

ΠΜΣ Ευφυείς Τεχνολογίες Διαδικτύου

Μάθημα: Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές

ΑΝΑΦΟΡΑ ΥΛΟΠΟΙΗΣΗΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ

Movie Info

Ομάδα: Γκατζάρας Ιωάννης, Ελευθεριάδης Νίκος, Κουσκουβέλης Νίκος

ΙΑΝΟΥΑΡΙΟΣ 2018

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή	3
2. Εργαλεία που χρησιμοποιήθηκαν	4
3. Αρχιτεκτονική της εφαρμογής	5
4. Περιγραφή της εφαρμογής	10
5. The Movie DB API	12
6. Χρονοδιάγραμμα υλοποίησης της εφαρμογής	15
7. Δυσκολίες οργάνωσης και προγραμματισμού	15
8. Αξιολόγηση της εφαρμογής	15
9. Πηγές – Παραδείγματα	15

1. Εισαγωγή

Η εφαρμογή που αναπτύχθηκε από την ομάδα μας αφορά την αναζήτηση και την προβολή πληροφοριών ταινιών που προβάλλονται στους κινηματογράφους.

Είναι μια χρήσιμη πιστεύουμε εφαρμογή για τους σινεφίλ γιατί προσφέρει ένα εύχρηστο και απλό περιβάλλον μέσω του οποίου ο χρήστης θα λάβει τις πληροφορίες που αναζητά.

Υπάρχουν πολλές εφαρμογές που κάνουν αναζήτηση και προβολή ταινιών, με την διαφορά ότι έχουν τεράστιο όγκο πληροφοριών και χρησιμοποιούν βάση δεδομένων για την αποθήκευση αυτών. Κάποιες από αυτές είναι οι The Movie Database (TMDb), retro DB, IMDb, popcorn time, ενώ υπάρχουν και εφαρμογές που υποστηρίζουν σε κάποιες κατηγορίες του μενού τους μεταξύ των άλλων την εμφάνιση και προβολή ταινιών με ελάχιστες πληροφορίες όμως, όπως το athinorama.gr.

Εμείς μελετήσαμε το The Movie Database (TMDb) από το οποίο μας δόθηκε API key για ακαδημαϊκή χρήση, με τον μόνο περιορισμό να χρησιμοποιηθεί το logo της εφαρμογής σε εμφανή σημείο και να είναι σύνδεσμος στην ιστοσελίδα τους. Περιοριστήκαμε μόνο στην ενότητα ταινίες κινηματογράφου και με λιτό μα χρηστικό interface προβάλουμε πληροφορίες για την ταινία που αναζητά ο χρήστης, εμφανίζουμε τις δημοφιλέστερες ταινίες (βάση ψηφοφορίας, άνω του 5) καθώς και τις ταινίες που παίζονται τώρα στους κινηματογράφους.

Το link της εφαρμογής μας είναι: <https://users.it.teithe.gr/~ait122017/>

2. Εργαλεία που χρησιμοποιήθηκαν

Οι τεχνολογίες (γλώσσες προγραμματισμού) που χρησιμοποιήθηκαν ώστε δημιουργηθεί η εφαρμογή και να υπάρξει η κατάλληλη αλληλεπίδραση με την βάση δεδομένων της TMDb ώστε να υπάρξει ανταπόκριση μεταξύ Server και Client ήταν οι εξής:

➤ Από την πλευρά του Client

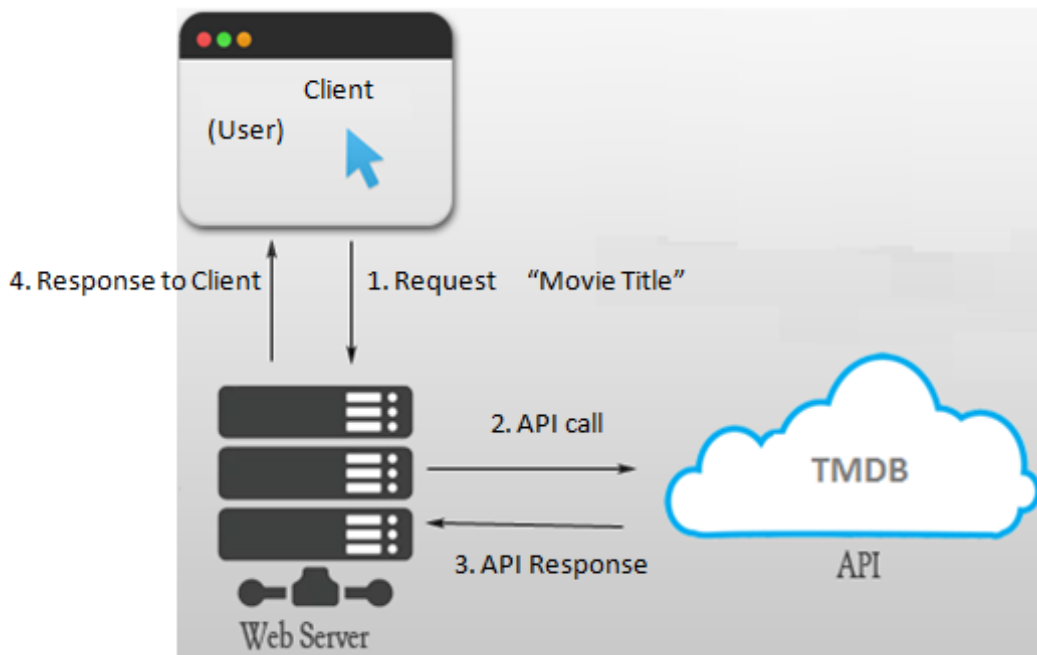
- Html 5
- JavaScript
- JQuery
- Ajax
- Css
- Json
- Bootstrap 4 beta 3

