Πτυχιακή Εργασία: Σχεδιασμός και Υλοποίηση της Πλατφόρμας Watchable

[Όνομα Φοιτητή]

[Ημερομηνία]

Table of Contents

# 1 Περίληψη

Η παρούσα πτυχιακή εργασία πραγματεύεται τον σχεδιασμό και την υλοποίηση της πλατφόρμας Watchable, μιας διαδικτυακής εφαρμογής που στοχεύει στη βελτίωση της εμπειρίας παρακολούθησης και αξιολόγησης οπτικοακουστικού περιεχομένου. Αρχικά, παρουσιάζεται το θεωρητικό υπόβαθρο και οι σύγχρονες ανάγκες των χρηστών για προσωποποιημένες προτάσεις και κοινωνική αλληλεπίδραση γύρω από ταινίες και σειρές. Στη συνέχεια, αναλύονται οι απαιτήσεις του συστήματος και περιγράφεται ο σχεδιασμός της αρχιτεκτονικής, με έμφαση στη φιλικότητα προς τον χρήστη, την επεκτασιμότητα και την ασφάλεια των δεδομένων.

Ακολουθεί αναλυτική περιγραφή της υλοποίησης της πλατφόρμας, όπου παρουσιάζονται τα βασικά χαρακτηριστικά, όπως η αναζήτηση και αξιολόγηση περιεχομένου, η δημιουργία λιστών παρακολούθησης και οι δυνατότητες κοινωνικής δικτύωσης μεταξύ των χρηστών. Έπεται η ενότητα ελέγχου και αξιολόγησης της εφαρμογής, όπου μέσω δοκιμών και ανατροφοδότησης χρηστών εξετάζεται η αποτελεσματικότητα και η χρηστικότητά της. Η εργασία ολοκληρώνεται με τη διατύπωση συμπερασμάτων και προτάσεων για μελλοντικές επεκτάσεις, καταδεικνύοντας ότι το Watchable ανταποκρίνεται επιτυχώς στις ανάγκες των χρηστών και θέτει τις βάσεις για περαιτέρω εξέλιξη στον τομέα αυτό.

# 2 Abstract

This thesis addresses the design and implementation of Watchable, a web platform aimed at enhancing the experience of watching and evaluating audiovisual content. It begins with the theoretical background and the modern user needs for personalized recommendations and social interaction around movies and series. Next, the system requirements are analyzed and the architectural design is described, with emphasis on user-friendliness, scalability, and data security.

A detailed description of the platform’s implementation follows, presenting key features such as content search and rating, creation of watchlists, and social networking capabilities among users. The subsequent section focuses on the evaluation and testing of the application, where user feedback and trials assess its effectiveness and usability. The thesis concludes with a summary of findings and suggestions for future improvements, demonstrating that Watchable successfully meets user needs and lays the groundwork for further development in this field.

# 3 Πίνακας Περιεχομένων

1. Εισαγωγή 1.1 Σκοπός και στόχοι της εργασίας 1.2 Κίνητρα και σημασία του θέματος 1.3 Δομή του εγγράφου
2. Θεωρητικό υπόβαθρο και σχετική έρευνα 2.1 Υφιστάμενες λύσεις και ανταγωνιστικές πλατφόρμες 2.1.1 IMDb (Internet Movie Database) 2.1.2 TMDB (The Movie Database) 2.1.3 Letterboxd 2.1.4 GoodWatch 2.1.4.1 Συγκριτικός πίνακας βασικών χαρακτηριστικών ανταγωνιστικών πλατφορμών 2.2 Τεχνολογίες διαδικτύου και μοντέρνες εφαρμογές 2.2.1 Βασικές τεχνολογίες διαδικτύου 2.2.2 Σύγχρονες αρχές και πρακτικές ανάπτυξης 2.3 Το TMDB API και τα δεδομένα πολυμέσων 2.3.1 Βασικά χαρακτηριστικά και δεδομένα του TMDB API 2.3.2 Η σημασία του TMDB API για το Watchable
3. Ανάλυση απαιτήσεων και σχεδιασμός συστήματος 3.1 Ανάλυση απαιτήσεων χρηστών 3.1.1 Εστίαση στον χρήστη: σχεδιασμός και λειτουργικότητα 3.1.2 Ανάγκες και προσδοκίες 3.1.3 Περιορισμοί και προκλήσεις 3.1.4 Συμπεράσματα ανάλυσης 3.2 Λειτουργικές και μη λειτουργικές απαιτήσεις 3.2.1 Λειτουργικές απαιτήσεις 3.2.2 Μη λειτουργικές απαιτήσεις 3.2.3 Συνοπτική απεικόνιση απαιτήσεων 3.3 Αρχιτεκτονική συστήματος 3.3.1 Κύρια δομικά στοιχεία 3.3.2 Ροή δεδομένων και αλληλεπίδραση 3.3.3 Αρχιτεκτονικές αρχές και βέλτιστες πρακτικές 3.3.4 Ενδεικτικό διάγραμμα αρχιτεκτονικής
4. Υλοποίηση της εφαρμογής Watchable 4.1 Τεχνολογίες που χρησιμοποιήθηκαν 4.1.1 Angular 4.1.2 RxJS 4.1.3 SCSS (Sassy CSS) 4.1.4 placehold.co 4.1.5 FontAwesome 4.1.6 Swiper 4.1.7 TMDB API 4.1.8 TypeScript 4.1.9 Ποιότητα Κώδικα: Linters & Prettier 4.1.10 Εργαλεία Ανάπτυξης & Διαχείρισης 4.2 Δομή και οργάνωση του κώδικα 4.3 Διασύνδεση με το TMDB API 4.3.1 Αρχιτεκτονική διασύνδεσης (drivers, gateways, facades) 4.3.2 Επικοινωνία και ασφάλεια 4.3.3 Διαδικασία login και register μέσω TMDB 4.3.4 Οφέλη και προκλήσεις 4.4 Σχεδίαση και υλοποίηση διεπαφής χρήστη 4.4.1 Αρχές σχεδίασης και στόχοι 4.4.2 Χρωματική παλέτα, τυπογραφία και εικονίδια 4.4.3 Custom SCSS, animations και skeleton loaders 4.4.4 Δομή βασικών σελίδων και επαναχρησιμοποιήσιμα components 4.4.5 Προσβασιμότητα και τεχνικές βελτιστοποίησης 4.5 Ενδεικτικά σενάρια χρήσης 4.5.1 Σενάριο 1: Αναζήτηση και εξερεύνηση ταινιών 4.5.2 Σενάριο 2: Προσθήκη ταινίας στη watchlist 4.5.3 Σενάριο 3: Αξιολόγηση ταινίας και διαχείριση βαθμολογιών 4.5.4 Σενάριο 4: Πλοήγηση σε κατηγορίες και εξερεύνηση περιεχομένου 4.5.5 Σενάριο 5: Σύνδεση και αυθεντικοποίηση μέσω TMDB
5. Έλεγχος, αξιολόγηση και βελτιστοποίηση 5.1 Μέθοδοι ελέγχου και δοκιμών 5.1.1 Έλεγχος λειτουργικότητας 5.1.2 Έλεγχος συμβατότητας και προσβασιμότητας 5.1.3 Έλεγχος απόδοσης και βελτιστοποίησης 5.1.4 Έλεγχος ποιότητας κώδικα 5.1.5 Εσωτερική αξιολόγηση και δοκιμές από χρήστες 5.1.6 Περιορισμοί και μελλοντικές βελτιώσεις 5.2 Προτάσεις για μελλοντικές βελτιώσεις 5.2.1 Ενίσχυση αυτοματοποίησης ελέγχων και ποιότητας 5.2.2 Βελτιστοποίηση απόδοσης και τεχνικής υποδομής 5.2.3 Εμπλουτισμός λειτουργιών και εμπειρίας χρήστη 5.2.4 Επέκταση διασύνδεσης και υποδομής 5.2.5 Τεκμηρίωση και κοινότητα
6. Αποτελέσματα και παρατηρήσεις 6.1 Αποτελέσματα ως προς τη λειτουργικότητα 6.2 Αποτελέσματα ως προς την απόδοση 6.3 Αποτελέσματα ως προς τη συμβατότητα και προσβασιμότητα 6.4 Αποτελέσματα ως προς την εμπειρία χρήστη (UX) 6.5 Αποτελέσματα ως προς την ποιότητα κώδικα 6.6 Συνοπτική αποτίμηση
7. Συμπεράσματα 7.1 Ανακεφαλαίωση βασικών σημείων 7.2 Προοπτικές αποτίμησης και εμπειρίες 7.2.1 Εμβάθυνση σε σύγχρονες τεχνολογίες και μεθοδολογίες 7.2.2 Προκλήσεις και τρόποι αντιμετώπισης 7.2.3 Απόκτηση δεξιοτήτων και προσωπική εξέλιξη 7.2.4 Συμπεράσματα και μελλοντικές προοπτικές
8. Βιβλιογραφία 8.1 Εισαγωγή 8.2 Βιβλία & Επιστημονικά Άρθρα 8.3 Διαδικτυακές Πηγές & Τεκμηρίωση 8.4 Παρατηρήσεις για τη Βιβλιογραφία
9. Παραρτήματα 9.1 Οδηγίες εγκατάστασης και χρήσης 9.1.1 Προαπαιτούμενα 9.1.2 Λήψη του πηγαίου κώδικα 9.1.3 Εγκατάσταση εξαρτήσεων 9.1.4 Εκκίνηση της εφαρμογής σε περιβάλλον ανάπτυξης 9.1.5 Βασικές οδηγίες χρήσης 9.1.6 Συχνά προβλήματα και λύσεις 9.1.7 Κλείσιμο 9.2 Πρόσθετα σχήματα, screenshots, διαγράμματα 9.2.1 Εισαγωγή 9.2.2 Διάγραμμα αρχιτεκτονικής εφαρμογής 9.2.3 Screenshot αρχικής σελίδας εφαρμογής 9.2.4 Διάγραμμα δομής φακέλων και αρχείων 9.2.5 Screenshot λειτουργίας αναζήτησης 9.2.6 Screenshot watchlist και αξιολόγησης 9.2.7 Screenshot responsive σχεδίασης 9.2.8 Κλείσιμο

# 4 1. Εισαγωγή

## 4.1 1.1 Σκοπός και στόχοι της εργασίας

Η παρούσα πτυχιακή εργασία έχει ως βασικό σκοπό τη μελέτη, σχεδίαση και υλοποίηση μιας σύγχρονης διαδικτυακής εφαρμογής για την αναζήτηση, οργάνωση και αξιολόγηση πολυμεσικού περιεχομένου, με έμφαση στις ταινίες και τις τηλεοπτικές σειρές. Η εφαρμογή, με την ονομασία «Watchable», στοχεύει να προσφέρει στους χρήστες ένα φιλικό και λειτουργικό περιβάλλον, όπου θα μπορούν να ανακαλύπτουν νέους τίτλους, να διαχειρίζονται τις προσωπικές τους λίστες και να μοιράζονται τις προτιμήσεις τους με άλλους.

