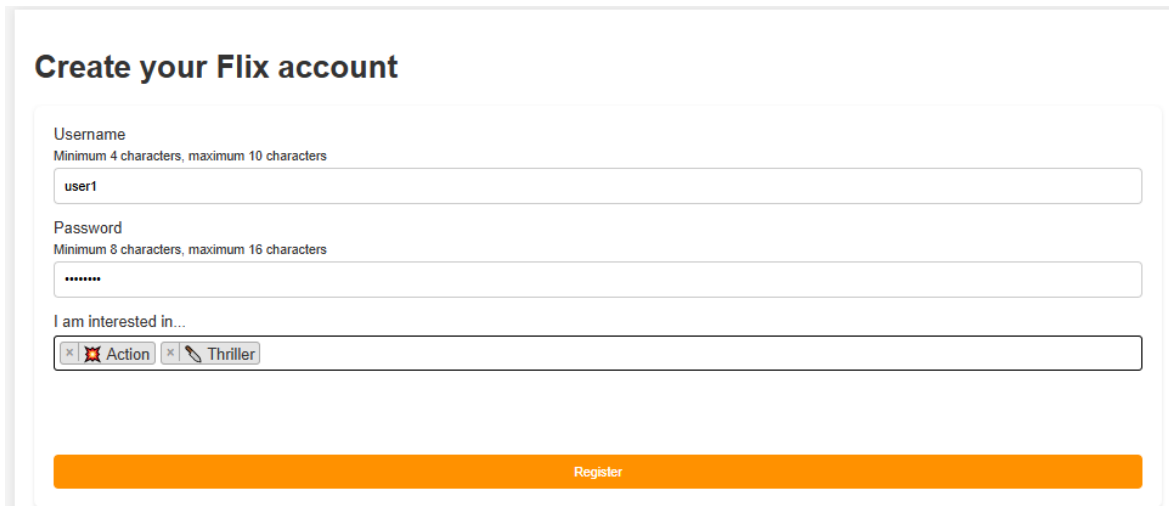
Ο διαμεσολαβητής που χρησιμοποιώ για την επικοινωνία της εφαρμογής με την βάση δεδομένων είναι η κλάση SQLiteManager. Στην κλάση αυτή υπάρχει κώδικας της δημιουργίας της βάσης αλλά και των updates που έχω εφαρμόσει, αφού λόγω της εξέλιξης της εφαρμογής κατά την ανάπτυξή της χρειάστηκε να προσαρμόσω κατάλληλα τα tables.

Η SQLiteManager παίζει πολύ μεγάλο ρόλο και κατά την διάρκεια της λειτουργίας της εφαρμογής. Αρχικά, με την εκκίνηση του Flix, χρησιμοποιούνται μέθοδοι που στεγάζει η SQLiteManager, που αφορούν την φόρτωση των δεδομένων στα κατάλληλα ArrayLists. Είναι επίσης πολλές φορές που καλούνται functions ώστε να ανανεωθούν, να προστεθούν ή και να διαγραφτούν διάφορα στοιχεία. Υπάρχουν και άλλες σημαντικές λειτουργίες όπως η εύρεση και η αντιστοίχιση των ID μεταξύ των tables.

Σενάρια χρήσης του Συστήματος με Screenshots

Σε αυτή την ενότητα θα υπάρξουν ορισμένα παραδείγματα με μορφή screenshot, που απεικονίζουν την αλληλεπίδραση μεταξύ συστήματος και χρήστη. Αναλυτική περιγραφή για την συμπεριφορά της εφαρμογής θα γίνει στο εγχειρίδιο χρήστη. Θα υποθέσουμε ότι υπάρχουν 2 χρήστες για την απεικόνιση των σεναρίων, ο **user1** και ο **user2**.

Αρχικά δημιουργούμε λογαριασμό για τον **user1** και διαλέγουμε ως ενδιαφερόμενες κατηγορίες ταινιών τις «action» και «thriller».



Create your Flix account

Username
Minimum 4 characters, maximum 10 characters
user1

Password
Minimum 8 characters, maximum 16 characters
.....

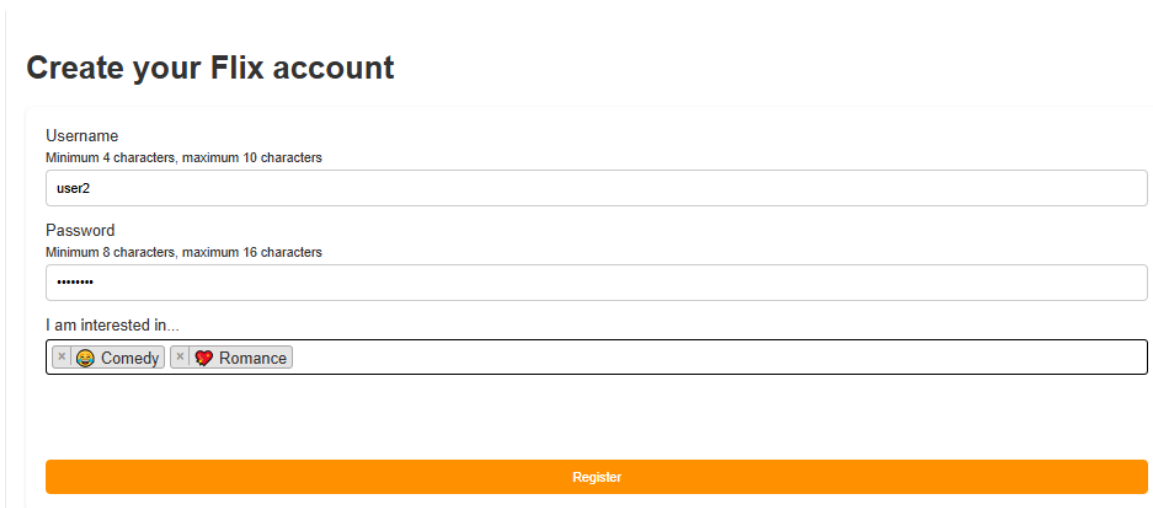
I am interested in...

Action Thriller

Register

Εικόνα 27 Φόρμα Δημιουργίας Λογαριασμού(User1)

Επαναλαμβάνουμε την ίδια διαδικασία και για τον **user2** αλλά διαλέγουμε διαφορετικές κατηγορίες(comedy, romance).



Create your Flix account

Username
Minimum 4 characters, maximum 10 characters
user2

Password
Minimum 8 characters, maximum 16 characters
.....

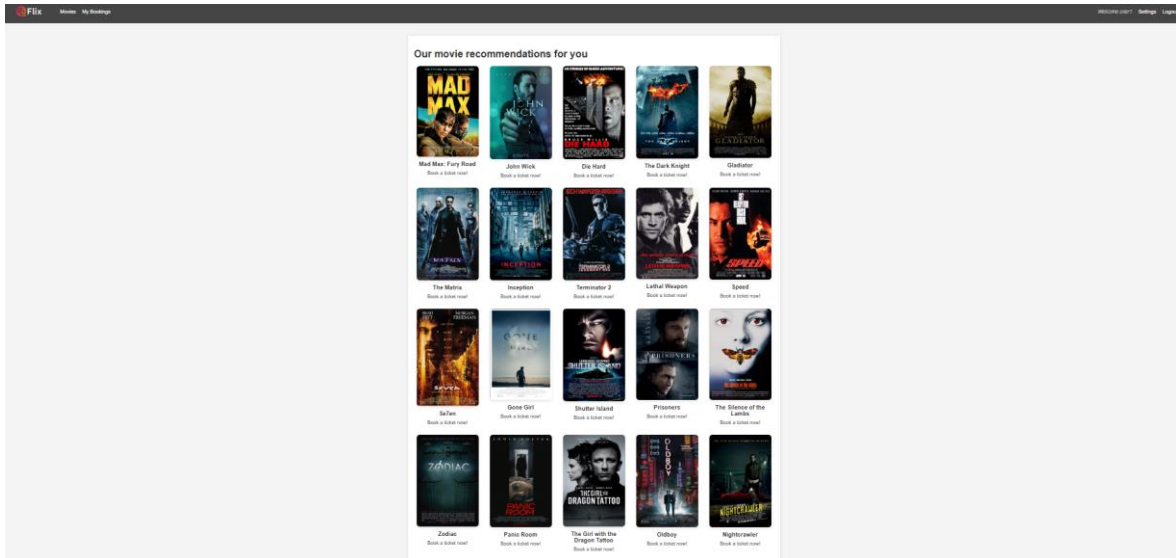
I am interested in...