➤ Από την πλευρά του Server

- Apache Server
- Php

Χρησιμοποιήσαμε την πλατφόρμα PhpStorm 2017.2.4 για να δουλεύουμε, να εκτελούμε τον κώδικα (μέσω XAMPP) των παραπάνω γλωσσών, για την συνεργασία μας στο Project και τον διαμοιρασμό του κώδικα το GitHub, που είναι ένα social repository για projects ανοιχτού κώδικα που βασίζονται πάνω στο Git version control system.

3. Αρχιτεκτονική της εφαρμογής



Γενική περιγραφή

Ο χρήστης (user) μέσω του πλαισίου αναζήτησης στην αρχική σελίδα (index.php), δίνοντας τον τίτλο της ταινίας ζητά από τον Server να του επιστραφούν τα επιθυμητά αποτελέσματα. Η παράμετρος αποθηκεύεται τοπικά σε μια μεταβλητή και κατασκευάζεται ερώτημα με jquery προς τη βάση δεδομένων TMDb μέσω μεθόδου GET χρησιμοποιώντας το API key που μας δόθηκε .

Ο Server, έπειτα από την επιτυχή αποστολή του ερωτήματος, επιστρέφει μία απάντηση σε μορφή json (τη μόνη που υποστηρίζει το TMDb). Αυτή την απάντηση τη διαχειριζόμαστε μέσω ajax. Η συνολική απάντηση αντιστοιχείται σε μια μεταβλητή και στη συνέχεια μέσω JavaScript δημιουργούμε ένα δυναμικό DIV στο οποίο τοποθετούμε τα στοιχεία προς εμφάνιση κάθε φορά.

Δημιουργήσαμε και ορίσαμε τα παρακάτω αρχεία:

- το **index.php** ως σελίδα υποδοχής, που περιλαμβάνει το πεδίο αναζήτησης και το χώρο εμφάνισης των αποτελεσμάτων.
- το **index_2.php** ως σελίδα εμφάνισης αποτελεσμάτων των δημοφιλέστερων ταινιών (βάση βαθμολογίας).

Movie Info

Movie Search Project

- το ***index_3.php*** ως σελίδα εμφάνισης αποτελεσμάτων των ταινιών που προβάλλονται τώρα στους κινηματογράφους.
- το ***show_movie.php*** ως σελίδα εμφάνισης επιπλέον πληροφοριών για την ταινία που θα επιλέξουμε
- το ***MoviesInfo.css***, ως (εξωτερικό) αρχείο που περιλαμβάνει τον ορισμό των μορφοποιήσεων των διαφόρων αντικειμένων που χρησιμοποιούμε μέσα στη ιστοσελίδα μας.
- το ***header_menu.php*** ως μπάρα πλοήγησης στο άνω μέρος της σελίδας μας, που επαναλαμβάνεται σε όλες τις σελίδες μιας και περιέχει το link για την επιστροφή στην αρχική σελίδα (***index.php***) και στο site ***themoviedb.org***.
- το ***jquery-3.2.1.js*** ως βασική βιβλιοθήκη.
- τον φάκελο ***functions*** που περιέχονται όλες οι συναρτήσεις μας σε JavaScript (.js) αρχεία, μέσω των οποίων υλοποιούνται οι βασικότερες λειτουργίες της εφαρμογής και περιλαμβάνει:
 - το ***enter_key.js***, περιλαμβάνει τη συνάρτηση ***enterSearch(e)*** η οποία ανιχνεύει την περίπτωση που θα πατήσει ο χρήστης το enter αντί για το κουμπί Search ώστε να ξεκινήσει η αναζήτηση και εμφάνιση των αποτελεσμάτων. Η συνάρτηση δουλεύει κανονικά είτε δοθεί κάποιο όνομα είτε όχι.
 - το ***now_playing_movies.js***, περιλαμβάνει τη συνάρτηση ***nowPlayingMovies()*** που ανακτά και εμφανίζει από την βάση δεδομένων της TMDB τις ταινίες που προβάλλονται τώρα στις κινηματογραφικές αίθουσες.
 - το ***popular_movies.js***, περιλαμβάνει τη συνάρτηση ***popularMovies()*** που ανακτά και εμφανίζει από την βάση δεδομένων της TMDB τις δημοφιλέστερες ταινίες βάση ψηφοφορίας.
- το ***redirects.js***, περιλαμβάνει δυο συναρτήσεις που είναι οι ***redirectToIndex2()***, ***redirectToIndex3***, οι οποίες όταν καλεστούν μας οδηγούν στις σελίδες ***index_2.php*** και ***index_3.php*** αντίστοιχα.