Οι κύριοι στόχοι της εργασίας συνοψίζονται ως εξής:

* Η διερεύνηση των αναγκών και των συνηθειών των χρηστών σχετικά με την παρακολούθηση πολυμέσων.
* Η ανάλυση των υφιστάμενων λύσεων και η ανάδειξη των πλεονεκτημάτων και αδυναμιών τους.
* Η επιλογή κατάλληλων τεχνολογιών για την ανάπτυξη μιας μοντέρνας και επεκτάσιμης εφαρμογής.
* Η υλοποίηση βασικών λειτουργιών, όπως η αναζήτηση και προβολή πληροφοριών για ταινίες/σειρές, η δημιουργία και διαχείριση λιστών, καθώς και η διασύνδεση με εξωτερικές βάσεις δεδομένων (όπως το TMDB API).

## 4.2 1.2 Κίνητρα και σημασία του θέματος

Η ραγδαία ανάπτυξη του διαδικτύου και των ψηφιακών τεχνολογιών έχει αλλάξει ριζικά τον τρόπο με τον οποίο οι άνθρωποι αναζητούν, παρακολουθούν και διαχειρίζονται το πολυμεσικό περιεχόμενο. Η πληθώρα διαθέσιμων ταινιών και σειρών, σε συνδυασμό με την ύπαρξη πολλών διαφορετικών πλατφορμών, δημιουργεί την ανάγκη για εργαλεία που βοηθούν τους χρήστες να οργανώνουν τις επιλογές τους και να ανακαλύπτουν νέο περιεχόμενο που ανταποκρίνεται στα ενδιαφέροντά τους.

Το κίνητρο για την υλοποίηση της παρούσας εργασίας προέκυψε από την παρατήρηση ότι, παρά την ύπαρξη αρκετών σχετικών εφαρμογών, συχνά απουσιάζει η ευκολία χρήσης, η εξατομίκευση και η δυνατότητα διασύνδεσης με αξιόπιστες βάσεις δεδομένων, όπως το TMDB. Επιπλέον, η ανάγκη για μια εφαρμογή που να συνδυάζει μοντέρνο σχεδιασμό, λειτουργικότητα και επεκτασιμότητα, αποτέλεσε βασικό παράγοντα για την επιλογή του θέματος.

Η σημασία της εργασίας έγκειται στο γεγονός ότι ανταποκρίνεται σε πραγματικές ανάγκες των χρηστών, προσφέροντας μια ολοκληρωμένη λύση για την οργάνωση και αξιολόγηση πολυμέσικου περιεχομένου. Παράλληλα, συμβάλλει στην εξοικείωση με σύγχρονες τεχνολογίες ανάπτυξης λογισμικού και στην κατανόηση των προκλήσεων που σχετίζονται με τη διαχείριση μεγάλου όγκου δεδομένων και τη βελτιστοποίηση της εμπειρίας χρήστη.

## 4.3 1.3 Δομή του εγγράφου

Το παρόν έγγραφο είναι δομημένο με τρόπο που να διευκολύνει την κατανόηση της πορείας υλοποίησης της πτυχιακής εργασίας, από τη θεωρητική θεμελίωση έως την τελική αξιολόγηση και τα συμπεράσματα. Κάθε κεφάλαιο επικεντρώνεται σε συγκεκριμένες πτυχές του θέματος, ακολουθώντας μια λογική και εξελικτική σειρά.

Συγκεκριμένα, το πρώτο κεφάλαιο περιλαμβάνει την εισαγωγή, όπου παρουσιάζονται ο σκοπός, τα κίνητρα και η σημασία της εργασίας, καθώς και η συνοπτική περιγραφή της δομής του εγγράφου. Στο δεύτερο κεφάλαιο αναλύεται το θεωρητικό υπόβαθρο και γίνεται αναφορά στη σχετική έρευνα και στις υφιστάμενες λύσεις. Το τρίτο κεφάλαιο εστιάζει στην ανάλυση των απαιτήσεων και στον σχεδιασμό του συστήματος, ενώ το τέταρτο περιγράφει αναλυτικά τη διαδικασία υλοποίησης της εφαρμογής Watchable.

Στη συνέχεια, το πέμπτο κεφάλαιο αφορά τον έλεγχο, την αξιολόγηση και τις προτάσεις για μελλοντικές βελτιώσεις. Το έκτο κεφάλαιο περιλαμβάνει τα συμπεράσματα και την προσωπική αποτίμηση, ενώ το έβδομο και όγδοο κεφάλαιο περιέχουν τη βιβλιογραφία και τα παραρτήματα αντίστοιχα, με πρόσθετο υλικό και οδηγίες χρήσης.

Η δομή αυτή διασφαλίζει τη συστηματική παρουσίαση όλων των σταδίων της εργασίας και επιτρέπει στον αναγνώστη να παρακολουθήσει με σαφήνεια την εξέλιξη και τα αποτελέσματα του έργου.

# 5 2. Θεωρητικό υπόβαθρο και σχετική έρευνα

## 5.1 2.1 Υφιστάμενες λύσεις και ανταγωνιστικές πλατφόρμες

Η συνεχής αύξηση του πολυμεσικού περιεχομένου και η ποικιλία των διαθέσιμων πλατφορμών έχουν οδηγήσει στη δημιουργία πολλών διαδικτυακών λύσεων που βοηθούν τους χρήστες να αναζητούν, να οργανώνουν και να αξιολογούν ταινίες και τηλεοπτικές σειρές. Στην ενότητα αυτή παρουσιάζονται οι σημαντικότερες ανταγωνιστικές πλατφόρμες, τα βασικά χαρακτηριστικά τους και η θέση τους στην αγορά, με στόχο την ανάδειξη των πλεονεκτημάτων και των αδυναμιών τους σε σχέση με τη φιλοσοφία της εφαρμογής Watchable.

### 5.1.1 2.1.1 IMDb (Internet Movie Database)

Το IMDb αποτελεί μία από τις πιο γνωστές και διαδεδομένες πλατφόρμες παγκοσμίως για την αναζήτηση πληροφοριών σχετικά με ταινίες, σειρές, ηθοποιούς και συντελεστές. Παρέχει εκτενείς βάσεις δεδομένων με βαθμολογίες, κριτικές, trailers, λίστες και στατιστικά, ενώ επιτρέπει στους χρήστες να δημιουργούν προσωπικές watchlists και να συμμετέχουν σε κοινότητες. Η ευρεία αποδοχή του IMDb οφείλεται κυρίως στην πληρότητα των δεδομένων και στη διασύνδεση με τη βιομηχανία του κινηματογράφου, ωστόσο η διεπαφή του συχνά θεωρείται περίπλοκη και λιγότερο φιλική για τον μέσο χρήστη (IMDb, 2025).

### 5.1.2 2.1.2 TMDB (The Movie Database)

Η πλατφόρμα TMDB προσφέρει επίσης μια πλούσια βάση δεδομένων για ταινίες και σειρές, με έμφαση στη διασύνδεση με εφαρμογές τρίτων μέσω του ανοιχτού API της. Οι χρήστες μπορούν να δημιουργούν λογαριασμό, να διατηρούν προσωπικές λίστες, να βαθμολογούν και να σχολιάζουν περιεχόμενο. Το TMDB ξεχωρίζει για την κοινότητα που το υποστηρίζει, τη δυνατότητα συνεισφοράς νέων δεδομένων και την ευκολία ενσωμάτωσης σε άλλες εφαρμογές, όπως το Watchable. Ωστόσο, ο σχεδιασμός της διεπαφής του είναι αρκετά απλός και λιτός, γεγονός που μπορεί να θεωρηθεί αρνητικό για χρήστες που αναζητούν πιο εξελιγμένη ή ελκυστική εμπειρία πλοήγησης (TMDB, 2025).

### 5.1.3 2.1.3 Letterboxd

Το Letterboxd αποτελεί μια κοινωνική πλατφόρμα αφιερωμένη κυρίως στους λάτρεις του κινηματογράφου. Επιτρέπει στους χρήστες να καταγράφουν τις ταινίες που έχουν παρακολουθήσει, να δημιουργούν λίστες, να γράφουν κριτικές και να αλληλεπιδρούν με άλλους μέσω σχολίων και ακολουθιών. Η έμφαση στη διαμοίραση εμπειριών και στη δημιουργία κοινοτήτων καθιστά το Letterboxd ιδιαίτερα δημοφιλές σε νεότερες ηλικίες και σε όσους αναζητούν κοινωνική διάσταση στην παρακολούθηση ταινιών. Ωστόσο, η εστίαση στις σειρές είναι περιορισμένη και ορισμένες λειτουργίες διατίθενται μόνο σε συνδρομητές (Letterboxd, 2025).

### 5.1.4 2.1.4 GoodWatch

Το GoodWatch είναι μια νεότερη πλατφόρμα που συνδυάζει χαρακτηριστικά αναζήτησης, αξιολόγησης και προσωποποιημένων προτάσεων για ταινίες και σειρές. Ξεχωρίζει για τον αλγόριθμο συστάσεων και τη μοντέρνα διεπαφή, ενώ δίνει έμφαση στην ανάλυση των προτιμήσεων του χρήστη. Παρότι δεν διαθέτει ακόμη το εύρος δεδομένων των μεγαλύτερων πλατφορμών, αναπτύσσεται δυναμικά και προσελκύει χρήστες που αναζητούν εξατομικευμένη εμπειρία (GoodWatch, 2025).