Comedy Romance

Register

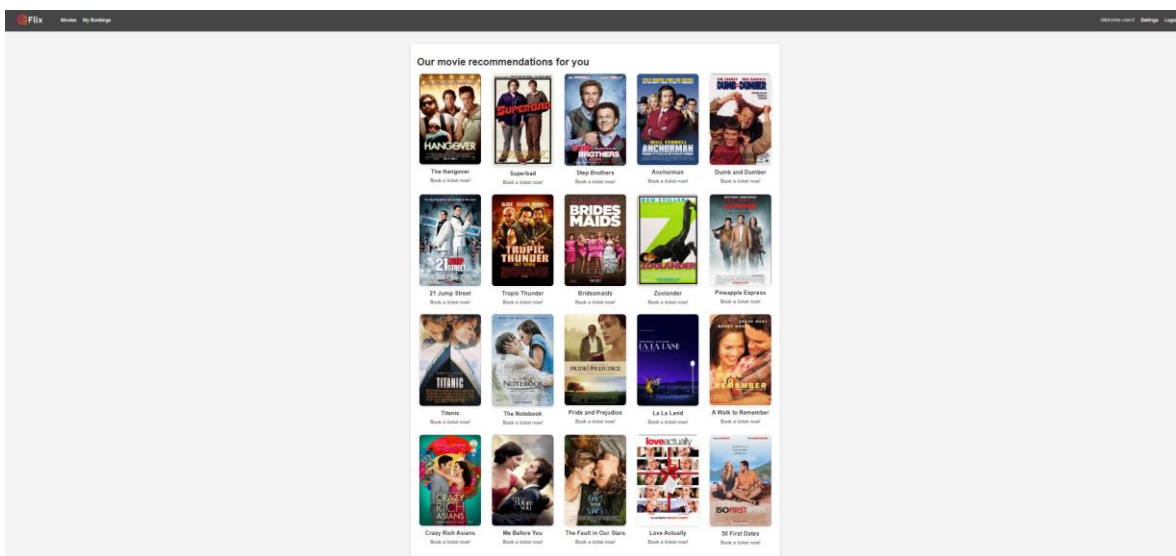
Εικόνα 28 Φόρμα Δημιουργίας Λογαριασμού(User2)

Αφού κάνει login ο **user1**, τον καλωσορίζει η εφαρμογή(“welcome user1”) και αφού πατήσει την επιλογή **movies** θα του εμφανιστούν οι διαθέσιμες ταινίες που παίζονται στους κινηματογράφους και ανήκουν στις κατηγορίες που προτιμάει(thriller, action). Εδώ φαίνεται και το χαρακτηριστικό της εξατομίκευσης.



Εικόνα 29 Λίστα Ταινιών(User1)

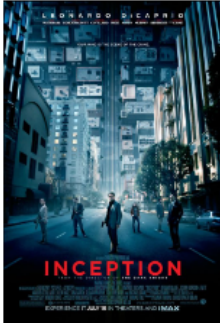
Αντίθετα στον **user2** θα του εμφανιστούν οι διαθέσιμες ταινίες που ανήκουν στις κατηγορίες comedy και romance όπως ο ίδιος έχει επιλέξει.



Εικόνα 30 Λίστα Ταινιών(User2)

Έστω ότι ο κάθε user αγοράζει εισιτήριο για μία ταινία της αρεσκείας του. Για παράδειγμα ο **user1** το «Inception» και ο **user2** το «The Hangover». Αρχικά βάζουν όλες τις απαραίτητες πληροφορίες στην φόρμα αγοράς εισιτηρίου για να ολοκληρωθεί η ηλεκτρονική αγορά.

Inception



Booking Details

Date

Hour

Cinema

Number of Tickets

Select Seats

A	1	2	3	4	5	X	X	B	9	10
B	1	2	3	4	5	6	7	B	X	10
C	X	2	3	4	5	6	7	B	9	10
D	X	2	X	X	X	6	7	B	9	10
E	1	X	3	4	5	X	7	B	X	10

Payment Details

Credit Card Number

Expiration Date (MM/YY)


CVV

Name on Card

Book Tickets

Εικόνα 31 Επιλεγμένη Ταινία(User1)

The Hangover



Booking Details

Date

11/21/2024 🗓

Hour

12:00 PM ▼

Cinema

Village Cinemas - The Mall Athens ▼

Number of Tickets

-

1

+

Select Seats

A	1	2	3	X	5	X	7	B	X	X
B	X	2	3	4	5	6	7	B	9	10
C	1	2	3	4	5	6	7	B	9	10
D	1	X	3	4	X	X	7	B	9	10
E	1	2	3	X	5	6	X	B	X	10

Payment Details

Credit Card Number

4538950985473897

Expiration Date (MM/YY)

06/20

CVV

222

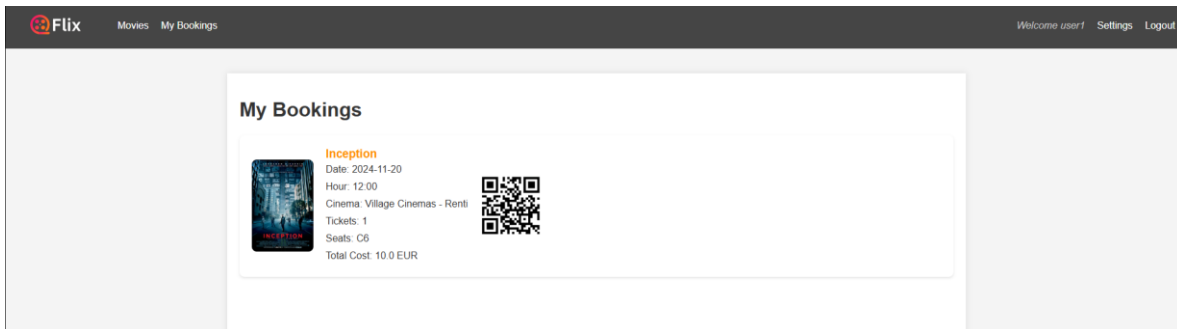
Name on Card

Konstantinos Rizos

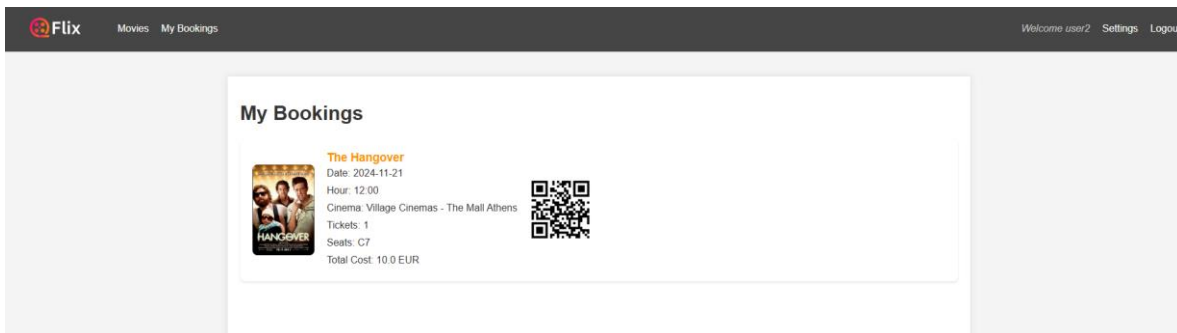
Book Tickets

Εικόνα 32 Επιλεγμένη Ταινία(User2)

Τέλος, αν οι χρήστες διαλέξουν την επιλογή «My Bookings», θα δει ο καθένας την αντίστοιχη κράτησή του.