- το *search_functions.js*, περιλαμβάνει τη συνάρτηση *movieSearch(x)* που αναζητά και προβάλλει τα συνολικά αποτελέσματα για την ταινία που ψάχνουμε

Ειδική περιγραφή

function movieSearch(x): Πρόκειται για τη συνάρτηση που υλοποιεί την κύρια λειτουργία της εφαρμογής Movie Info.

- Αρχικά, παίρνει τη παράμετρο που έχει πληκτρολογήσει ο χρήστης στο ειδικό text box αναζήτησης ταινίας και την αποθηκεύει τοπικά στη μεταβλητή userInput.

- Έπειτα κατασκευάζει το ερώτημα (με jquery) προς τη βάση δεδομένων της TMDB, όπου αιτείται μία απάντηση υπό τη μορφή json αντικειμένου. Πρόκειται για μία κλήση GET, της οποίας το URL είναι:

[https://api.themoviedb.org/3/search/movie?api_key=b5456db86ace1556b60313e04972fc9f&query="+userInput](https://api.themoviedb.org/3/search/movie?api_key=b5456db86ace1556b60313e04972fc9f&query=)

Στο URL προστίθεται το κλειδί που μας δόθηκε από την TMDB και η userInput.

- ο Server, έπειτα από την επιτυχή αποστολή του ερωτήματος, επιστρέφει μία απάντηση σε μορφή json (τη μόνη που υποστηρίζει το tmdb). Επιστρέφει ένα αντικείμενο json της μορφής (Object { page: 1, total_results: 73572, total_pages: 3679, results: [...] }).

Αυτό το αποτέλεσμα (την settings) το διαχειριζόμαστε μέσω ajax. Η συνολική απάντηση αντιστοιχείται στη μεταβλητή response. Με τη response και τα ονόματα των παραμέτρων που υπάρχουν στο αντικείμενο json, έχουμε πρόσβαση στα διάφορα στοιχεία που μας δίνει το API προς εμφάνιση (όπως: τίτλος ταινίας, εικόνα ταινίας, περίληψη, ονόματα ηθοποιών, κ.α).

- Κατασκευάζουμε δυναμικά ένα div (με javascript) μέσα στο οποίο τοποθετούμε τα στοιχεία της ταινίας που επιλέγουμε να εμφανίσουμε (και αυτά δυναμικά κατασκευασμένα). Λέγοντας δυναμικά, εννοούμε πως τα στοιχεία DIV, P, IMG, δημιουργούνται ξανά με κάθε νέα αναζήτηση του χρήστη και “στέλνονται” με appendChild στο body της index.php.

Δημιουργούμε ένα loop ώστε να προβληθούν αρχικά τα πρώτα τρία αποτελέσματα της αναζήτησης για τρεις ταινίες αρχικά (ο αριθμός των επαναλήψεων καθορίζεται από παράμετρο που περνά στη συνάρτηση).

- Ο τίτλος της κάθε ταινίας, είναι ενεργός σύνδεσμος προς τη σελίδα `show_movie.php`, όπου εμφανίζονται περισσότερες πληροφορίες για μία ταινία.

- Υπάρχει έλεγχος και εμφάνιση σχετικού μηνύματος στις περιπτώσεις: α) όταν ο χρήστης πληκτρολογήσει μία ταινία που δεν υπάρχει στη βάση δεδομένων της TMDB ή κάποιο λανθασμένο όνομα, β) όταν πατηθεί το enter χωρίς να δοθεί όνομα προς αναζήτηση και γ) όταν πατηθεί μόνο ο χαρακτήρας κενού (space) και προσπαθήσει να βρει την ταινία αυτή.