#### 5.1.4.1 2.1.4.1 Συγκριτικός πίνακας βασικών χαρακτηριστικών ανταγωνιστικών πλατφορμών

| Πλατφόρμα | Βάση Δεδομένων | Προσωπικές Λίστες | Βαθμολογίες/Κριτικές | Κοινωνικά Χαρακτηριστικά | API/Ενσωμάτωση | Εστίαση σε Σειρές | Σχεδιασμός Διεπαφής |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **IMDb** | Πολύ μεγάλη | Ναι | Ναι | Μέτρια | Όχι | Ναι | Πλούσιος, αλλά περίπλοκος |
| **TMDB** | Μεγάλη | Ναι | Ναι | Μέτρια | Ναι | Ναι | Απλός και λιτός |
| **Letterboxd** | Μεσαία | Ναι | Ναι | Έντονα κοινωνικά | Όχι | Όχι (κυρίως ταινίες) | Μοντέρνος, ελκυστικός |
| **GoodWatch** | Μικρότερη | Ναι | Ναι | Περιορισμένα | Όχι | Ναι | Μοντέρνος, φιλικός |

Συνοψίζοντας, οι υφιστάμενες λύσεις προσφέρουν πλούσιες δυνατότητες αναζήτησης, οργάνωσης και αξιολόγησης πολυμεσικού περιεχομένου, με κάθε πλατφόρμα να δίνει έμφαση σε διαφορετικές πτυχές της εμπειρίας χρήστη. Η ανάπτυξη της εφαρμογής Watchable βασίζεται στη μελέτη αυτών των λύσεων, με στόχο να συνδυάσει τα πλεονεκτήματά τους και να προσφέρει μια πιο φιλική, εξατομικευμένη και επεκτάσιμη προσέγγιση για τον σύγχρονο χρήστη.

## 5.2 2.2 Τεχνολογίες διαδικτύου και μοντέρνες εφαρμογές

Η εξέλιξη του διαδικτύου και των σχετικών τεχνολογιών έχει διαμορφώσει ένα νέο τοπίο στην ανάπτυξη εφαρμογών, προσφέροντας εργαλεία και πρότυπα που επιτρέπουν τη δημιουργία σύγχρονων, δυναμικών και φιλικών προς τον χρήστη λύσεων. Στην ενότητα αυτή παρουσιάζονται οι βασικές τεχνολογίες και οι αρχές που χαρακτηρίζουν τις μοντέρνες διαδικτυακές εφαρμογές, με έμφαση σε εκείνες που αξιοποιούνται στην υλοποίηση της Watchable.

### 5.2.1 2.2.1 Βασικές τεχνολογίες διαδικτύου

* **HTML5**: Η σύγχρονη έκδοση της βασικής γλώσσας δομής του διαδικτύου, που επιτρέπει τη δημιουργία πλούσιων, διαδραστικών και πολυμεσικών ιστοσελίδων, με ενσωματωμένη υποστήριξη για βίντεο, ήχο και προηγμένα στοιχεία διάταξης.
* **CSS3**: Η τεχνολογία μορφοποίησης που επιτρέπει τον σχεδιασμό ελκυστικών διεπαφών, με υποστήριξη για animations, responsive layouts (CSS Grid, Flexbox) και custom styles.
* **JavaScript & TypeScript**: Η JavaScript αποτελεί τη βασική γλώσσα προγραμματισμού του διαδικτύου, ενώ η TypeScript προσθέτει τύπους και αυξάνει την αξιοπιστία και τη συντηρησιμότητα του κώδικα (TypeScript, 2025).
* **Angular**: Ένα από τα πιο διαδεδομένα frameworks για την ανάπτυξη μοντέρνων εφαρμογών (SPA), προσφέροντας δομημένη αρχιτεκτονική, επαναχρησιμοποιήσιμα components, reactive programming και εργαλεία για διαχείριση κατάστασης, δρομολόγηση και ασφάλεια (Angular, 2025; RxJS, 2025).
* **RESTful APIs**: Τα σύγχρονα APIs επιτρέπουν την επικοινωνία με εξωτερικές υπηρεσίες (όπως το TMDB), προσφέροντας πρόσβαση σε δεδομένα και λειτουργίες μέσω τυποποιημένων HTTP αιτημάτων (TMDB, 2025).

### 5.2.2 2.2.2 Σύγχρονες αρχές και πρακτικές ανάπτυξης

* **Responsive Design**: Η προσαρμογή της διεπαφής σε διαφορετικές συσκευές και μεγέθη οθόνης αποτελεί βασική απαίτηση, με τεχνικές όπως το mobile-first design και τα media queries.
* **User Experience (UX) & Προσβασιμότητα**: Η έμφαση στη φιλικότητα προς τον χρήστη, την ευκολία πλοήγησης και την προσβασιμότητα για άτομα με αναπηρία, αποτελεί πλέον πρότυπο στη σχεδίαση εφαρμογών (WCAG, 2025).
* **Ασφάλεια**: Η προστασία των δεδομένων και η ασφαλής διαχείριση ταυτοποίησης (π.χ. OAuth) είναι κρίσιμες για κάθε σύγχρονη εφαρμογή.
* **Απόδοση & Βελτιστοποίηση**: Τεχνικές όπως το lazy loading, η βελτιστοποίηση εικόνων και η χρήση cache συμβάλλουν στη γρήγορη και ομαλή λειτουργία (Lighthouse, 2025).
* **Σύγχρονες αρχιτεκτονικές**: Η χρήση patterns όπως Facade, Gateway και Driver, η διαχωρισμένη δομή σε components και services, και η αξιοποίηση του reactive programming, ενισχύουν τη συντηρησιμότητα και την επεκτασιμότητα του κώδικα (Martin, 2008).

[ΣΧΗΜΑ: Αρχιτεκτονική μοντέρνας διαδικτυακής εφαρμογής με διασύνδεση frontend-backend και εξωτερικά APIs]

Συνοψίζοντας, οι τεχνολογίες και οι πρακτικές που εφαρμόζονται στη σύγχρονη ανάπτυξη διαδικτυακών εφαρμογών, όπως το Watchable, διασφαλίζουν υψηλή ποιότητα, ασφάλεια και εμπειρία χρήστη, ενώ επιτρέπουν την εύκολη προσαρμογή και μελλοντική επέκταση της εφαρμογής.

## 5.3 2.3 Το TMDB API και τα δεδομένα πολυμέσων

Το TMDB (The Movie Database) αποτελεί μία από τις μεγαλύτερες και πιο αξιόπιστες διαδικτυακές βάσεις δεδομένων για ταινίες, τηλεοπτικές σειρές και άτομα του χώρου του θεάματος. Το TMDB API (Application Programming Interface) δίνει τη δυνατότητα σε εφαρμογές τρίτων, όπως το Watchable, να έχουν άμεση πρόσβαση σε πλούσιο και διαρκώς ενημερωμένο πολυμεσικό περιεχόμενο (TMDB, 2025).

### 5.3.1 2.3.1 Βασικά χαρακτηριστικά και δεδομένα του TMDB API

Το TMDB API προσφέρει πρόσβαση σε:

* **Ταινίες**: Πληροφορίες για εκατομμύρια ταινίες, όπως τίτλος, περίληψη, ημερομηνία κυκλοφορίας, είδος, διάρκεια, συντελεστές, βαθμολογίες, trailers, αφίσες και εικόνες.
* **Τηλεοπτικές σειρές**: Αναλυτικά δεδομένα για σειρές, σεζόν, επεισόδια, cast, κριτικές, εικόνες και βίντεο.
* **Άτομα**: Βιογραφικά, φιλμογραφία, φωτογραφίες και συνδέσεις με ταινίες/σειρές για ηθοποιούς, σκηνοθέτες και λοιπούς συντελεστές.
* **Εικόνες & Βίντεο**: Υψηλής ανάλυσης αφίσες, backdrops, trailers και άλλο οπτικοακουστικό υλικό.
* **Αξιολογήσεις & Κριτικές**: Βαθμολογίες χρηστών, κριτικές, δημοτικότητα και τάσεις.
* **Λίστες & Watchlists**: Δημιουργία και διαχείριση προσωπικών λιστών και watchlists.
* **Μεταδεδομένα**: Πληροφορίες για είδη, γλώσσες, χώρες παραγωγής, εταιρείες, λέξεις-κλειδιά, πιστοποιήσεις κ.ά.

Το TMDB API υποστηρίζει προηγμένες δυνατότητες αναζήτησης και φιλτραρίσματος, όπως εύρεση ταινιών/σειρών με βάση είδος, δημοτικότητα, ημερομηνία, γλώσσα, χώρα, πάροχο streaming κ.ά. Επιπλέον, προσφέρει υποστήριξη για πολλές γλώσσες και διεθνή δεδομένα, καλύπτοντας τις ανάγκες ενός παγκόσμιου κοινού (TMDB, 2025).

### 5.3.2 2.3.2 Η σημασία του TMDB API για το Watchable

Η εφαρμογή Watchable αξιοποιεί το TMDB API για να παρέχει στους χρήστες της άμεση και αξιόπιστη πληροφόρηση για ταινίες και σειρές, να επιτρέπει την αναζήτηση και οργάνωση περιεχομένου, τη διαχείριση προσωπικών λιστών και την αξιολόγηση τίτλων. Η διασύνδεση με το TMDB API διασφαλίζει τη συνεχή ενημέρωση των δεδομένων, την ποικιλία επιλογών και τη δυνατότητα επέκτασης των λειτουργιών της εφαρμογής στο μέλλον.

[ΣΧΗΜΑ: Ροή δεδομένων μεταξύ Watchable και TMDB API (ενδεικτικά: αιτήματα, απαντήσεις, δεδομένα που αντλούνται)]

Συνοψίζοντας, το TMDB API αποτελεί τον πυρήνα της πληροφορίας για το Watchable, προσφέροντας πλούσια και αξιόπιστα δεδομένα πολυμέσων που ενισχύουν τη λειτουργικότητα και την εμπειρία χρήστη της εφαρμογής.