Εικόνα 33 Κράτηση(User1)

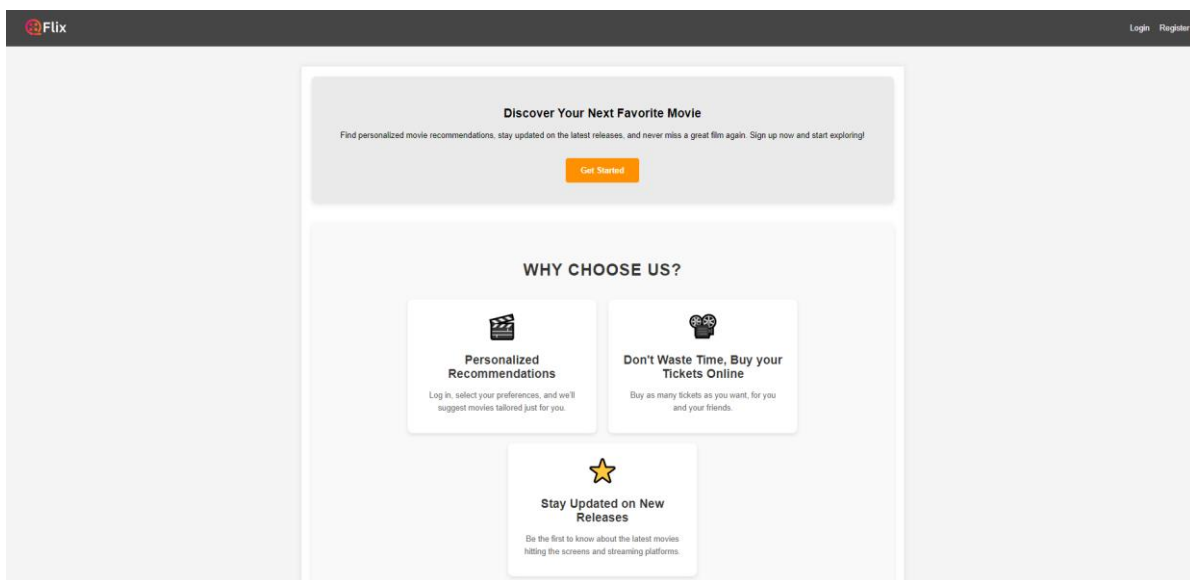


Εικόνα 34 Κράτηση(User2)

Εγχειρίδιο Χρήστη

Παρακάτω αναπτύσσεται το εγχειρίδιο χρήστη της εφαρμογής. Το εγχειρίδιο αυτό έχει δημιουργηθεί για να βοηθήσει τον χρήστη να εκμεταλλευτεί τις δυνατότητες του Flix στο έπακρο.

Αρχικά, βρίσκεστε στην αρχική οθόνη. Η πρώτες λειτουργίες που θα παρατηρήσετε είναι οι επιλογές «Login» και «Register». Παρακάτω υπάρχει ένα κουμπί που γράφει «Get Started» το οποίο αν το πατήσετε θα σας μεταφέρει στην φόρμα δημιουργίας λογαριασμού για να ξεκινήσει η εμπειρία σας στο Flix. Στην συνέχεια παρατηρούνται κάποια «γραφικά» που επισημαίνουν τους λόγους για τους οποίους κάποιος πρέπει να προτιμήσει το Flix.



Εικόνα 35 Αρχική Οθόνη(1)

Κατεβαίνοντας θα συναντήσετε κάποια σχόλια χρηστών που λένε την γνώμη τους για το Flix καθώς και κάποιες δημοφιλείς ταινίες που παίζονται τώρα. Υπάρχει ένα κουμπί «See More» που εμφανίζει όλες τις ταινίες που παίζονται αυτό τον καιρό. Στο κάτω μέρος της αρχικής σελίδας υπάρχει ένα πεδίο που μπορείτε να εισάγετε το e-mail σας και να πατήσετε το κουμπί «Subscribe» για να σας έρχονται e-mail για καινούριες ταινίες και σχετικά νέα(αφού γίνει subscribe εμφανίζεται αντίστοιχο μήνυμα).


What Our Users Say

"This platform helped me find the perfect movie for my weekend!" - Markos Nikolaou


"I discovered incredible films I wouldn't have found otherwise!" - Ariadni Charpantidou

"The personalized suggestions are spot on!" - Ioannis Papapetrou


Popular Right Now



The Conjuring



Inception



La La Land

[See More](#)

Stay Updated

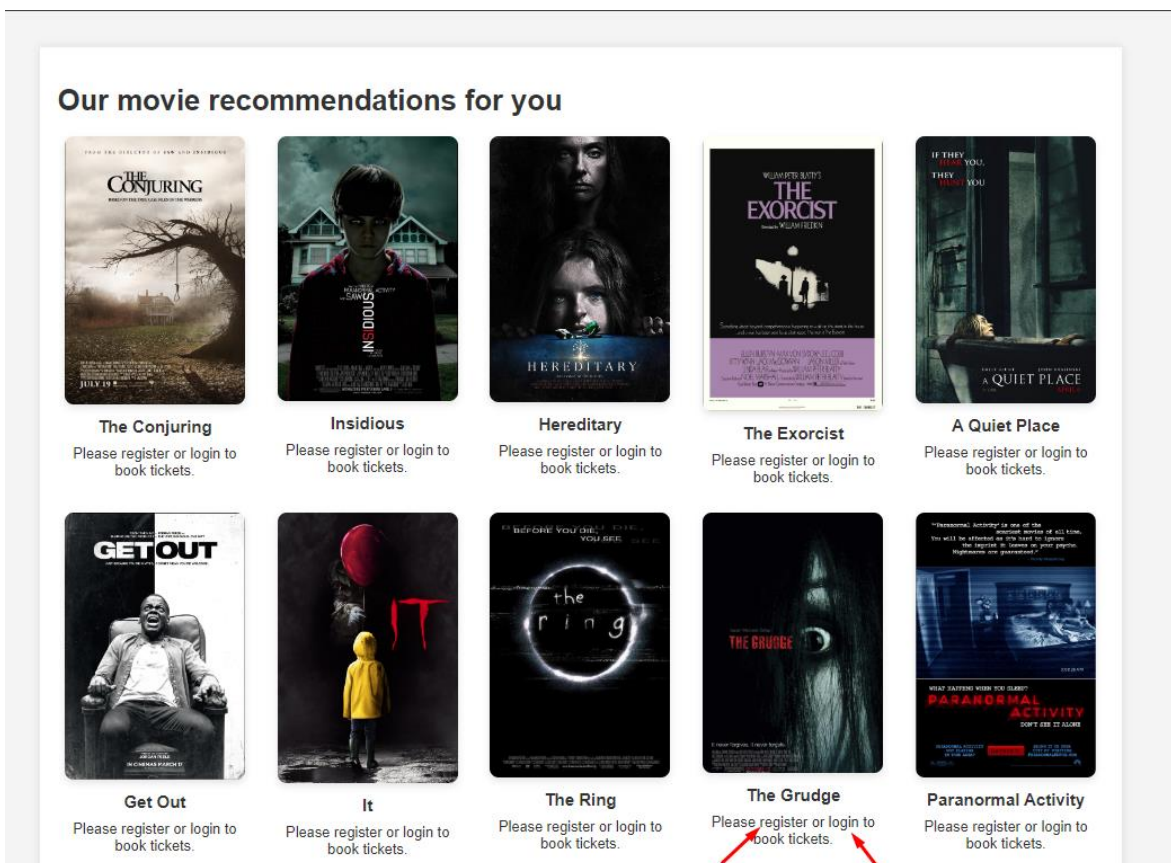
Subscribe to our newsletter to get updates on new releases, exclusive offers, and more.