- Τέλος, έπειτα από τη σωστή εμφάνιση των αποτελεσμάτων, γίνεται ορατό ένα κουμπί για την εμφάνιση περισσότερων αποτελεσμάτων-ταινιών καθώς και έλεγχος για το πόσες ταινίες θα εμφανιστούν.

- **show more button:** Στο σημείο εμφάνισης ταινιών μετά την αναζήτηση γίνεται έλεγχος για το πόσα αποτελέσματα επέστρεψε το TMDB και ανάλογα εμφανίζονται οι αντίστοιχες ταινίες στο site καθώς και το κουμπί "Show More Movies!". Εάν τα αποτελέσματα είναι λιγότερα ή ίσα με 3, δεν εμφανίζεται το κουμπί και εμφανίζονται όσα αποτελέσματα λήφθηκαν. Εάν τα αποτελέσματα είναι παραπάνω από 3 εμφανίζονται αρχικά τα 3 πρώτα αποτελέσματα και έπειτα πατώντας το κουμπί εμφανίζονται τα υπόλοιπα. Μετά από δοκιμές, ο μέγιστος αριθμός αποτελεσμάτων που λαμβάνονται είναι 20.

function showMovieWithID(id): η μέθοδος αυτή λειτουργεί όπως η `movieSearch(x)` με τη διαφορά ότι γίνεται κλήση στο API για μία μόνο ταινία χρησιμοποιώντας το `id` της ταινίας που λαμβάνεται. Το `id` λαμβάνεται χρησιμοποιώντας τη μέθοδο GET μέσω της PHP. Ο τίτλος των ταινιών που ήρθαν σαν αποτέλεσμα της αναζήτησης είναι ενεργός σύνδεσμος που με το κλικ καλείται η σελίδα `show_movie.php` και περνάει σαν παράμετρος στο URL το `id` της ταινίας. Μετά τη φόρτωση της `show_movie.php` στέλνεται το `id` της ταινίας (που υπάρχει στο URL) στην μέθοδο και γίνεται η αναζήτηση με `id`. Τέλος, με τη μέθοδο αυτή εμφανίζονται επιπλέον πληροφορίες στη `show_movie.php` για τη συγκεκριμένη ταινία που πριν δεν εμφανίζονταν στα αποτελέσματα αναζήτησης.

Movie Info Movie Search Project

function popularMovies(): εμφανίζει τις επίκαιρες και δημοφιλείς ταινίες, όπως αυτές έχουν προκύψει από την βαθμολογία χρηστών της σελίδας themoviedb.org

function nowPlayingMovies(): εμφανίζει τις ταινίες που προβάλλονται τώρα στους κινηματογράφους.

Οι δύο συναρτήσεις αυτές, είναι όμοια κατασκευασμένες όπως και η συνάρτηση movieSearch(x). Η διαφορά τους με τη movieSearch(x), είναι πως σε αυτές δεν ορίζουμε παράμετρο για τις επαναλήψεις. Δε δέχονται παράμετρο που να έχει δοθεί από τον χρήστη (και συνεπώς δεν έχουν αντίστοιχο έλεγχο για το αν υπάρχει αυτό που πληκτρολόγησε ο χρήστης) και εμφανίζονται τα 20 αποτελέσματα (τόσα μας δίνει το API) που αντλούνται από τη βάση δεδομένων της TMDb.

Τι είναι το bootstrap;

Το bootstrap είναι ένα framework το οποίο συνδυάζει css και javascript (την βιβλιοθήκη jQuery) για να μας παρέχει έναν πιο δομημένο και εύκολο τρόπο σχεδιασμού και ανάπτυξης ιστοσελίδων. Επειδή η πλοήγηση στο διαδίκτυο από κινητά τηλέφωνα και tablets μεγαλώνει και τείνει να καταλάβει την πρώτη θέση ανάμεσα τρόπους προβολής ιστοσελίδων. Έτσι δημιουργήθηκε η ανάγκη για ιστοσελίδες σχεδιασμένες με τέτοιο τρόπο ώστε όταν αλλάζει το μέγεθος του μέσου προβολής (pc, tablet, mobile) να αλλάζει το μέγεθος, η δομή και η εμφάνιση των στοιχείων, για να είναι σωστή η εμφάνιση της ιστοσελίδας. Μια κατ' αυτό τον τρόπο δομημένη ιστοσελίδα ονομάζεται responsive.