# 6 3. Ανάλυση απαιτήσεων και σχεδιασμός συστήματος

## 6.1 3.1 Ανάλυση απαιτήσεων χρηστών

Η ανάλυση απαιτήσεων χρηστών για την εφαρμογή Watchable βασίστηκε στη μελέτη των συνηθειών και των αναγκών ατόμων που παρακολουθούν συστηματικά ταινίες και τηλεοπτικές σειρές, καθώς και στη διερεύνηση των ελλείψεων που παρουσιάζουν οι υφιστάμενες πλατφόρμες οργάνωσης πολυμεσικού περιεχομένου. Η διαδικασία ανάλυσης περιλάμβανε παρατήρηση της συμπεριφοράς χρηστών σε αντίστοιχες εφαρμογές και μελέτη σχετικών ερευνών για τις τάσεις στην κατανάλωση πολυμέσικου περιεχομένου.

### 6.1.1 3.1.1 Εστίαση στον χρήστη: σχεδιασμός και λειτουργικότητα

Κεντρικός άξονας του Watchable είναι ο σύγχρονος, μινιμαλιστικός σχεδιασμός και η απλότητα στη χρήση. Η διεπαφή έχει σχεδιαστεί ώστε να παρέχει άμεση πρόσβαση στις βασικές λειτουργίες που χρησιμοποιεί η πλειοψηφία των χρηστών, χωρίς περιττές πολυπλοκότητες. Οι βασικές λειτουργίες που αναδείχθηκαν ως απαραίτητες είναι:

* Αναζήτηση και προβολή πληροφοριών για ταινίες και σειρές με εύχρηστα φίλτρα και προτάσεις.
* Δημιουργία, διαχείριση και αξιολόγηση προσωπικών λιστών (π.χ. «Θέλω να δω», «Έχω δει»).
* Εξατομικευμένες προτάσεις βάσει προτιμήσεων και ιστορικού χρήστη.
* Δυνατότητα αξιολόγησης, σχολιασμού και διαμοιρασμού περιεχομένου με άλλους χρήστες.
* Προβολή βασικών στατιστικών (π.χ. αριθμός ταινιών που έχουν παρακολουθηθεί, αγαπημένα είδη).

Η σχεδίαση της εφαρμογής ακολουθεί τις αρχές του responsive design, ώστε να είναι πλήρως λειτουργική και ευχάριστη σε όλες τις συσκευές (desktop, tablet, κινητό). Ιδιαίτερη έμφαση δόθηκε στην ταχύτητα φόρτωσης, στην καθαρότητα της πληροφορίας και στη δυνατότητα γρήγορης εκτέλεσης των βασικών ενεργειών με ελάχιστα κλικ.

Παράλληλα, η εφαρμογή αποφεύγει την υπερφόρτωση με δευτερεύουσες λειτουργίες που θα μπορούσαν να αποπροσανατολίσουν τον χρήστη, εστιάζοντας σε ό,τι πραγματικά προσφέρει αξία στην καθημερινή εμπειρία του.

### 6.1.2 3.1.2 Ανάγκες και προσδοκίες

Η ανάλυση ανέδειξε τις παρακάτω βασικές ανάγκες:

* **Ευκολία στη χρήση και προσβασιμότητα**: Η διεπαφή πρέπει να είναι φιλική, κατανοητή και προσβάσιμη από διαφορετικές συσκευές (υπολογιστές, tablets, κινητά).
* **Αξιοπιστία και ασφάλεια**: Διασφάλιση της προστασίας των προσωπικών δεδομένων και της αδιάλειπτης λειτουργίας της εφαρμογής.
* **Εξατομίκευση εμπειρίας**: Δυνατότητα δημιουργίας και διαχείρισης προσωπικών λιστών, αποθήκευσης προτιμήσεων και προβολής εξατομικευμένων προτάσεων.
* **Κοινωνική διάσταση**: Δυνατότητα αξιολόγησης, σχολιασμού και διαμοιρασμού περιεχομένου με άλλους χρήστες.
* **Ενσωμάτωση αξιόπιστων δεδομένων**: Άμεση διασύνδεση με το TMDB API για ενημερωμένες πληροφορίες και πλούσιο πολυμεσικό υλικό (TMDB, 2025).

### 6.1.3 3.1.3 Περιορισμοί και προκλήσεις

* **Διαφορετικά επίπεδα εξοικείωσης με την τεχνολογία**: Η εφαρμογή πρέπει να καλύπτει τόσο αρχάριους όσο και έμπειρους χρήστες, χωρίς να θυσιάζει τη λειτουργικότητα ή την απλότητα.
* **Προσβασιμότητα**: Η διεπαφή οφείλει να είναι προσβάσιμη σε άτομα με αναπηρίες, ακολουθώντας τις σχετικές οδηγίες σχεδιασμού (WCAG, 2025).
* **Διαχείριση μεγάλου όγκου δεδομένων**: Η ταχύτητα και η απόκριση της εφαρμογής πρέπει να παραμένουν υψηλές, ακόμη και με μεγάλο αριθμό χρηστών και δεδομένων.
* **Διατήρηση ισορροπίας μεταξύ απλότητας και λειτουργικότητας**: Η πρόκληση είναι να προσφέρονται όλες οι βασικές λειτουργίες χωρίς να γίνεται το περιβάλλον περίπλοκο ή αποθαρρυντικό για τον μέσο χρήστη.

### 6.1.4 3.1.4 Συμπεράσματα ανάλυσης

Η συστηματική ανάλυση των απαιτήσεων χρηστών αποτέλεσε τη βάση για τον καθορισμό των λειτουργικών και μη λειτουργικών απαιτήσεων της Watchable, διασφαλίζοντας ότι η τελική εφαρμογή ανταποκρίνεται στις πραγματικές ανάγκες του κοινού-στόχου και υπερβαίνει τις αδυναμίες των υφιστάμενων λύσεων.

## 6.2 3.2 Λειτουργικές και μη λειτουργικές απαιτήσεις

Η σαφής διατύπωση των λειτουργικών και μη λειτουργικών απαιτήσεων αποτελεί κρίσιμο στάδιο για τον επιτυχημένο σχεδιασμό και υλοποίηση της εφαρμογής Watchable. Οι απαιτήσεις αυτές προέκυψαν από τη συστηματική ανάλυση των αναγκών των χρηστών και τις ιδιαιτερότητες του αντικειμένου της εργασίας.

### 6.2.1 3.2.1 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις βασικές δυνατότητες και υπηρεσίες που πρέπει να προσφέρει η εφαρμογή, ώστε να καλύπτει τις ανάγκες των χρηστών:

* **Αναζήτηση πολυμεσικού περιεχομένου**: Ο χρήστης μπορεί να αναζητήσει ταινίες και σειρές με βάση τίτλο, είδος, έτος, ηθοποιούς, κ.ά.
* **Προβολή λεπτομερειών**: Παρουσίαση αναλυτικών πληροφοριών για κάθε ταινία/σειρά (περίληψη, είδος, διάρκεια, συντελεστές, αξιολογήσεις).
* **Διαχείριση προσωπικών λιστών**: Δυνατότητα δημιουργίας, επεξεργασίας και διαγραφής λιστών (π.χ. «Θέλω να δω», «Έχω δει», «Αγαπημένα»).
* **Αξιολόγηση και σχολιασμός**: Ο χρήστης μπορεί να βαθμολογεί και να σχολιάζει ταινίες/σειρές, καθώς και να διαβάζει αξιολογήσεις άλλων.
* **Διασύνδεση με TMDB API**: Αυτόματη ενημέρωση και εμπλουτισμός των δεδομένων μέσω του TMDB API (TMDB, 2025).
* **Εγγραφή/σύνδεση χρήστη**: Υποστήριξη δημιουργίας λογαριασμού, σύνδεσης και ανάκτησης κωδικού.

### 6.2.2 3.2.2 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις αφορούν τα ποιοτικά χαρακτηριστικά της εφαρμογής και διασφαλίζουν την ομαλή και αποδοτική λειτουργία της:

* **Απόδοση και ταχύτητα**: Η εφαρμογή πρέπει να ανταποκρίνεται άμεσα στις ενέργειες του χρήστη, με ελάχιστο χρόνο φόρτωσης.
* **Διαθεσιμότητα και αξιοπιστία**: Η υπηρεσία πρέπει να είναι διαθέσιμη και λειτουργική χωρίς διακοπές, με πρόβλεψη για ανάκτηση από σφάλματα.
* **Ασφάλεια**: Προστασία προσωπικών δεδομένων, ασφαλής διαχείριση λογαριασμών και επικοινωνίας με εξωτερικές υπηρεσίες.
* **Προσβασιμότητα**: Συμμόρφωση με τα πρότυπα WCAG για άτομα με αναπηρίες (WCAG, 2025).
* **Συμβατότητα**: Ομαλή λειτουργία σε διαφορετικούς browsers και συσκευές (desktop, tablet, κινητό).
* **Επεκτασιμότητα**: Δυνατότητα μελλοντικής προσθήκης νέων λειτουργιών χωρίς σημαντικές αλλαγές στη δομή του συστήματος.
* **Ευκολία συντήρησης**: Καθαρός και τεκμηριωμένος κώδικας, ώστε να διευκολύνεται η συντήρηση και η εξέλιξη της εφαρμογής.

### 6.2.3 3.2.3 Συνοπτική απεικόνιση απαιτήσεων

[ΕΙΚΟΝΑ: Πίνακας ή διάγραμμα που συνοψίζει τις βασικές λειτουργικές και μη λειτουργικές απαιτήσεις της εφαρμογής Watchable.]

## 6.3 3.3 Αρχιτεκτονική συστήματος

Η αρχιτεκτονική της Watchable βασίζεται αποκλειστικά στο frontend, χωρίς τη χρήση ξεχωριστού backend. Όλη η λογική, η διαχείριση δεδομένων και η αλληλεπίδραση με τον χρήστη υλοποιούνται στην πλευρά του client, αξιοποιώντας την Angular. Η εφαρμογή επικοινωνεί απευθείας με το TMDB API μέσω HTTP αιτημάτων, χρησιμοποιώντας το παρεχόμενο API key (Angular, 2025).