[Subscribe](#)

© Fix 2024. All rights reserved. University Project by Konstantinos Rizos

Εικόνα 36 Αρχική Οθόνη(2)

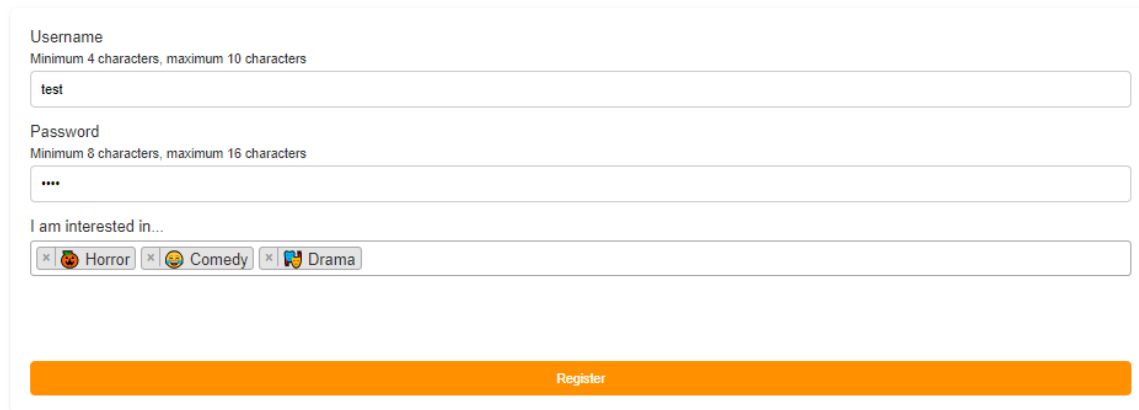
Αν πατήσετε την επιλογή «See More» θα σας εμφανιστούν όλες οι ταινίες που παίζονται αυτή την περίοδο στον κινηματογράφο αλλά δεν θα μπορείτε να κάνετε κράτηση αν δεν έχετε κάνει login στον λογαριασμό σας. Θα σας βγάλει ένα μήνυμα κάτω από κάθε ταινία « Please register or login to book tickets» . Αν πατήσετε σε αυτό το κείμενο, είτε το register είτε το login θα σας βγάλει στην αντίστοιχη φόρμα, δηλαδή στην φόρμα δημιουργίας λογαριασμού αν πατήσετε το register αλλιώς στην φόρμα εισόδου αν πατήσετε το login.



Εικόνα 37 Λίστα ταινιών

Ας περάσουμε τώρα στις 2 βασικές λειτουργίες της αρχικής οθόνης. Στο register και στο login. Αν δεν διαθέτετε ήδη λογαριασμό θα πρέπει να δημιουργήσετε έναν πατώντας την επιλογή register. Εκεί θα πρέπει να εισάγετε ένα username(4-10 χαρακτήρες), έναν κωδικό(8-16 χαρακτήρες) και να δηλώσετε τις κατηγορίες ταινιών(πχ. horror, comedy κτλ) που προτιμάτε. Αν δεν τηρήσετε τους παραπάνω κανόνες θα λάβετε αντίστοιχο μήνυμα που θα σας ενημερώνει για το λάθος πεδίο.

Create your Flix account



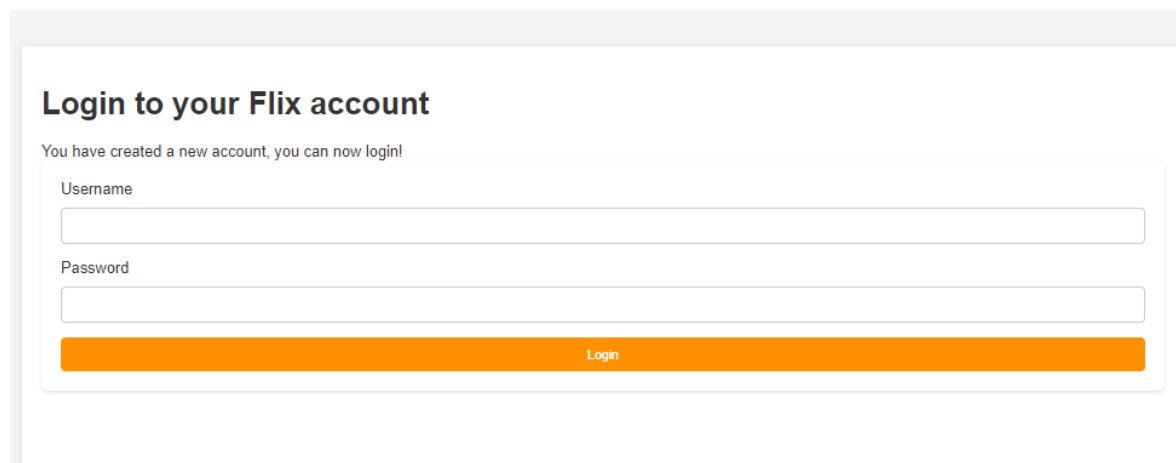
The screenshot shows a registration form with the following fields:

- Username:** A text input field containing the text "test". Below it, the text "Minimum 4 characters, maximum 10 characters" is displayed.
- Password:** A password input field with masked characters "****". Below it, the text "Minimum 8 characters, maximum 16 characters" is displayed.
- I am interested in...:** A category selection field containing three tags: "Horror" (with a horror icon), "Comedy" (with a comedy icon), and "Drama" (with a drama icon).

At the bottom of the form is a large orange button labeled "Register".

Εικόνα 38 Δημιουργία Λογαριασμού

Αφού δημιουργήσετε λογαριασμό, η εφαρμογή θα βγάλει μήνυμα επιτυχίας δημιουργίας λογαριασμού και θα σας μεταφέρει στην φόρμα login. Εκεί θα πρέπει να εισάγετε το username και το password και αν είναι σωστά η εφαρμογή θα σας μεταφέρει στην αρχική οθόνη, θα σας καλωσορίσει και θα σας εμφανίσει περισσότερες λειτουργίες μιας και είστε και επίσημα μέλος του Flix πλέον. Αλλιώς θα λάβετε μήνυμα για λάθος κωδικό ή username αντίστοιχα.



The screenshot shows a login form with the following elements:

- Header:** "Login to your Flix account".
- Message:** "You have created a new account, you can now login!".
- Username:** A text input field.
- Password:** A password input field.

At the bottom of the form is a large orange button labeled "Login".

