

## Περιγραφή Υλοποίησης και Καταγραφή Απαιτήσεων

Η παρούσα εργασία αποτελεί μία προσπάθεια δημιουργίας μιας διαδικτυακής εφαρμογής και πιο συγκεκριμένα μίας μηχανής αναζήτησης βίντεο, η οποία θα χρησιμοποιεί το YouTube API.

Στόχος της είναι να παρέχει στους χρήστες τη δυνατότητα να αναζητήσουν, και τελικά να βρίσκουν, γρήγορα και αποτελεσματικά βίντεο της επιλογής τους.

Για τη σχεδίαση και την ανάπτυξη της εφαρμογής αυτής έχουν ληφθεί υπόψη οι συνηθέστερες ανάγκες και απαιτήσεις των χρηστών, όπως είναι:

- Η ευκολία στη χρήση.
- Η γρήγορη αναζήτηση.
- Ο σύντομος χρόνος απόκρισης της εφαρμογής.
- Η ακεραιότητα στα αποτελέσματα.
- Η αδιάλειπτη λειτουργία της εφαρμογής.
- Η δυνατότητα αναζήτησης βάσει των διαφορετικών χαρακτηριστικών των βίντεο, π.χ. τη διάρκειά τους.
- Η ταξινόμηση και η εμφάνιση των αποτελεσμάτων με βάση την επιθυμία των χρηστών, π.χ. με βάση τη σχετικότητα, την ημερομηνία, την βαθμολογία, το πλήθος προβολών.
- Η απλή και εύκολη πλοήγηση.
- Η σύγχρονη και αισθητικά ευχάριστη εμφάνιση της εφαρμογής.
- Η προαιρετική είσοδος στο σύστημα για την χρήση των υπηρεσιών.
- Η δυνατότητα εισόδου στο σύστημα με διάφορους τρόπους, όπως π.χ. μέσω της δημιουργίας νέου λογαριασμού ή με τη χρήση κάποιου ήδη υπάρχοντος λογαριασμού (π.χ. Facebook ή Google).
- Η προστασία των προσωπικών στοιχείων των χρηστών.
- Η δυνατότητα να υλοποιούνται όλες οι λειτουργίες στο ίδιο παράθυρο, ώστε να μην υπάρχουν ανακατευθύνσεις σε άλλες καρτέλες ή σε μη έμπιστους εξωτερικούς συνδέσμους.

- Η ικανοποιητική υποστήριξη και η μεγάλη αποκρισιμότητα της εφαρμογής σε όλα τα μεγέθη οθονών και τους διάφορους browsers.

Επιπλέον, πέρα από όλα όσα αφορούν τους χρήστες, ορισμένες εξίσου σημαντικές απαιτήσεις που έχουν ληφθεί υπόψη για την αποτελεσματικότητα της εφαρμογής είναι οι εξής:

- Η εφαρμογή να λειτουργεί και να συμπεριφέρεται σύμφωνα με τις καταγεγραμμένες απαιτήσεις χρηστών.
- Εύκολη τροποποίηση της εφαρμογής.
- Δυνατότητα επαναχρησιμοποίησης και επεκτασιμότητας.

## API

Για την ανάπτυξη της εφαρμογής και την υλοποίηση της διαδικασίας της αναζήτησης χρησιμοποιήθηκε το YouTube API. Συγκεκριμένα, αφού, μέσω των συναρτήσεων `search()` και `getRadioVal()`, πραγματοποιηθεί η λήψη της τιμής αναζήτησης και των παραμέτρων αναζήτησης του χρήστη και προκειμένου να διεκπεραιωθεί η αναζήτηση και η επιστροφή των κατάλληλων αποτελεσμάτων σε μορφή JSON, χρησιμοποιούνται κατάλληλα AJAX Calls προς το API.

Έπειτα από την λήψη των αποτελεσμάτων σε μορφή JSON, χρησιμοποιούνται κατάλληλες συναρτήσεις για την τροποποίηση και για την δυναμική απεικόνισή τους, όπως π.χ. `addDotSeparator(number)`, `getPublishedDate(publishedAt)`, `convertISO8601ToSeconds(input)` και `getOutput(item, index)`.

Για την υλοποίηση της σελιδοποίησης (pagination) χρησιμοποιούνται, επίσης, κατάλληλα AJAX Calls προς το API με τη χρήση της παραμέτρου `pageToken` και της συνάρτησης `getButtons(prevPageToken, nextPageToken)`.

Επιπλέον, οι χρήστες έχουν την δυνατότητα να επιλέξουν:

- το πλήθος των αποτελεσμάτων που επιθυμούν να βλέπουν ανά σελίδα (`maxResults: 5, 10, 25`),

- διαφορετικά κριτήρια αναζήτησης (videoDuration: Any, Short, Medium, Long) και
- διαφορετικούς τρόπους ταξινόμησης (order: Relevance, Date, Rating, Views).