### 6.3.1 3.3.1 Κύρια δομικά στοιχεία

* **Frontend (client):**
  + Υλοποιείται με Angular, αξιοποιώντας το οικοσύστημα του framework για τη δημιουργία μοντέρνων, διαδραστικών και επεκτάσιμων διεπαφών χρήστη.
  + Η πλοήγηση, η διαχείριση λιστών, η αναζήτηση, η προβολή λεπτομερειών και όλες οι βασικές λειτουργίες υλοποιούνται ως αυτόνομα components.
  + Η επικοινωνία με το TMDB API γίνεται μέσω ειδικών services και gateways/drivers που αναλαμβάνουν τη διαχείριση των αιτημάτων και την κατασκευή των κατάλληλων URLs.
* **Εξωτερικές υπηρεσίες:**
  + **TMDB API:** Όλα τα δεδομένα για ταινίες, σειρές, ηθοποιούς, εικόνες, αξιολογήσεις κ.λπ. αντλούνται σε πραγματικό χρόνο απευθείας από το TMDB API, με χρήση του API key (TMDB, 2025).
  + Δεν χρησιμοποιείται βάση δεδομένων ή backend server.

### 6.3.2 3.3.2 Ροή δεδομένων και αλληλεπίδραση

1. Ο χρήστης αλληλεπιδρά με το UI (π.χ. αναζήτηση, προσθήκη σε λίστα).
2. Το frontend δημιουργεί το κατάλληλο endpoint για το TMDB API και στέλνει HTTP αίτημα.
3. Τα δεδομένα που επιστρέφονται από το TMDB API επεξεργάζονται και εμφανίζονται στον χρήστη.

### 6.3.3 3.3.3 Αρχιτεκτονικές αρχές και βέλτιστες πρακτικές

* **Απλότητα και ταχύτητα:** Η απουσία backend μειώνει την πολυπλοκότητα και βελτιώνει την απόκριση της εφαρμογής.
* **Διαχωρισμός ανησυχιών:** Η λογική για τα αιτήματα προς το TMDB API, η διαχείριση URLs, η τοπική αποθήκευση και το UI είναι διαχωρισμένα σε services, drivers/gateways και components (TMDB, 2025).
* **Ασφάλεια API key:** Το API key χρησιμοποιείται μόνο για ανάγνωση δημόσιων δεδομένων. Δεν αποθηκεύονται ευαίσθητα δεδομένα χρηστών σε server.
* **Προσβασιμότητα και συμβατότητα:** Υποστήριξη για διαφορετικές συσκευές και browsers, συμμόρφωση με πρότυπα προσβασιμότητας.
* **Επεκτασιμότητα:** Η αρχιτεκτονική επιτρέπει μελλοντική προσθήκη backend αν απαιτηθούν επιπλέον λειτουργίες.

### 6.3.4 3.3.4 Ενδεικτικό διάγραμμα αρχιτεκτονικής

[ΕΙΚΟΝΑ: Διάγραμμα που απεικονίζει το frontend (Watchable) να επικοινωνεί απευθείας με το TMDB API και να διαχειρίζεται τοπικά τις λίστες/ρυθμίσεις χρήστη.]

# 7 4. Υλοποίηση της εφαρμογής Watchable

## 7.1 4.1 Τεχνολογίες που χρησιμοποιήθηκαν

### 7.1.1 4.1.1 Angular

Η Angular αποτελεί ένα σύγχρονο, ολοκληρωμένο framework για την ανάπτυξη δυναμικών, επεκτάσιμων και δομημένων web εφαρμογών. Προσφέρει αρχιτεκτονική βασισμένη σε components, ενσωματωμένο routing, διαχείριση κατάστασης, φόρμες, animations και εργαλεία για ασφαλή διασύνδεση με APIs. Η επιλογή της Angular έγινε λόγω της σταθερότητας, της μεγάλης κοινότητας υποστήριξης, της πληρότητας εργαλείων και των βέλτιστων πρακτικών που προσφέρει για μεγάλα έργα, ενώ η αυστηρή δομή της βοηθά σημαντικά στη συντηρησιμότητα και την επεκτασιμότητα του κώδικα. Στην εφαρμογή Watchable, κάθε λειτουργικό τμήμα (σελίδες, widgets, λίστες, φόρμες) υλοποιήθηκε ως ξεχωριστό component, το routing διαχειρίζεται τη μετάβαση μεταξύ σελίδων και τα services αναλαμβάνουν την επικοινωνία με το TMDB API και τη διαχείριση της κατάστασης (Angular, 2025).

### 7.1.2 4.1.2 RxJS

Η RxJS (Reactive Extensions for JavaScript) είναι μια βιβλιοθήκη για τη διαχείριση ασύγχρονων ροών δεδομένων μέσω observables, operators και subscriptions. Επιλέχθηκε επειδή επιτρέπει την αποτελεσματική διαχείριση δεδομένων που αλλάζουν δυναμικά, όπως τα αποτελέσματα αναζήτησης ή οι ενημερώσεις της watchlist, προσφέροντας real-time εμπειρία και καθαρό κώδικα. Στο Watchable, όλα τα αιτήματα προς το TMDB API, οι ενημερώσεις της διεπαφής και οι αλληλεπιδράσεις του χρήστη υλοποιούνται με observables και operators της RxJS, διασφαλίζοντας ασύγχρονη και αποδοτική ροή δεδομένων (RxJS, 2025).

### 7.1.3 4.1.3 SCSS (Sassy CSS)

Το SCSS είναι μια μεταγλωττιζόμενη γλώσσα που επεκτείνει το CSS με μεταβλητές, nesting, mixins, functions, imports και άλλες δυνατότητες για καλύτερη οργάνωση και επαναχρησιμοποίηση στυλ. Χάρη σε αυτές τις δυνατότητες, το SCSS προσφέρει modularity, ευκολία συντήρησης, παραμετροποίηση και τη δυνατότητα δημιουργίας πολύπλοκων, responsive και επεκτάσιμων UI χωρίς επανάληψη κώδικα, καθιστώντας το ιδανικό για μεγάλα έργα. Στο Watchable, όλο το UI σχεδιάστηκε με custom SCSS, με modules για χρώματα, τυπογραφία, spacing, components, utilities και animations. Χρησιμοποιήθηκαν μεταβλητές για χρώματα και διαστάσεις, mixins για κοινά patterns, nesting για οργάνωση, και custom animations για βελτιωμένη εμπειρία χρήστη (Sass, 2025).

### 7.1.4 4.1.4 placehold.co

Για την περίπτωση που λείπει κάποια εικόνα (όπως poster ή profile), αξιοποιείται το placehold.co, μια online υπηρεσία που δημιουργεί δυναμικά placeholder εικόνες με προσαρμοσμένες διαστάσεις και κείμενο. Η επιλογή αυτής της προσέγγισης εξασφαλίζει ομοιόμορφη και καλαίσθητη εμφάνιση όταν λείπουν εικόνες από το TMDB API, χωρίς να διακόπτεται η εμπειρία χρήστη. Σε κάθε σημείο όπου απαιτείται εικόνα, αν δεν υπάρχει διαθέσιμη, εμφανίζεται αυτόματα placeholder από το placehold.co με κατάλληλο μήνυμα και διαστάσεις (Placehold, 2025).

### 7.1.5 4.1.5 FontAwesome

Το FontAwesome είναι μια βιβλιοθήκη εικονιδίων με χιλιάδες σύγχρονα, αναγνωρίσιμα icons για web εφαρμογές. Προσφέρει πλούσιο οπτικό λεξιλόγιο, ευκολία ενσωμάτωσης και προσαρμογής, και μεγάλη ποικιλία εικονιδίων, γι’ αυτό και επιλέχθηκε για το Watchable. Χρησιμοποιήθηκαν εικονίδια για bookmarks, ratings, navigation, social media και άλλα, ενσωματωμένα απευθείας στα components μέσω SCSS και HTML (FontAwesome, 2025).

### 7.1.6 4.1.6 Swiper

Η βιβλιοθήκη Swiper χρησιμοποιήθηκε για τη δημιουργία responsive sliders/carousels με πλούσιες δυνατότητες παραμετροποίησης και animations. Επιλέχθηκε επειδή παρέχει ομαλές μεταβάσεις, υποστήριξη αφής, responsive συμπεριφορά και ευκολία χρήσης σε Angular projects. Στην εφαρμογή, υλοποιήθηκαν καρουζέλ για ταινίες, σειρές, ηθοποιούς και σεζόνς, με δυνατότητα οριζόντιας κύλισης, αυτόματης προσαρμογής και animations (Swiper, 2025).

### 7.1.7 4.1.7 TMDB API

Το The Movie Database API παρέχει πρόσβαση σε δεδομένα για ταινίες, σειρές, ηθοποιούς, εικόνες, trailers και πολλά ακόμη. Επιλέχθηκε επειδή είναι πλούσιο, ενημερωμένο, δωρεάν για μη εμπορική χρήση και προσφέρει όλα τα απαραίτητα δεδομένα για μια εφαρμογή τύπου Watchable. Μέσω Angular services και RxJS, αντλούνται δεδομένα για προβολή, αναζήτηση, watchlists, λεπτομέρειες, εικόνες και trailers, με caching και βελτιστοποίηση αιτημάτων (TMDB, 2025).

### 7.1.8 4.1.8 TypeScript

Η εφαρμογή υλοποιήθηκε εξ ολοκλήρου σε TypeScript, ένα υπερσύνολο της JavaScript με στατική τυποποίηση, interfaces, enums και σύγχρονες δυνατότητες για ασφαλέστερο και πιο δομημένο κώδικα. Η επιλογή του TypeScript ελαχιστοποιεί τα λάθη, βελτιώνει την τεκμηρίωση και την αναγνωσιμότητα, και είναι η προτεινόμενη γλώσσα για Angular. Στο Watchable, αξιοποιήθηκαν interfaces, enums, types και strict mode για μέγιστη ασφάλεια και συντηρησιμότητα (TypeScript, 2025).