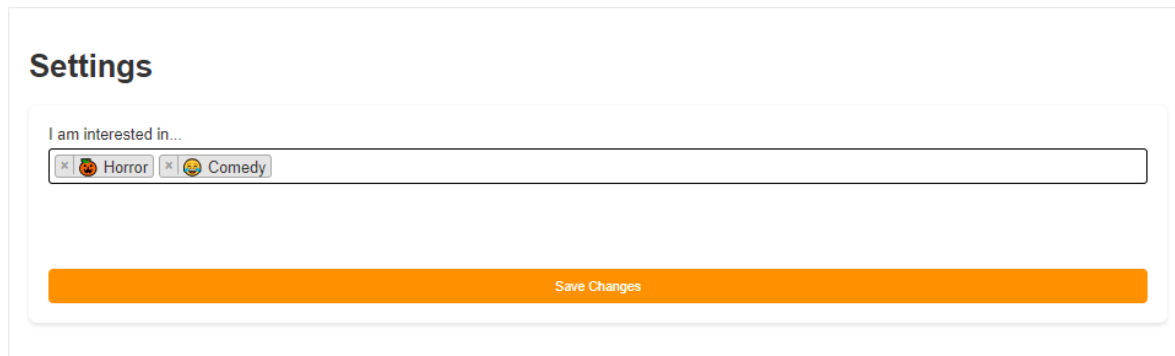
Εικόνα 39 Είσοδος Χρήστη



Εικόνα 40 Καλωσόρισμα Χρήστη

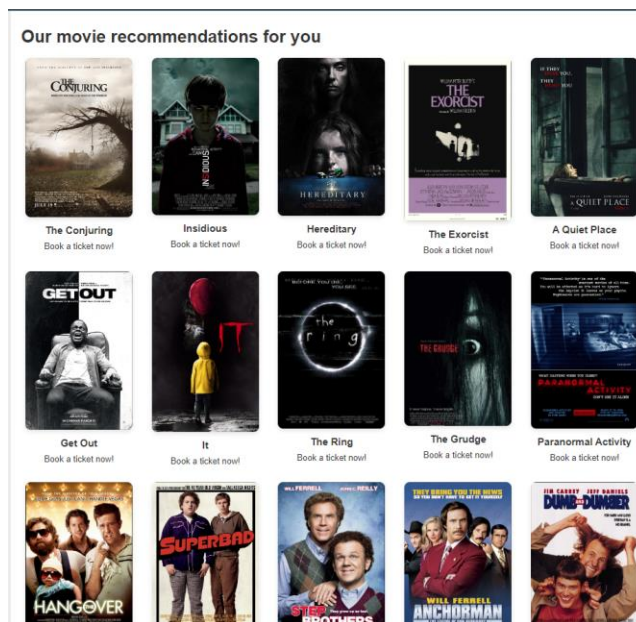
Το παραπάνω screenshot επειδή δεν φαίνεται καλά δείχνει τις επιπλέον λειτουργίες movies, my bookings, settings και σας καλωσορίζει (Welcome test1) μετά το login.

Αφού έχετε κάνει login στον λογαριασμό σας, η πιο απλή από τις νέες λειτουργίες είναι η επιλογή settings. Αυτή, σας δίνει την δυνατότητα να επεξεργαστείτε τις προτιμήσεις σας. Δηλαδή μπορείτε να προσθέσετε ή και να διαγράψετε κατηγορίες ταινιών.



Εικόνα 41 Επεξεργασία Προτιμήσεων Χρήστη

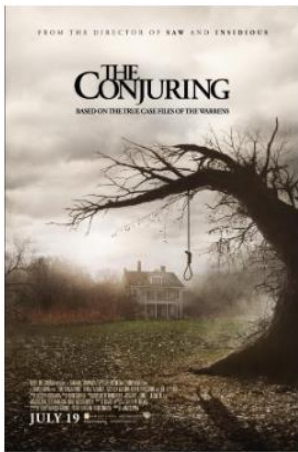
Μία από τις βασικές λειτουργίες είναι η λειτουργία Movies. Αφού «κλικάρετε» αυτή την επιλογή θα σας εμφανιστούν όλες οι διαθέσιμες ταινίες που παίζονται στους κινηματογράφους και ανήκουν στις κατηγορίες ταινιών που έχετε δηλώσει. Στο παρακάτω παράδειγμα οι κατηγορίες που έχουν δηλωθεί είναι comedy και horror. Αν αποφασίσετε για ποια ταινία θέλετε να αγοράσετε εισιτήριο, πρέπει να πατήσετε στο «Book a ticket now!» κάτω από την ταινία της αρεσκείας σας και θα μεταβείτε στο επόμενο βήμα.



Εικόνα 42 Λίστα Ταινιών με Βάση τις Προτιμήσεις του Χρήστη

Στην συνέχεια θα μεταβείτε στην σελίδα για να ολοκληρώσετε την κράτηση για την ταινία που διαλέξατε(πχ The Conjuring). Εκεί θα χρειαστεί να συμπληρώσετε όλα τα πεδία με τις επιλογές σας(ημερομηνία, ώρα, κινηματογράφο, αριθμό εισιτηρίων, θέσεις) καθώς και να εισάγετε τα στοιχεία της κάρτα σας(16ψηφιο αριθμό, όνομα κατόχου, λήξη, τριψήφιο αριθμό ασφαλείας) για να ολοκληρωθεί η ηλεκτρονική συναλλαγή. Με το πάτημα του κουμπιού Book tickets γίνεται η απαραίτητη διασφάλιση ότι τα δεδομένα είναι σωστά και αν δεν υπάρχει κάποιος λάθος ολοκληρώνεται η διαδικασία.

The Conjuring



FROM THE DIRECTOR OF SAW AND INSIDIOUS
THE CONJURING
BASED ON THE TRUE CASE FILES OF THE WARREN
JULY 19

Booking Details

Date

Hour

Cinema

Number of Tickets

Εικόνα 43 Συμπλήρωση Στοιχείων(1)

Number of Tickets

-

+

Select Seats

A	X	2	3	X	X	6	7	X	X	10
B	1	2	3	4	X	6	7	8	9	X
C	1	2	3	4	5	6	X	8	X	10
D	X	2	3	4	X	6	7	X	9	10
E	1	2	3	4	5	6	X	X	X	10

Payment Details

Credit Card Number

Expiration Date (MM/YY)