Εμείς χρησιμοποιήσαμε το bootstrap σε κάθε σελίδα της εφαρμογής μας κυρίως στην γραμμή πλοήγησης, στο πάνω μέρος.

Εισάγοντας τον παρακάτω κώδικα:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/css/bootstrap.min.css" integrity="sha384-Zug+QiDoJOrZ5t4lssLdxGhVrurbmBWopoEl+M6BdEfwnCJZtKxi1KgxUyJq13dy" crossorigin="anonymous">
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.3/js/bootstrap.min.js" integrity="sha384-a5N7Y/aK3qNeh15eJKGWxsqtnX/wWdSZSKp+81YjTmS15nvnvxKHuzaWwXHDli+4" crossorigin="anonymous"></script>
```

και για mobile responsive περιβάλλον:

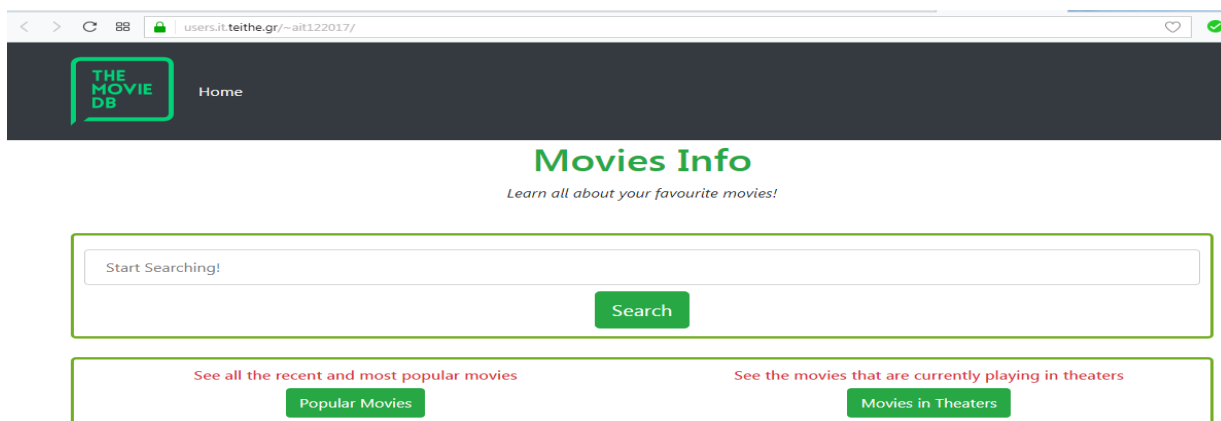
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
εισάγεται η βιβλιοθήκη bootstrap με την οποία κάναμε εισαγωγή κάποια γραφικά που μας παρέχει. Χρησιμοποιώντας κλάσεις του bootstrap στα κατάλληλα αντικείμενα DOM της σελίδας (π.χ. div, button, p κ.α.) προσαρμόζουμε τα επιθυμητά γραφικά που θέλουμε.

Το πλεονέκτημα είναι ότι το site έγινε mobile friendly καθώς αλλάζοντας τις διαστάσεις της οθόνης αλλάζουν αντίστοιχα και τα αντικείμενα, έτσι για παράδειγμα, το κουμπί search που είναι στο κέντρο, θα παραμείνει στο κέντρο και σε κινητές συσκευές με μικρή οθόνη. Επίσης το div που αφορά την επιλογή για τις δημοφιλείς ταινίες και ταινίες που παίζονται τώρα είναι χωρισμένο σε 2 στήλες μέσω της κλάσης col-md-6 και όταν η οθόνη είναι μικρή το ένα αντικείμενο έρχεται κάτω από το άλλο για να εμφανίζονται επαρκώς.

--header_menu.php: στο header_menu.php το οποίο καλείται από κάθε εμφανιζόμενη σελίδα είναι η μπάρα πλοήγησης η οποία προσαρμόζεται να επεκτείνεται σε όλη τη γραμμή της (κάθε διάστασης) οθόνης. Στο αριστερό κομμάτι υπάρχει το logo (που λαμβάνεται διαδικτυακά) της Movie DB το οποίο πατώντας το γίνεται μεταφορά στο site της. Κατόπιν ορίζεται το κουμπί που εμφανίζεται σε μικρές οθόνες όπως τα κινητά και μέσα περιέχει το home button καθώς δε χωράει στη μπάρα. Όταν η οθόνη είναι μεγάλη το κουμπί δεν εμφανίζεται. Τέλος το κουμπί home εμφανίζεται σε κάθε εμφανιζόμενη σελίδα ώστε ο χρήστης να μπορεί να επανέλθει στην αρχική σελίδα σε περίπτωση που χαθεί κάπου.

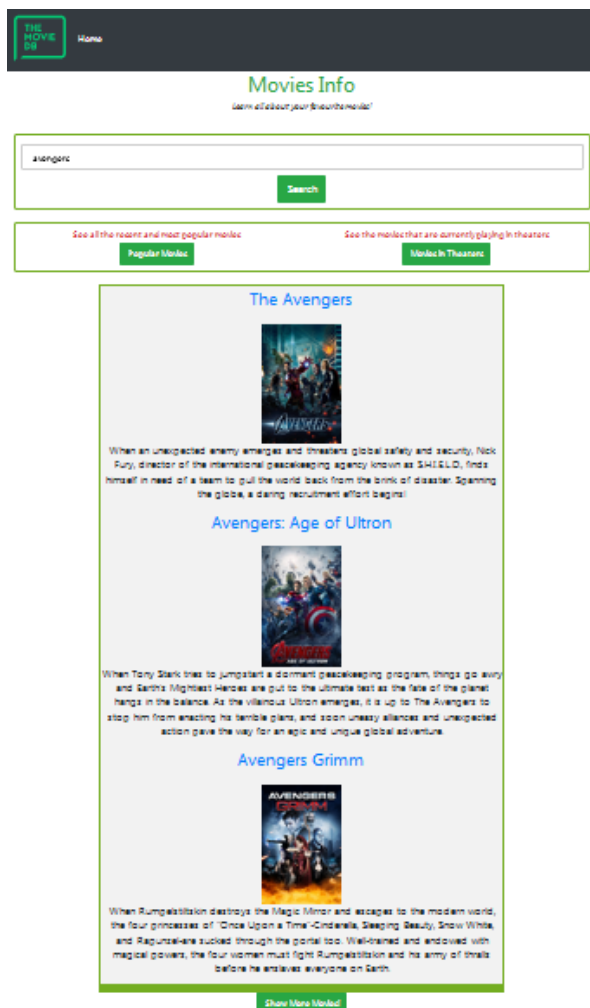
4. Περιγραφή της εφαρμογής

Μεταβαίνοντας ο χρήστης στη σελίδα <https://users.it.teithe.gr/~ait122017/> βλέπει την εικόνα 1.

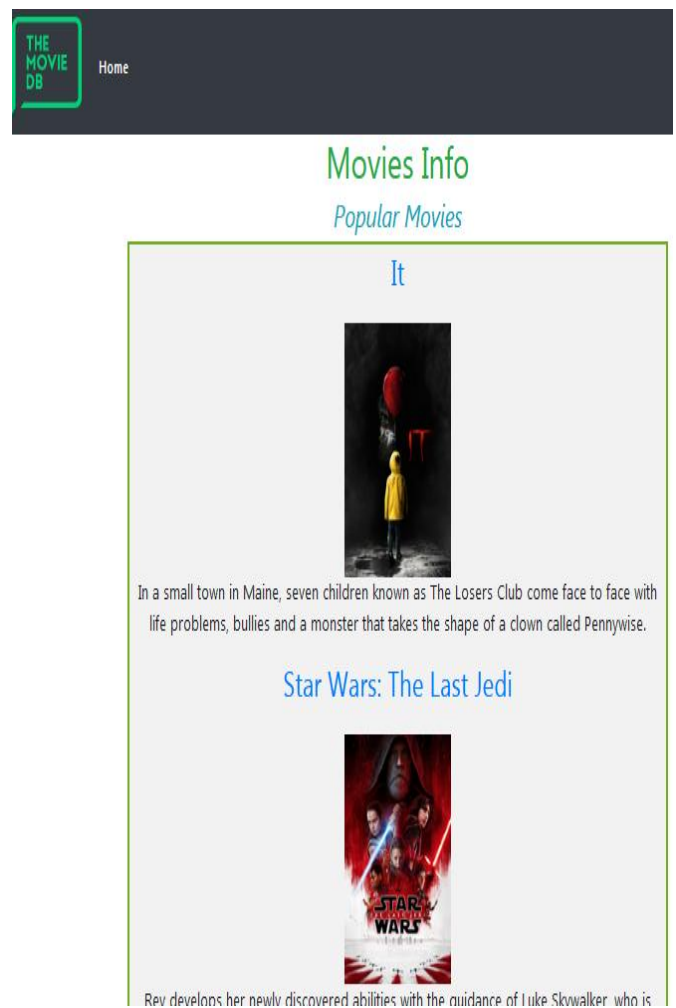


Εικόνα 1