### 7.1.9 4.1.9 Ποιότητα Κώδικα: Linters & Prettier

Για τη διασφάλιση της ποιότητας και της συνέπειας του κώδικα χρησιμοποιήθηκαν εργαλεία αυτόματου ελέγχου ποιότητας (ESLint, Angular ESLint, RxJS plugins) και μορφοποίησης. Αυτά εξασφαλίζουν καθαρό, συνεπή και ευανάγνωστο κώδικα, αποτρέπουν λάθη και ενισχύουν τη συνεργασία στην ανάπτυξη. Εφαρμόστηκαν κανόνες linting για Angular, TypeScript και RxJS, αυτόματο format με Prettier, sorting imports και αφαίρεση αχρησιμοποίητων imports σε όλο το project (Prettier, 2025).

### 7.1.10 4.1.10 Εργαλεία Ανάπτυξης & Διαχείρισης

Για τη διαχείριση εξαρτήσεων, scripts και την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν το Angular CLI, npm, @angular-devkit, @angular-eslint, @typescript-eslint και άλλα εργαλεία. Αυτά αυτοματοποιούν τη δημιουργία, διαχείριση και ανάπτυξη της εφαρμογής, βελτιστοποιούν τη ροή εργασίας και διασφαλίζουν συμβατότητα και ποιότητα. Όλες οι εντολές ανάπτυξης, build, testing, linting και format εκτελούνται μέσω CLI και scripts, ενώ η διαχείριση εξαρτήσεων γίνεται με npm.

## 7.2 4.2 Δομή και οργάνωση του κώδικα

Η οργάνωση του κώδικα στην εφαρμογή Watchable βασίζεται σε μια σύγχρονη, ξεκάθαρη και επεκτάσιμη αρχιτεκτονική, που διευκολύνει τη συντήρηση, την ανάπτυξη νέων λειτουργιών και τη συνεργασία μεταξύ προγραμματιστών. Η δομή των φακέλων και των αρχείων έχει σχεδιαστεί ώστε κάθε λειτουργικό κομμάτι να είναι διακριτό, επαναχρησιμοποιήσιμο και εύκολα εντοπίσιμο, ενώ η ονοματολογία και η τοποθέτηση των αρχείων ακολουθεί τις βέλτιστες πρακτικές του Angular οικοσυστήματος.

Στον root φάκελο της εφαρμογής (src/app) βρίσκονται τα βασικά αρχεία εκκίνησης και ρύθμισης της εφαρμογής. Το αρχείο app.component.ts αποτελεί το “κεντρικό δοχείο” της εφαρμογής, δηλαδή το component που περιέχει το βασικό layout, το navigation και το σημείο όπου “φορτώνονται” δυναμικά όλες οι σελίδες μέσω του Angular Router. Το app.component.html περιέχει τη βασική δομή της σελίδας (header, footer, router outlet), ενώ το app.component.scss τα βασικά στυλ που αφορούν το συνολικό layout. Το app.routes.ts ορίζει όλες τις διαδρομές (routes) της εφαρμογής, δηλαδή ποιο component εμφανίζεται σε κάθε URL, και πώς γίνεται η πλοήγηση μεταξύ των σελίδων. Το app.config.ts περιέχει ρυθμίσεις και παραμέτρους που αφορούν τη συνολική λειτουργία της εφαρμογής.

Ο φάκελος **features** περιέχει όλες τις βασικές λειτουργικές ενότητες της εφαρμογής, οργανωμένες ανά θεματική περιοχή. Για παράδειγμα, υπάρχουν ξεχωριστοί φάκελοι για ταινίες (movies), σειρές (tv-shows), αναζήτηση (search), watchlist, trending, people, ratings κ.ά. Κάθε τέτοιος φάκελος περιλαμβάνει τα components που αφορούν αποκλειστικά τη συγκεκριμένη λειτουργία, τα routes για την πλοήγηση στις επιμέρους σελίδες (π.χ. δημοφιλείς ταινίες, επερχόμενες, λεπτομέρειες ταινίας), καθώς και υποφακέλους για τις σελίδες και τα επιμέρους στοιχεία. Για παράδειγμα, στο features/movies/pages/popular βρίσκονται τα αρχεία που αφορούν τη σελίδα με τις δημοφιλείς ταινίες, ενώ στο features/movies/pages/movie τα αρχεία για τη σελίδα με τις λεπτομέρειες μιας ταινίας. Αυτή η οργάνωση επιτρέπει την απομόνωση της λογικής κάθε λειτουργίας, τη γρήγορη επέκταση ή τροποποίηση χωρίς να επηρεάζεται το υπόλοιπο σύστημα και τη σαφήνεια στην πλοήγηση του κώδικα.

Ο φάκελος **shared** συγκεντρώνει όλα τα κοινόχρηστα στοιχεία που χρησιμοποιούνται σε πολλά σημεία της εφαρμογής. Εδώ βρίσκονται:

* Τα **components** (π.χ. κουμπιά, headers, carousels, κάρτες, loaders, διαλόγους, ειδοποιήσεις, banners, tabs, κ.ά.), που σχεδιάστηκαν ώστε να είναι επαναχρησιμοποιήσιμα και παραμετροποιήσιμα. Για παράδειγμα, το component button χρησιμοποιείται σε όλη την εφαρμογή με διαφορετικές παραμέτρους, ενώ το carousel-media-item εμφανίζει δυναμικά περιεχόμενο σε καρουζέλ.
* Τα **services** (π.χ. authentication, διαχείριση ειδοποιήσεων, local storage, διαχείριση σφαλμάτων, αναζήτηση, διαχείριση διαλόγων), που αναλαμβάνουν τη λογική και τη διαχείριση δεδομένων. Για παράδειγμα, το auth.service.ts διαχειρίζεται τη σύνδεση/αποσύνδεση χρήστη, ενώ το snackbar.service.ts εμφανίζει προσωρινές ειδοποιήσεις.
* Τα **abstract** (αφηρημένες κλάσεις και interfaces που ορίζουν βασικές συμβάσεις και κοινή λογική για components, λίστες, carousels, loaders, κ.λπ.), που επιτρέπουν την υλοποίηση επαναχρησιμοποιήσιμων και επεκτάσιμων δομών με συνέπεια. Για παράδειγμα, η base-media-list-item.abstract.ts ορίζει κοινή συμπεριφορά για όλα τα στοιχεία λίστας πολυμέσων.
* Τα **config** (αρχεία ρυθμίσεων για βιβλιοθήκες ή λειτουργίες, όπως το swiper.config.ts για το carousel ή το navigation.config.ts για το μενού πλοήγησης), που συγκεντρώνουν παραμετροποιήσιμες επιλογές σε ένα σημείο, διευκολύνοντας την αλλαγή συμπεριφοράς χωρίς να επηρεάζεται ο υπόλοιπος κώδικας.
* Τα **interceptors** (π.χ. error.interceptor.ts), που παρεμβάλλονται στα HTTP αιτήματα και απαντήσεις για να προσθέσουν οριζόντια λειτουργικότητα, όπως διαχείριση σφαλμάτων ή authentication tokens, βελτιώνοντας την ασφάλεια και την εμπειρία χρήστη.
* Τα **models** (τύποι και interfaces που περιγράφουν τη δομή των δεδομένων, π.χ. movie.model.ts, tv-show.model.ts, review.model.ts), που διασφαλίζουν αυστηρή τυποποίηση και αποφυγή λαθών κατά την επεξεργασία δεδομένων από το API ή εσωτερικά στην εφαρμογή.
* Τα **drivers** (αρχεία που υλοποιούν συγκεκριμένη λογική για την αλληλεπίδραση με εξωτερικές πηγές ή APIs, π.χ. movie.driver.ts, auth.driver.ts), που διαχωρίζουν την πρόσβαση στα δεδομένα από την υπόλοιπη λογική, επιτρέποντας ευκολότερη συντήρηση και testing.
* Τα **gateways** (αρχεία που λειτουργούν ως “γέφυρες” μεταξύ της εφαρμογής και εξωτερικών υπηρεσιών ή APIs, π.χ. movie.gateway.ts, search.gateway.ts), που συγκεντρώνουν όλη την επικοινωνία με το TMDB API ή άλλες υπηρεσίες, διασφαλίζοντας καθαρό διαχωρισμό ευθυνών και ευκολία στην αλλαγή ή επέκταση της διασύνδεσης.
* Τα **facades** (αρχεία που λειτουργούν ως ενδιάμεσο επίπεδο μεταξύ των components και της λογικής, π.χ. movie.facade.ts, tv-show.facade.ts), που συγκεντρώνουν και απλοποιούν την πρόσβαση σε δεδομένα και λειτουργίες από πολλαπλά gateways, drivers ή services, προσφέροντας μεθόδους που καλύπτουν πλήρως τις ανάγκες της διεπαφής χρήστη. Για παράδειγμα, το facade μπορεί να συνδυάσει δεδομένα από διαφορετικές πηγές (π.χ. δημοφιλείς, επερχόμενες και κορυφαίες ταινίες) και να τα επιστρέψει με μία μόνο κλήση προς το component. Έτσι, τα components παραμένουν απλά, χωρίς να γνωρίζουν τις λεπτομέρειες της επικοινωνίας με το API ή της διαχείρισης των δεδομένων.
* Τα **helpers** (βοηθητικές συναρτήσεις για επεξεργασία δεδομένων, URLs, φίλτρα, genres, authentication, trailers), που διευκολύνουν την επαναχρησιμοποίηση λογικής σε πολλά σημεία του κώδικα.
* Τα **constants** (σταθερές τιμές για default ρυθμίσεις, URLs, παραμέτρους εμφάνισης), όπως το defaults.constant.ts που ορίζει fallback εικόνες, μεγέθη, όρια λιστών κ.λπ.
* Τα **pipes** (για μετασχηματισμό δεδομένων στην εμφάνιση, π.χ. format αριθμών, χρόνου, φύλου, avatar), που επιτρέπουν την εύκολη και καθαρή μετατροπή δεδομένων απευθείας στο template.
* Τα **directives** (για ειδική συμπεριφορά στο UI, όπως fade-in, διαχείριση paths εικόνων, dropdown animations), που προσθέτουν δυναμική συμπεριφορά στα στοιχεία της διεπαφής.
* Τα **enumerations** (για σταθερές κατηγορίες/τιμές που χρησιμοποιούνται σε όλη την εφαρμογή), που διασφαλίζουν συνέπεια και αποφυγή λαθών.
* Τα **styles** (SCSS modules για χρώματα, τυπογραφία, spacing, components, utilities, animations), που διασφαλίζουν ενιαία και παραμετροποιήσιμη εμφάνιση σε όλη την εφαρμογή. Για παράδειγμα, το \_animations.scss περιέχει όλα τα custom keyframes και mixins για εφέ, το \_colors.scss συγκεντρώνει την παλέτα χρωμάτων, ενώ το \_components.scss ορίζει κοινά στυλ για όλα τα UI elements.