CVV

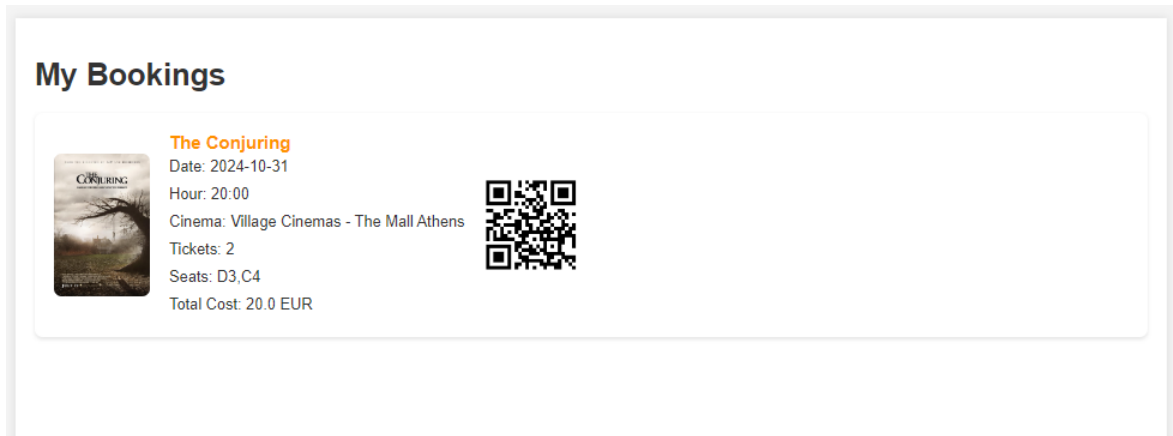
Name on Card

Book Tickets

Total Cost: 20 EUR

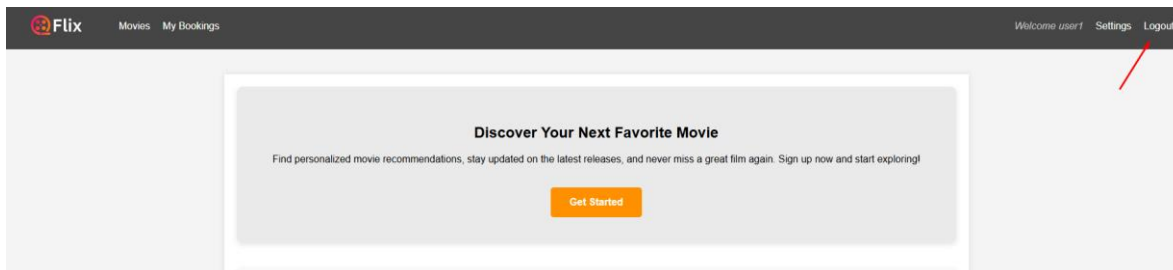
Εικόνα 44 Συμπλήρωση Στοιχείων(2)

Τέλος αφού ολοκληρωθεί η παραπάνω διαδικασία θα μεταφερθείτε στην σελίδα «My bookings» όπου εκεί φαίνονται όλες οι κρατήσεις σας με τις λεπτομέρειες καθώς και με ένα QR που θα χρησιμοποιήσετε για να εισέλθετε στον κινηματογράφο για την ταινία που κλείσατε.



Εικόνα 45 Κρατήσεις Χρήστη

Τέλος μπορείτε να κάνετε έξοδο από τον λογαριασμό σας πατώντας το κουμπί «logout» και να επιστρέψετε στην αρχική σελίδα.



Εικόνα 46 Έξοδος

Οφέλη που αναμένουμε να έχουμε από την λύση της Πτυχιακής

Ζούμε στον 21ο αιώνα, μια περίοδο που τα πάντα εξελίσσονται με εξωπραγματικά γρήγορους ρυθμούς. Η τεχνολογία έχει φτάσει σε ένα σημείο, όπου κανείς ποτέ δεν μπορούσε να φανταστεί. Η τεχνητή νοημοσύνη έχει «εισχωρήσει» σε πολλούς κλάδους και έχει φέρει τα πάνω κάτω στον τρόπο προσέγγισης των πραγμάτων. Η παρέμβαση του ανθρώπου πια, είναι αμελητέα στην πλειοψηφία των περιπτώσεων. Τα «έξυπνα σπίτια», τα ρομπότ που έχουν αντικαταστήσει τους ανθρώπους σε αρκετές εργασίες και τα αυτοκίνητα που δεν χρειάζονται οδηγό, είναι μερικά παραδείγματα για να καταλάβει κάποιος πόσο πολύ έχει αναπτυχθεί η τεχνολογία στις μέρες μας. Όπως είναι λογικό όμως, κάποιου κλάδοι δέχονται περισσότερο «προσοχή» και άλλοι λιγότερο. Με λίγα λόγια κάποια πράγματα δεν έχουν εξελιχθεί σημαντικά συγκριτικά με την γενική εξέλιξη των πραγμάτων.

Θα εστιάσουμε πιο συγκεκριμένα στον κινηματογράφο, ο οποίος είναι ο κεντρικός άξονας της πτυχιακής μου εργασίας. Δεν υπάρχει καμία αμφιβολία ότι ο κινηματογράφος έχει κάνει τεράστιο άλμα τα τελευταία χρόνια. Η ζήτηση του κόσμου για ποιοτικές ταινίες έχει αποτελέσει θεμέλιο ώστε να συμβεί το παραπάνω. Οι χώροι και οι εγκαταστάσεις των κινηματογράφων έχουν βελτιωθεί σημαντικά, οι παραγωγοί, οι ηθοποιοί και οι σκηνοθέτες ξεπερνούν τους εαυτούς τους μέρα με την μέρα και το προϊόν που καταλήγει στα «χέρια» μας, προσφέρει άπειρη ικανοποίηση. Άλλες φορές επειδή βλέπουμε τους «εαυτούς» μας να προβάλλονται σε ταινίες και άλλοτε επειδή αυτό που παρακολουθούμε μας ταξιδεύει σε διαφορετικούς κόσμους, βοηθώντας μας να ξεφύγουμε από την καθημερινότητα.

Για την πτυχιακή μου εργασία, αποφάσισα να ασχοληθώ με κάτι που υστερούσε σε κάποια πράγματα και είχε αρκετά περιθώρια βελτίωσης. Κατέληξα στο συμπέρασμα ότι από τα εργαλεία που χρησιμοποιούνται για την αγορά εισιτηρίων στους ελληνικούς κινηματογράφους, λείπουν βασικά χαρακτηριστικά. Για αυτό και ανέπτυξα το Flix, την διαδικτυακή εφαρμογή που θα έλυσε τα χέρια των χρηστών και θα ήταν εύκολη στην χρήση για όλους ανεξαρτήτως ηλικίας και τεχνογνωσίας.

Το Flix, αρχικά είναι απλό στην χρήση και δεν κουράζει τον χρήστη με τις περιττές λειτουργίες που παρατηρούμε σε άλλες εφαρμογές. Το User Interface είναι φτιαγμένο με τέτοιο τρόπο ώστε να είναι αισθητικά ωραίο στα μάτια του χρήστη, χωρίς όμως να υποκύπτει σε υπερβολές. Επίσης οι λειτουργίες του Flix είναι οι βασικές λειτουργίες που απαιτούνται για την αγορά ενός εισιτηρίου online. Θεωρούσα υπερβολή απλά να προσθέσω περιττές λειτουργίες μόνο και μόνο για να φαίνεται γεμάτη η ιστοσελίδα, χωρίς να προσφέρουν κάτι ουσιαστικό στην εμπειρία του χρήστη. Τέλος, αυτό που κάνει το Flix να ξεχωρίζει από παρόμοιες εφαρμογές είναι η εξατομικευμένη εμπειρία που παρέχει στον χρήστη. Ο χρήστης έχει την δυνατότητα να δηλώσει τις προτιμήσεις του σχετικά με τα είδη των ταινιών κατά την δημιουργία του λογαριασμού του. Στην συνέχεια μπορεί να τις επεξεργαστεί αν δεν είναι σίγουρος για κάτι. Έτσι δεν χρειάζεται κάποιος να χάνει χρόνο προσπελάζοντας ταινίες που δεν είναι του γούστου του. Μπορεί να επιλέξει κατευθείαν ταινία που ανήκει στο φάσμα προτιμήσεων του και να προχωρήσει στην αγορά εισιτηρίου γρήγορα και εύκολα χωρίς να χρειαστεί να κάνει αχρείαστα «βήματα».

Το όφελος που αναμένω από την ανάπτυξη αυτής της εφαρμογής είναι η τελευταία να αποτελέσει έμπνευση για άλλους προγραμματιστές. Να ξεκινήσουν από αυτό το σκεπτικό με την εξατομικευση του χρήστη και να ανεβάσουν το επίπεδο προσθέτοντας και άλλα χαρακτηριστικά που οι ίδιοι κρίνουν απαραίτητα.