Στο σημείο αυτό έχει τρεις επιλογές: α) στο πλαίσιο αναζήτησης να ορίσει το όνομα μιας ταινίας και κάνοντας είτε κλικ στο κουμπί Search είτε πατήσει το enter να του εμφανιστούν αποτελέσματα για την ταινία αυτή. Αυτά τα αποτελέσματα εμφανίζονται δυναμικά στο κάτω μέρος της σελίδας και αν είναι περισσότερα από τρία, τότε δημιουργείται ένα κουμπί, το **Show More Movies!** το οποίο δίνει την δυνατότητα πατώντας το να εμφανίσει και τα υπόλοιπα. (εικόνα 2) β) να κάνει κλικ στο κουμπί Popular Movies και να εμφανίσει σε ξεχωριστή σελίδα, την index_2.php, τα αποτελέσματα των δημοφιλέστερων ταινιών και γ) να κάνει κλικ στο κουμπί Movies in Theaters και να εμφανίσει σε ξεχωριστή σελίδα, την index_3.php, τα αποτελέσματα των ταινιών που προβάλλονται αυτή τη στιγμή στους κινηματογράφους.




Εικόνα 2



Εικόνα 3


Από τα αποτελέσματα που θα εμφανίσει με έναν από τους παραπάνω τρόπους, ο χρήστης κάνοντας κλικ στον τίτλο της ταινίας θα οδηγηθεί σε μια άλλη σελίδα, την show_movie.php, όπου θα έχει την δυνατότητα να δει περεταίρω πληροφορίες, όπως το

πότε κυκλοφόρησε η ταινία, τι δημοτικότητα έχει, ποια είναι η βαθμολογία της, σε ποιο είδος κατατάσσεται καθώς ποιοι ηθοποιοί παίζουν (εικόνα 4).

 Home

Movies Info

Star Wars: The Last Jedi



Rey develops her newly discovered abilities with the guidance of Luke Skywalker, who is unsettled by the strength of her powers. Meanwhile, the Resistance prepares to do battle with the First Order.

Release Date: 2017-12-13

Popularity: 610.60

Average Vote: 7.2

Genres:
Action, Adventure, Fantasy, Science Fiction

Actors:

Mark Hamill, Carrie Fisher, Adam Driver, Daisy Ridley, John Boyega, Oscar Isaac, Andy Serkis, Lupita Nyong'o, Domhnall Gleeson, Anthony Daniels, Gwendoline Christie, Kelly Marie Tran, Laura Dern, Benicio del Toro, Frank Oz, Joonas Suotamo, Peter Mayhew, Billie Lourd, Mike Quinn, Tim Rose, Amanda Lawrence, Mark Lewis Jones, Adrian Edmondson, Dave Chapman, Brian Herring, Ben Schwartz, Bill Hader, Jimmy Vee, Veronica Ngo, Justin

Εικόνα 4

Σε όλες τις παραπάνω περιπτώσεις, στην γραμμή πλοήγησης στο επάνω μέρος των σελίδων μπορεί να κάνει κλικ στην επιλογή Home και να επιστρέψει στην αρχική σελίδα (index.php) ή να κάνει κλικ στο logo της THE MOVIE DB και να οδηγηθεί στη σελίδα <https://www.themoviedb.org/>

5. The MovieDB API