Η ύπαρξη αυτών των φακέλων διασφαλίζει ότι κάθε κοινόχρηστο στοιχείο υλοποιείται μία φορά και επαναχρησιμοποιείται όπου χρειάζεται, μειώνοντας την επανάληψη και τα λάθη. Για παράδειγμα, το ίδιο component για κάρτα ταινίας μπορεί να χρησιμοποιηθεί τόσο στη σελίδα των δημοφιλών ταινιών όσο και στη watchlist, με διαφορετικά δεδομένα.

Ο φάκελος **assets** περιέχει όλα τα στατικά αρχεία της εφαρμογής, όπως εικόνες, εικονίδια, γραμματοσειρές και το manifest του site. Η σωστή οργάνωση των assets διευκολύνει την αναφορά και τη διαχείρισή τους, ενώ συμβάλλει στη βελτιστοποίηση της απόδοσης και της εμφάνισης. Για παράδειγμα, τα favicon και τα εικονίδια για mobile συσκευές βρίσκονται εδώ, όπως και τα custom fonts που χρησιμοποιούνται στο UI.

Ο φάκελος **environments** περιλαμβάνει τα αρχεία ρυθμίσεων για τα διαφορετικά περιβάλλοντα (ανάπτυξη, παραγωγή), επιτρέποντας την εύκολη παραμετροποίηση της εφαρμογής ανάλογα με το context εκτέλεσης. Έτσι, μπορούν να αλλάζουν εύκολα τα API keys, τα URLs ή άλλες ρυθμίσεις χωρίς να επηρεάζεται ο υπόλοιπος κώδικας.

Η ροή των δεδομένων στην εφαρμογή ακολουθεί τη λογική της Angular: τα components επικοινωνούν με τα services για να αντλήσουν ή να διαχειριστούν δεδομένα, ενώ τα services χρησιμοποιούν την RxJS για ασύγχρονη διαχείριση και ενημέρωση της διεπαφής. Για παράδειγμα, όταν ο χρήστης κάνει αναζήτηση, το component της αναζήτησης καλεί το αντίστοιχο service, το οποίο επικοινωνεί με το TMDB API και επιστρέφει τα αποτελέσματα ως observable. Τα pipes και τα helpers χρησιμοποιούνται για να μετασχηματίσουν ή να μορφοποιήσουν τα δεδομένα πριν εμφανιστούν στον χρήστη.

Η χρήση αυτής της δομής προσφέρει πολλαπλά οφέλη: διευκολύνει την ανάπτυξη νέων λειτουργιών, επιτρέπει την εύκολη συντήρηση και εντοπισμό σφαλμάτων, ενισχύει τη συνεργασία μεταξύ προγραμματιστών και διασφαλίζει τη σταθερότητα και την ποιότητα του τελικού προϊόντος. Επιπλέον, η modular και feature-based προσέγγιση επιτρέπει την κλιμάκωση της εφαρμογής χωρίς να δημιουργείται σύγχυση. Για παράδειγμα, η προσθήκη μιας νέας λειτουργίας (όπως μια νέα σελίδα για αγαπημένες ταινίες) μπορεί να γίνει με τη δημιουργία ενός νέου φακέλου στο features, χωρίς να επηρεαστεί ο υπόλοιπος κώδικας.

[ΕΙΚΟΝΑ: Αναλυτικό διάγραμμα δομής φακέλων και ροής δεδομένων στην εφαρμογή]

Συνοψίζοντας, η δομή και η οργάνωση του κώδικα στο Watchable αποτελούν θεμέλιο λίθο για την επιτυχία, τη σταθερότητα και τη μελλοντική εξέλιξη της εφαρμογής, προσφέροντας ένα καθαρό, ευέλικτο και επεκτάσιμο περιβάλλον ανάπτυξης.

## 7.3 4.3 Διασύνδεση με το TMDB API

Η διασύνδεση με το TMDB API αποτελεί τον πυρήνα της λειτουργικότητας της εφαρμογής Watchable, καθώς μέσω αυτής αντλούνται όλα τα δεδομένα που αφορούν ταινίες, σειρές, ηθοποιούς, εικόνες, trailers, κριτικές και πολλά ακόμη. Η επιλογή του TMDB ως βασικής πηγής δεδομένων έγινε λόγω της πληρότητας, της αξιοπιστίας και της συνεχούς ενημέρωσης που προσφέρει, καθώς και της ευκολίας πρόσβασης μέσω ενός σύγχρονου RESTful API (TMDB, 2025).

### 7.3.1 4.3.1 Αρχιτεκτονική διασύνδεσης (drivers, gateways, facades)

Η επικοινωνία με το TMDB API υλοποιείται με μια πολυεπίπεδη αρχιτεκτονική που διασφαλίζει καθαρό διαχωρισμό ευθυνών, επαναχρησιμοποίηση και ευκολία συντήρησης. Στο χαμηλότερο επίπεδο, τα **drivers** είναι υπεύθυνα για την πραγματοποίηση των πραγματικών HTTP αιτημάτων προς το TMDB API. Κάθε driver (π.χ. movie.driver.ts) λαμβάνει ένα έτοιμο URL και χρησιμοποιεί μια κοινή υπηρεσία HTTP για να εκτελέσει το αίτημα (GET, POST κ.λπ.), επιστρέφοντας τα δεδομένα ως observables. Οι drivers δεν γνωρίζουν τίποτα για τη λογική της εφαρμογής ή για το πώς κατασκευάζεται το URL – είναι απλοί “εκτελεστές” αιτημάτων.

Πάνω από τους drivers, τα **gateways** αναλαμβάνουν να συνθέσουν τα κατάλληλα URLs για κάθε endpoint του TMDB, να διαχειριστούν παραμέτρους, να συνδυάσουν δεδομένα από πολλαπλά drivers αν χρειάζεται, και να προσφέρουν μεθόδους που αντιστοιχούν σε συγκεκριμένες λειτουργίες της εφαρμογής (π.χ. “φέρε τις δημοφιλείς ταινίες”, “βάλε βαθμολογία σε ταινία”). Τα gateways είναι το σημείο όπου γίνεται η “μετάφραση” των αναγκών της εφαρμογής σε συγκεκριμένα αιτήματα προς το TMDB API. Έτσι, αν αλλάξει η δομή των endpoints ή προστεθούν νέες λειτουργίες, οι αλλαγές περιορίζονται στα gateways.

Σε ακόμα υψηλότερο επίπεδο, τα **facades** λειτουργούν ως ενιαίο API για τα components της εφαρμογής. Ένα facade (π.χ. movie.facade.ts) συγκεντρώνει και απλοποιεί την πρόσβαση σε δεδομένα και λειτουργίες από πολλαπλά gateways, drivers ή services, προσφέροντας μεθόδους που καλύπτουν πλήρως τις ανάγκες της διεπαφής χρήστη. Για παράδειγμα, το facade μπορεί να συνδυάσει δεδομένα από διαφορετικές πηγές (π.χ. δημοφιλείς, επερχόμενες και κορυφαίες ταινίες) και να τα επιστρέψει με μία μόνο κλήση προς το component. Έτσι, τα components παραμένουν απλά, χωρίς να γνωρίζουν τις λεπτομέρειες της επικοινωνίας με το API ή της διαχείρισης των δεδομένων.

Αυτή η αρχιτεκτονική επιλογή προσφέρει σημαντικά οφέλη: διαχωρίζει καθαρά τις ευθύνες (separation of concerns), διευκολύνει την επαναχρησιμοποίηση και τη συντήρηση του κώδικα, επιτρέπει την εύκολη επέκταση ή αλλαγή της διασύνδεσης με το API χωρίς να επηρεάζονται τα components, και καθιστά τη δομή της εφαρμογής πιο κατανοητή και ευέλικτη. Επιπλέον, διευκολύνει το testing, καθώς κάθε επίπεδο μπορεί να ελεγχθεί ανεξάρτητα με mock δεδομένα ή υπηρεσίες.

Για παράδειγμα, όταν ένα component χρειάζεται να εμφανίσει τις δημοφιλείς ταινίες, καλεί μια μέθοδο του movie facade. Το facade ζητά τα δεδομένα από το movie gateway, το gateway κατασκευάζει το κατάλληλο URL και καλεί τον movie driver, και ο driver εκτελεί το HTTP αίτημα προς το TMDB API. Τα δεδομένα επιστρέφουν με τη μορφή observables, περνούν από τα ενδιάμεσα επίπεδα και φτάνουν στο component, έτοιμα για εμφάνιση.

### 7.3.2 4.3.2 Επικοινωνία και ασφάλεια

Η επικοινωνία με το TMDB API πραγματοποιείται αποκλειστικά μέσω ασφαλών HTTP αιτημάτων (HTTPS), διασφαλίζοντας ότι τα δεδομένα που ανταλλάσσονται δεν μπορούν να υποκλαπούν ή να αλλοιωθούν κατά τη μεταφορά. Για την πρόσβαση στα δεδομένα του TMDB απαιτείται η χρήση ενός API key, το οποίο αποθηκεύεται σε αρχεία περιβάλλοντος (environment) της εφαρμογής και προστίθεται αυτόματα σε κάθε αίτημα προς το API.