Στα API αιτήματα χρησιμοποιούνται κατάλληλες παράμετροι, όπως:

Παράμετροι	Παράδειγμα χρήσης	Περιγραφή
<b>id</b>	videoId	Για τον προσδιορισμό του id του βίντεο
<b>q</b>	query	Για τον προσδιορισμό της αναζήτησης του χρήστη.
<b>type</b>	'video'	Για τον προσδιορισμό του είδους του αντικειμένου αναζήτησης.
<b>key</b>	'AlzaSyBAN-71jVHKbUzBluoQS_OVMb9mLctpEUU'	Δημόσιο κλειδί για την κλήση του API.
<b>part</b>	'contentDetails,statistics,snippet'	Για τον προσδιορισμό συγκεκριμένων τμημάτων του JSON.
<b>fields</b>	'items(contentDetails(duration),statistics,snippet(description))'	Για τον προσδιορισμό συγκεκριμένων πεδίων των τμημάτων του JSON.
<b>order</b>	Relevance, Date, Rating, Views	Για την υλοποίηση της ταξινόμησης και τον προσδιορισμό της σειράς εμφάνισης των αποτελεσμάτων.
<b>videoDuration</b>	Any, Short, Medium, Long	Για τον προσδιορισμό της διάρκειας των προς αναζήτηση βίντεο.
<b>maxResults</b>	5, 10, 15	Για τον προσδιορισμό του πλήθους των αποτελεσμάτων ανά σελίδα.
<b>pageToken</b>	token	Για την υλοποίηση της σελιδοποίησης (pagination)

## FancyBox

Πηγή: <http://fancybox.net/>

Προκειμένου η αναπαραγωγή των βίντεο να υλοποιείται στην ίδια σελίδα και να μην υπάρχουν ανακατευθύνσεις σε άλλες καρτέλες ή σε εξωτερικούς συνδέσμους, χρησιμοποιήθηκε το FancyBox ως ένας embedded video player. Το FancyBox είναι ένα εργαλείο, το οποίο δημιουργήθηκε με τη χρήση της βιβλιοθήκης JQuery και χρησιμοποιείται για την προβολή εικόνων, περιεχομένου Html και πολυμέσων με ένα "Mac-style lightbox", το οποίο "επιπλέει (floats)" πάνω από μία σελίδα.

## Παράδειγμα εφαρμογής

### (searchPage.html - Αρχικοποίηση)

```
<!--fancybox initialize-->

<script type="text/javascript" src="scripts/jquery.fancybox.js"></script>

<script type="text/javascript">

    $(document).ready(function(){

        $('[data-fancybox]').fancybox();

    });

</script>
```

### (apiScripts.js - Εφαρμογή)

```
'<a data-fancybox data-type="iframe"
href="http://www.youtube.com/embed/'+item.id.videoid+'"><img src="" +
item.snippet.thumbnails.high.url + "'></a>' +
```

## Login

Για την χρήση της εφαρμογής δεν απαιτείται είσοδος, οι χρήστες μπορούν απλά να ανοίξουν το URL στον browser τους και να αναζητήσουν το video που επιθυμούν.

Παρ' όλα αυτά, παρέχεται παράλληλα και η δυνατότητα δημιουργίας λογαριασμού και εισόδου με τη χρήση μοναδικού ονόματος χρήστη και κωδικού.

Σε περίπτωση που ο χρήστης επιθυμήσει να συνδεθεί με το λογαριασμό του στην εφαρμογή, έχει τις παρακάτω απαιτήσεις:

- **Ασφάλεια:** να γνωρίζει δηλαδή ο χρήστης ότι ο κωδικός του δεν είναι ορατός σε τρίτους.
- **Αξιοπιστία:** να είναι σίγουρος ότι αν εισάγει τα διαπιστευτήριά του σωστά θα εισαχθεί στο σύστημα, ενώ θα λάβει μήνυμα σφάλματος σε αντίθετη περίπτωση.
- **Αναγνώριση:** να ξέρει ότι η εφαρμογή τον διακρίνει από τα διαπιστευτήρια του.

Για τους παραπάνω λόγους, γίνεται προσπάθεια να χρησιμοποιηθεί κάποιο δωρεάν πιστοποιητικό SSL, ώστε να μπορέσει να ανέβει η εφαρμογή σε HTTPS. Επίσης, οι κωδικοί αποθηκεύονται κατακερματισμένοι και όχι σε μορφή απλού κειμένου. Τέλος, προφανώς τα ονόματα εισόδου των χρηστών είναι μοναδικά.

Η δημιουργία λογαριασμού δίνει τη δυνατότητα στην εφαρμογή να κρατά τις προτιμήσεις των χρηστών και ίσως και κάποιο ιστορικό προβολών.

Για την υλοποίηση χρησιμοποιείται βάση δεδομένων MySQL.

Τέλος, εναλλακτικές μέθοδοι εισόδου, π.χ. το Facebook Login, μελετώνται για να μπορεί ο χρήστης να συνδεθεί χωρίς τη δημιουργία λογαριασμού, αλλά με τη χρήση κάποιου ήδη υπάρχοντος λογαριασμού, όπως Facebook ή Google.

## **Bootstrap**

Για την σύγχρονη εμφάνιση της εφαρμογής προτιμήθηκε να χρησιμοποιηθεί η βιβλιοθήκη Bootstrap, η οποία με τη χρήση έτοιμων κλάσεων CSS επιτρέπει να καθορίσουμε την εμφάνιση αλλά και κάποια λειτουργικότητα των στοιχείων μας. Επίσης, με το Bootstrap επιτυγχάνουμε τη μέγιστη αποκρισιμότητα σε όλα τα μεγέθη browsers, από κινητά, tablets έως και μεγάλες οθόνες υπολογιστών desktop.