Το API της Movie DB παρέχεται δωρεάν αρκεί να μην χρησιμοποιηθεί για εμπορική χρήση και απαραίτητη προϋπόθεση είναι να εμφανίζεται το logo της ιστοσελίδας με το σχετικό link που παραπέμπει στο να φαίνεται ότι χρησιμοποιείται 3rd party περιεχόμενο

Movie Info Movie Search Project

από αυτή την ιστοσελίδα. Υπάρχει ένας λογαριασμός στο API με τον οποίο έχουμε ένα κλειδί για χρήση του API. Με το κλειδί αυτό κάνουμε authorize το λογαριασμό μας σε κάθε αίτημά μας προς το API.

Κάποιους περιορισμούς που έχει το API είναι ότι δεν μπορεί να επεξεργαστεί - είναι closed source, περιορίζει τον αριθμό αιτημάτων σε αυτό στα 40 ανά 10 δευτερόλεπτα (που μας καλύπτουν για την εργασία), πρέπει να αναφέρεται στην σελίδα ότι χρησιμοποιούμε το API και ότι δεν έχει πιστοποιηθεί η σελίδα από το TMDb.

Χρησιμοποιούμε το API κάνοντας jQuery ερωτήματα και χρησιμοποιώντας URL διευθύνσεις που μπορούμε να τις βρούμε στο documentation του API (developers.themoviedb.org/3). Το API απαντάει με JSON κώδικα τον οποίο επεξεργαζόμαστε για να εμφανίσουμε το αποτέλεσμα που θέλουμε.

Χρησιμοποιούμε την /search μέθοδο του API η οποία επιστρέφει αποτελέσματα σύμφωνα με το όνομα της ταινίας που θα δώσουμε στο όρισμα: "https://api.themoviedb.org/3/search/movie?api_key=<api key> &query=<τίτλος ταινίας>". Η μέθοδος που χρησιμοποιείται είναι GET και χρησιμοποιείται η ajax με τις κατάλληλες παραμέτρους που γίνεται κλήση στο API και λαμβάνεται ένας πίνακας results με τα αποτελέσματα της αναζήτησης από το response που επιστρέφεται.

Παρακάτω εμφανίζονται οι παράμετροι που μπορούμε να εισαγάγουμε για ένα /search ερώτημα στο API και τα πεδία που μας επιστρέφει σε JSON μορφή:

Query String

api_key	string	default: <<api_key>>	required
language	string	Pass a ISO 639-1 value to display translated data for the fields that support it. minLength: 2 pattern: ([a-z]{2})-([A-Z]{2}) default: en-US	optional
query	string	Pass a text query to search. This value should be URI encoded. minLength: 1	required

Movie Info

Movie Search Project

page	integer	Specify which page to query. minimum: 1 maximum: 1000 default: 1	optional
include_adult	boolean	Choose whether to include adult (pornography) content in the results. default	optional
region	string	Specify a ISO 3166-1 code to filter release dates. Must be uppercase. pattern: <code>^[A-Z]{2}\$</code>	optional
year	integer		optional