Συμπεράσματα

Στην ενότητα αυτή, θα μιλήσω για τα συμπεράσματα που έχουν προκύψει από την ανάπτυξη της εφαρμογής, την εμπειρία μου μέσα από αυτή, αλλά και τις δυσκολίες που προέκυψαν καθ' όλη την διάρκεια. Θα αναλυθεί ο τρόπος που αυτές οι δυσκολίες ξεπεράστηκαν, άλλοτε με την εύρεση μια λύσης και άλλες φορές με επιτακτική αλλαγή κατεύθυνσης. Τέλος, θα επισημάνω τις επεκτάσεις στις οποίες έχω καταλήξει μέχρι και σήμερα που ασχολούμαι με την πτυχιακή εργασία.

Αρχικά, θεωρώ ότι έχω καταφέρει να καλύψω τα ζητούμενα της εργασίας, αφού έχω καταφέρει να υλοποιήσω μια διαδικτυακή εφαρμογή που αφορά μία ψυχαγωγική δραστηριότητα(κινηματογράφο) και εστιάζει στις προτιμήσεις του χρήστη. Αυτά τα στοιχεία δημιουργούν την εξατομικευμένη έννοια που είναι ένα βασικό ζητούμενο του project. Έχω να προσθέσω ότι ήταν ένα ιδιαίτερα δύσκολο project για μένα διότι «βγήκα λίγο από τα νερά μου». Η ευελιξία της εργασίας ήταν μεγάλη καθώς μπορούσα να χρησιμοποιήσω όποια τεχνολογία επιθυμούσα για την υλοποίηση αυτής. Η αρχική ιδέα ήταν να χρησιμοποιήσω C# και πιο συγκεκριμένα να φτιάξω μια Windows Forms App. Ξεκινώντας την υλοποίηση και χρησιμοποιώντας αυτό το Framework, κατέληξα στο συμπέρασμα ότι οι δυνατότητες ήταν περιορισμένες σε σχέση με άλλα «εργαλεία». Έτσι, με όλα τα δεδομένα που διέθετα και το πλάνο για την εργασία που είχα στο μυαλό μου, ξεκίνησα την αναζήτηση για ένα πιο «βολικό» περιβάλλον που θα μπορούσα να το χρησιμοποιήσω για να φέρω την εργασία στα μέτρα μου, έτσι όπως ακριβώς την σκεφτόμουν. Ύστερα από αρκετό «ψάξιμο» συνειδητοποίησα ότι η Python σε συνδυασμό με το Flask(microframework) ήταν αυτό που ακριβώς έψαχνα. Το σκεπτικό ήταν να γράψω μια εφαρμογή σε python και με την χρήση του Flask να την μετατρέψω σε διαδικτυακή. Αυτό βέβαια προϋπόθετε καλή γνώση της python καθώς και HTML, CSS κτλ. Θα είμαι ειλικρινής, στην python είχα κάποιες στοιχειώδεις γνώσεις, αλλά κατά κύριο λόγο ήταν κάτι καινούριο για μένα. Το είδα σαν μια νέα πρόκληση λοιπόν και αφιέρωσα αρκετό χρόνο μελετώντας, βλέποντας tutorials και κάνοντας διάφορα σύντομα projects ώστε να την κατανοήσω σε βάθος. Αφού έφτασα σε ένα καλό επίπεδο και έμαθα πως να χρησιμοποιώ τα διάφορα πακέτα της, αποφάσισα να ξεκινήσω και επίσημα την υλοποίηση του Flix. Ωστόσο, έπρεπε να χρησιμοποιήσω και HTML καθώς και CSS για το “styling” της σελίδας. Είχα λίγο και καιρό να ασχοληθώ με τα παραπάνω, αλλά μια γρήγορη ματιά ήταν ότι ακριβώς χρειαζόμουν για να κάνω ένα γρήγορο «φρεσκάρισμα» στις γνώσεις μου.

Όλες οι παραπάνω αποφάσεις ανέβασαν πραγματικά το επίπεδο στην εργασία και δοκίμασαν τα όριά μου ως προγραμματιστής. Μπορώ να πω με σιγουριά πάντως, ότι αυτό είναι το μεγαλύτερο project που έχω αναλάβει και με την ολοκλήρωσή, του έχω βελτιωθεί σε μεγάλο βαθμό και ο ίδιος όσον αφορά τον προγραμματισμό. Η ανάπτυξη του λογισμικού απαιτούσε μεγάλη έρευνα, αφού οι συντριπτική πλειοψηφία των λειτουργιών που έπρεπε να εφαρμόσω ήταν κάτι καινούριο για μένα. Τελικά, είμαι αρκετά ευχαριστημένος με το τελικό αποτέλεσμα, αφού η εφαρμογή είναι αρκετά σταθερή και ανταποκρίνεται σε μεγάλο βαθμό στους στόχους που είχα θέσει από την μέρα που είχα αναλάβει το project.

Σε αυτή την παράγραφο θα επικεντρωθώ στις δυσκολίες με τις οποίες ήρθα αντιμέτωπος κατά την ανάπτυξη του Flix. Αρχικά, η πρώτη δυσκολία που συνάντησα ήταν η προσπάθεια κατασκευής ενός απλού, ωραίου αλλά και παράλληλα φιλικό προς τον χρήστη περιβάλλον. Όπως αναφέραμε και σε προηγούμενες ενότητες, οι περισσότερες παρόμοιες ιστοσελίδες έχουν μεν όμορφο interface αλλά συγχύζουν τον χρήστη με τις πολλές πληροφορίες και λειτουργίες που του προσφέρουν. Έτσι χρησιμοποίησα HTML και CSS για να φτιάξω ένα όμορφο αισθητικά, αλλά και πρακτικό περιβάλλον. Μια επίσης μεγάλη πρόκληση ήταν να αφομοιώσω στην εφαρμογή έναν εξατομικευμένο χαρακτήρα. Αυτό είναι το χαρακτηριστικό του Flix που το κάνει να ξεχωρίζει από τις υπόλοιπες εφαρμογές. Έπρεπε λοιπόν να φτιάξω τις βάσεις δεδομένων να συνδέονται με τέτοιο τρόπο ώστε να προβάλλονται σε κάθε χρήστη, ταινίες, που ανήκουν στις «αγαπημένες» του κατηγορίες.

Τέλος, μία δυσκολία που αντιμετώπιζα όλο αυτό τον καιρό και δεν έχει σχέση με κώδικα, είναι πως η ανάπτυξη του λογισμικού πραγματοποιήθηκε σε υπολογιστή χαμηλής απόδοσης. Το πρόβλημα υπήρξε κυρίως εξαιτίας των επεξεργαστικών απαιτήσεων που χρειάζεται το PyCharm, και κατά συνέπεια ολόκληρης της εφαρμογής. Αυτό οδηγούσε σε αρκετά αυξημένους χρόνους επεξεργασίας κάθε αλλαγής στην εφαρμογή και συνεπώς σε μία μέτρια εμπειρία προγραμματισμού.