primary_release_year	integer		optional

JSON Response

Object		
Page	integer	optional
total_results	integer	optional
total_pages	integer	optional
results	array[object]	{Movie List Result Object}

Results Array

poster_path	string or null	optional
adult	boolean	optional
overview	string	optional
release_date	string	optional
genre_ids	array[integer]	optional
id	integer	optional
original_title	string	optional
original_language	string	optional
title	string	optional
backdrop_path	string or null	optional
popularity	number	optional
vote_count	integer	optional
video	boolean	optional
vote_average	number	optional

6. Χρονοδιάγραμμα υλοποίησης της εφαρμογής

- Περίοδος προγραμματισμού: 1/11/17 – 9/1/18
- Προγραμματισμός βασικού κώδικα λειτουργίας: 1/11/17 – 31/12/17
- Προσθήκη επιπλέον λειτουργιών: 31/12/17 – 7/1/18
- Βελτίωση γραφικών: 7/1/18 – 10/1/18
- Συγγραφή τεχνικής αναφοράς: 7/1/18 – 12/1/18
- Συλλογή ερωτηματολογίων – ανάλυση: 15/12/17 – 12/1/18

7. Δυσκολίες οργάνωσης και προγραμματισμού

- Απαιτούνταν εξοικείωση με τον κώδικα προγραμματισμού και με νέες τεχνολογίες όπως η ajax, jquery, json, api εφαρμογής.
- Εξοικείωση με το phpstorm και github για συγγραφή κώδικα και σύνδεση με github για διαμοιρασμό.
- Έλλειψη εμπειρίας στον κώδικα με αποτέλεσμα περισσότερος χρόνος για εκμάθηση.
- Αποστάσεις μεταξύ μελών με αποτέλεσμα την από απόσταση επικοινωνία (πολλές φορές με τεχνικά προβλήματα).

8. Αξιολόγηση της εφαρμογής

Η αξιολόγηση της εφαρμογής πραγματοποιήθηκε με τη χρήση κάποιων Tasks και στη συνέχεια Web Usability Test με χρήση ερωτηματολογίου. Τα αποτελέσματα βρίσκονται στο αρχείο Usability Testing – Αναφορά.

9. Πηγές

https://www.w3schools.com/html/html_examples.asp

<https://www.w3schools.com/>

<https://stackoverflow.com/>

Movie Info

Movie Search Project

<https://developers.themoviedb.org/3/getting-started/introduction>

<https://www.themoviedb.org/talk>

<https://www.w3schools.com/bootstrap4/default.asp>

<https://www.w3schools.com/css/default.asp>

<http://www.tutor.edu.gr/index.php/ajax-examples/ajax-examples-3>

<http://www.tutor.edu.gr/index.php/jquery/jquery-ajax>

https://www.w3schools.com/js/js_functions.asp

https://www.w3schools.com/js/js_loop_for.asp

https://www.w3schools.com/php/php_forms.asp

https://www.w3schools.com/jquery/jquery_ajax_get_post.asp