Από τις δυσκολίες θα καταλήξω στις πιθανές μελλοντικές επεκτάσεις της εφαρμογής. Παρακάτω θα αναφερθούν κάποιες ιδέες που σκεφτόμουν κατά την διάρκεια ανάπτυξης της εφαρμογής. Είναι ιδέες που εφαρμόζονται σχετικά εύκολα, με βάση τον τρόπο με τον οποίο είναι προγραμματιστικά στημένο το app. Παραθέτω παρακάτω:

- Προσθήκη περισσότερων ψυχαγωγικών δραστηριοτήτων: Το Flix στην παρούσα φάση εστιάζει αποκλειστικά στον κινηματογράφο. Στο μέλλον μπορούν να προστεθούν και άλλες κατηγορίες δραστηριοτήτων όπως συναυλίες, θέατρο κτλ.
- Πιο έξυπνος Κώδικας και περισσότερη Αξιοποίηση AI: Η εφαρμογή με χρήση AI θα καταλάβαινε μοτίβα στις προτιμήσεις του χρήστη και θα του πρότεινε ακόμα πιο συγκεκριμένες ταινίες ανάλογα με τα γούστα του.
- Προσθήκη ρόλων-χρηστών: Στο μέλλον θα μπορούσε να προστεθεί ένας ρόλος διαχειριστή ο οποίος θα ενημέρωνε, θα πρόσθετε και θα διέγραφε ταινίες αλλά θα μπορούσε να επεξεργαστεί και τους λογαριασμούς των χρηστών.
- Styling Εφαρμογής και προσθήκη επιπλέον λειτουργιών: Στο αισθητικό κομμάτι της εφαρμογής έχω δώσει τόση έμφαση ώστε να είναι ωραίο στο μάτι του χρήστη. Θα μπορούσαν να γίνουν αρκετές βελτιώσεις ώστε να ήταν ακόμα πιο όμορφο αλλά αυτό δεν ήταν η κύρια προτεραιότητά μου. Τέλος, υπάρχουν αρκετές λειτουργίες που μπορούν να προστεθούν στο Flix αλλά όπως ανέφερα κύριο μέλημά μου ήταν να φτιάξω μια φιλική προς τον χρήστη εφαρμογή που θα του λύσει τα χέρια χωρίς να του προκαλεί σύγχυση.

Βιβλιογραφία

[1] “How To Make a Web Application Using Flask in Python 3”, by Abdelhadi Dyoury and Kathryn Hancox, DigitalOcean, December 2022, <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3>

[2] “Το μέλλον της εξατομίκευσης: Πώς η τεχνητή νοημοσύνη και η μηχανική μάθηση μεταμορφώνουν το ψηφιακό μάρκετινγκ”, by Jussi Hyvärinen, Rantracker, June 2023, <https://www.ranktracker.com/el/blog/the-future-of-personalization-how-ai-and-machine-learning-are-transforming-digital-marketing/>

[3] “Beautify Flask Web App using CSS, HTML”, by Yue Weng Mak, Medium, March 2020, <https://medium.com/an-idea/beautify-flask-web-app-using-css-html-d574332f710f>

[4] “Enhance Your Flask Web Project With a Database”, by Philipp Ascany, Real Python, December 2023, <https://realpython.com/flask-database/>

[5] Γερογιάννης, Βασίλης. Αντικειμενοστραφής Ανάπτυξη Λογισμικού με τη UML, Εκδόσεις Κλειδάριθμος, 2006

[6] “Village Cinemas”, Screenshots On Chrome, <https://www.villagecinemas.gr/el/>

[7] “Basic Concepts of Python Programming”, by teacher, Purple Tutor, February 2023, <https://purpletutor.com/coding/basic-concepts-python/>

[8] “Options Cinemas”, Screenshots on Chrome, <https://www.options-cinemas.gr/>

[9] Python Documentation. Retrieved from <https://docs.python.org/3/>

[10] SQLite Documentation. Retrieved from <https://www.sqlite.org/docs.html>

[11] Γκάντις, Τόνυ. Starting Out with Python, Εκδόσεις Da Vinci, 2015

[12] Rautio, K. (2022). Design and implementation of a web application for practising internet reading skills. Master of Science in Technology Thesis, University of Turku, Department of Computing

[13] Sonavane, N. (September 4, 2018). Characteristics of a web application. Retrieved from <https://www.linkedin.com/pulse/characteristics-web-application-nripesh-sonavane>

[14] Huang, Wei & Li, Ru & Maple, Carsten & Yang, Hongji & Foskett, David & Cleaver, Vince. (2010). A Novel Lifecycle Model for Web-based Application Development in Small and Medium Enterprises. *International Journal of Automation and Computing*. 7. 389-398

[15] Gurung, Gagan & Shah, Rahul & Jaiswal, Dhiraj. (2020). Software Development Life Cycle Models-A Comparative Study. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 30-37

[16] Tsihrintzis, G. A., Virvou, M., & Jain, L. C. (2016). *Intelligent Computing Systems: Emerging Application Areas* (pp. 1-4). Springer Berlin Heidelberg

[17] Virvou, M., 2023. Artificial Intelligence and User Experience in reciprocity: Contributions and state of the art. *Intelligent Decision Technologies* 17 (2023) 73–125 73 DOI 10.3233/IDT-230092 IOS Press

[18] Virvou, M., 2018, July. A new era towards more engaging and human-like computer-based learning by combining personalisation and artificial intelligence techniques. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 2 3)

[19] Virvou, M., Tsihrintzis, G. A., & Jain, L. C. (2022). Introduction to advances in selected artificial intelligence areas. In *Advances in Selected Artificial Intelligence Areas: World Outstanding Women in Artificial Intelligence* (pp. 1-7). Cham: Springer International Publishing

[20] Hatzilygeroudis, I., Tsihrintzis, G., Virvou, M., & Perikos, I. (2023). Special issue on information, intelligence, systems and applications. *Neural Computing and Applications*, 35(1), 1-2

[21] Sarabjot Singh Anand, Bamshad Mobasher (2005): *Intelligent Techniques for Web Personalization*. Lecture Notes in Artificial Intelligence (LNAI 3169), Springer

[22] A. Kumar, Dr. P. Thambidurai (1/2010): *Collaborative Web Recommendation Systems -A Survey Approach*. *Global Journal of Computer Science and Technology* Vol. 9 Issue 5 (Ver 2.0)

[23] Wei Chu, Seung-Taek Park (2009): *Personalized Recommendation on Dynamic Content Using Predictive Bilinear Models*. *Proceedings of the 18th International World Wide Web Conference, Social Networks and Web 2.0 / Seddion: Recommender Systems*

[24] Kuźelewska U (2014) Clustering algorithms in hybrid recommender system on MovieLens data. *Studies in logic, grammar and rhetoric* 37(1):125–139

[25] Hassanzadeh O, Consens M (2009) Linked movie data base. *Proceedings of the 2nd Workshop on Linked Data on the Web, April, Spain. (LDOW2009)*

[26] *The Firm: The Story of McKinsey and Its Secret Influence on American Business* (A Business Bestseller) Paperback – September 30, 2014

[27] O. Nasraoui, "World wide web personalization," in In J. Wang (ed), *Encyclopedia of Data Mining and Data Warehousing*, Idea Group, 2005

[28] C. Wei, W. Sen, Z. Yuan, and C. Lian-Chang, "Algorithm of mining sequential patterns for web personalization services," SIGMIS Database, vol. 40, no. 2, pp. 57–66, Apr. 2009

[29] M. Albanese, A. Picariello, C. Sansone, and L. Sansone, "Web personalization based on static information and dynamic user behavior," in Proceedings of the 6th annual ACM international workshop on Web information and data management, ser. WIDM '04. New York, NY, USA: ACM, 2004, pp. 80–87.

[30] C.-H. Lee, Y.-H. Kim, and P.-K. Rhee, "Web personalization expert with combining collaborative filtering and association rule mining technique," Expert Systems with Applications, vol. 21, no. 3, pp. 131 – 137, 2001.