Είναι σημαντικό να διευκρινιστεί ότι, λόγω της φιλοσοφίας και των περιορισμών του ίδιου του TMDB, το API key είναι ορατό σε όλα τα αιτήματα που γίνονται από το frontend. Το TMDB δεν προσφέρει εναλλακτικές μεθόδους αυθεντικοποίησης για απλά αιτήματα δεδομένων (όπως server-to-server authentication ή πλήρες OAuth flow), με αποτέλεσμα το API key να είναι αναγκαστικά δημόσιο. Ωστόσο, το συγκεκριμένο κλειδί δεν παρέχει πρόσβαση σε ευαίσθητα προσωπικά δεδομένα ή λειτουργίες που θα μπορούσαν να βλάψουν τον χρήστη ή την εφαρμογή. Ο ρόλος του είναι κυρίως η παρακολούθηση της χρήσης (rate limiting) και η αποτροπή κατάχρησης του API, και όχι η προστασία ευαίσθητων πληροφοριών. Για τους σκοπούς της παρούσας εργασίας, η χρήση του API key με αυτόν τον τρόπο είναι αποδεκτή και σύμφωνη με τις οδηγίες του TMDB.

Για την ενίσχυση της ασφάλειας και της αξιοπιστίας της επικοινωνίας, αξιοποιούνται interceptors της Angular. Οι interceptors αυτοί αναλαμβάνουν να προσθέτουν αυτόματα το API key σε κάθε εξερχόμενο αίτημα, να διαχειρίζονται σφάλματα που μπορεί να προκύψουν κατά την επικοινωνία με το API, και να εφαρμόζουν ομοιόμορφη επεξεργασία στις απαντήσεις και τα exceptions. Έτσι, διασφαλίζεται ότι όλα τα αιτήματα είναι σωστά διαμορφωμένα και ότι τυχόν σφάλματα ή αποκλίσεις αντιμετωπίζονται με ενιαίο και προβλέψιμο τρόπο σε όλη την εφαρμογή.

Αξίζει να σημειωθεί ότι η εφαρμογή Watchable δεν διακινεί ποτέ ευαίσθητα προσωπικά δεδομένα χρηστών μέσω του API key. Όλες οι λειτουργίες που απαιτούν ταυτοποίηση ή εξουσιοδότηση (όπως η διαχείριση watchlist ή η βαθμολόγηση ταινιών) υλοποιούνται μέσω του επίσημου authentication flow του TMDB, χωρίς η εφαρμογή να αποθηκεύει ή να διαχειρίζεται απευθείας κωδικούς ή προσωπικά στοιχεία. Η επιλογή αυτής της προσέγγισης είναι απολύτως αποδεκτή, καθώς η εφαρμογή Watchable έχει υλοποιηθεί αποκλειστικά για εκπαιδευτικούς σκοπούς, στο πλαίσιο πτυχιακής εργασίας. Δεν υπάρχει στόχος εμπορικής εκμετάλλευσης, συλλογής προσωπικών δεδομένων ή αύξησης του engagement των χρηστών. Όλες οι αρχιτεκτονικές και τεχνικές επιλογές έγιναν συνειδητά, με πλήρη γνώση των περιορισμών του TMDB API και των βέλτιστων πρακτικών, και είναι απόλυτα εναρμονισμένες με το πλαίσιο και τους στόχους της παρούσας εργασίας.

Συνολικά, η αρχιτεκτονική της επικοινωνίας και της ασφάλειας στο Watchable είναι σχεδιασμένη ώστε να προστατεύει τόσο την εφαρμογή όσο και τον χρήστη, ακολουθώντας τις βέλτιστες πρακτικές και τους περιορισμούς που θέτει το ίδιο το TMDB API. Στην επόμενη ενότητα παρουσιάζεται αναλυτικά η διαδικασία login και register μέσω TMDB (TMDB, 2025).

### 7.3.3 4.3.3 Διαδικασία login και register μέσω TMDB

Ένα ιδιαίτερο χαρακτηριστικό της εφαρμογής Watchable είναι ότι η διαδικασία εγγραφής (register) και σύνδεσης (login) του χρήστη δεν υλοποιείται τοπικά, αλλά γίνεται εξ ολοκλήρου μέσω του TMDB. Ο χρήστης, όταν επιλέξει να συνδεθεί ή να εγγραφεί, ανακατευθύνεται στη σχετική σελίδα του TMDB, όπου πραγματοποιεί την αυθεντικοποίηση και την εξουσιοδότηση της εφαρμογής Watchable να έχει πρόσβαση στα βασικά στοιχεία του λογαριασμού του (μέσω OAuth flow). Μετά την επιτυχή ολοκλήρωση της διαδικασίας, το TMDB επιστρέφει στην εφαρμογή ένα authentication token, το οποίο χρησιμοποιείται για την πραγματοποίηση ενεργειών που απαιτούν ταυτοποίηση (όπως η διαχείριση watchlist, η βαθμολόγηση ταινιών, κ.λπ.).

Αυτή η προσέγγιση έχει ως βασικό πλεονέκτημα ότι η εφαρμογή Watchable δεν διαχειρίζεται ποτέ απευθείας τα προσωπικά στοιχεία ή τους κωδικούς των χρηστών, αυξάνοντας έτσι το επίπεδο ασφάλειας και προστασίας της ιδιωτικότητας. Παράλληλα, ο χρήστης μπορεί να χρησιμοποιεί τον ίδιο λογαριασμό TMDB σε πολλαπλές εφαρμογές και υπηρεσίες, χωρίς να χρειάζεται να δημιουργεί νέο λογαριασμό ειδικά για το Watchable.

[ΕΙΚΟΝΑ: Διάγραμμα ροής authentication και διασύνδεσης με το TMDB API]

### 7.3.4 4.3.4 Οφέλη και προκλήσεις

Η διασύνδεση με το TMDB API προσφέρει πλούσιο και ενημερωμένο περιεχόμενο, ευελιξία στην υλοποίηση λειτουργιών και υψηλό επίπεδο ασφάλειας για τους χρήστες. Παράλληλα, απαιτεί προσεκτική διαχείριση των αιτημάτων, σωστή τυποποίηση των δεδομένων και συνεχή παρακολούθηση των αλλαγών στο API. Η αρχιτεκτονική που υιοθετήθηκε στο Watchable επιτρέπει την εύκολη προσαρμογή σε μελλοντικές αλλαγές ή επεκτάσεις, διατηρώντας τον κώδικα καθαρό, επεκτάσιμο και ασφαλή.

## 7.4 4.4 Σχεδίαση και υλοποίηση διεπαφής χρήστη

Η διεπαφή χρήστη (User Interface – UI) αποτελεί το βασικό σημείο επαφής του χρήστη με την εφαρμογή Watchable και καθορίζει σε μεγάλο βαθμό την εμπειρία, την ευχρηστία και την αποδοχή της εφαρμογής. Από τα πρώτα στάδια του σχεδιασμού, δόθηκε ιδιαίτερη έμφαση στη δημιουργία ενός μοντέρνου, ευχάριστου και λειτουργικού περιβάλλοντος, που να ανταποκρίνεται στις ανάγκες τόσο των αρχάριων όσο και των πιο έμπειρων χρηστών.

### 7.4.1 4.4.1 Αρχές σχεδίασης και στόχοι

Η σχεδίαση της διεπαφής βασίστηκε σε σύγχρονες αρχές UI/UX, με στόχο τη μέγιστη ευχρηστία, την αισθητική αρτιότητα και τη συνέπεια σε όλη την εφαρμογή. Ιδιαίτερη βαρύτητα δόθηκε στα εξής σημεία:

* Απλότητα και καθαρότητα: Η διάταξη των στοιχείων είναι λιτή, χωρίς περιττές πληροφορίες ή οπτικό θόρυβο, ώστε ο χρήστης να εστιάζει στο περιεχόμενο (ταινίες, σειρές, watchlist).
* Συνέπεια: Όλα τα UI elements (κουμπιά, κάρτες, headers, sliders, φόρμες) ακολουθούν ενιαία αισθητική και συμπεριφορά, με κοινή χρωματική παλέτα, τυπογραφία και animations.
* Προσβασιμότητα: Επιλέχθηκαν χρώματα με υψηλή αντίθεση, μεγέθη γραμματοσειρών που διαβάζονται εύκολα και σαφείς ενδείξεις για ενεργά/ανενεργά στοιχεία, ώστε η εφαρμογή να είναι φιλική σε όλους τους χρήστες (WCAG, 2025).
* Mobile-first και responsive design: Η διεπαφή σχεδιάστηκε εξαρχής ώστε να λειτουργεί άψογα σε κινητά, tablets και desktops, με δυναμική προσαρμογή της διάταξης και των στοιχείων ανάλογα με το μέγεθος της οθόνης.
* Άμεση ανατροφοδότηση: Ο χρήστης λαμβάνει άμεσα οπτικό feedback για κάθε ενέργεια (π.χ. προσθήκη σε watchlist, βαθμολόγηση, σφάλματα), με animations, ειδοποιήσεις και αλλαγές κατάστασης.
* Ευκολία πλοήγησης: Το μενού και τα φίλτρα έχουν σχεδιαστεί ώστε ο χρήστης να βρίσκει εύκολα το περιεχόμενο που τον ενδιαφέρει, με ελάχιστα κλικ.

[ΕΙΚΟΝΑ: Wireframe ή screenshot της αρχικής σελίδας και του βασικού navigation]

### 7.4.2 4.4.2 Χρωματική παλέτα, τυπογραφία και εικονίδια

Η αισθητική της εφαρμογής βασίζεται σε μια μοντέρνα χρωματική παλέτα με σκούρο φόντο (dark theme), έντονες αντιθέσεις και διακριτικές χρωματικές πινελιές για σημαντικές ενέργειες (π.χ. primary buttons, ratings). Η τυπογραφία υλοποιήθηκε με custom επιλογή γραμματοσειρών (Lexend, JetBrains Mono, Roboto), προσφέροντας μοντέρνα και ευανάγνωστη εμφάνιση. Τα εικονίδια του FontAwesome ενσωματώθηκαν σε όλα τα βασικά σημεία (navigation, actions, ratings), ενισχύοντας την αναγνωρισιμότητα και τη φιλικότητα του UI.

### 7.4.3 4.4.3 Custom SCSS, animations και skeleton loaders

Όλα τα στυλ της εφαρμογής υλοποιήθηκαν με custom SCSS, αξιοποιώντας μεταβλητές, mixins, nesting και modules για χρώματα, τυπογραφία, spacing, components και animations. Έτσι διασφαλίζεται η συνέπεια, η ευκολία συντήρησης και η δυνατότητα επέκτασης του UI. Ιδιαίτερη έμφαση δόθηκε στα animations (fade-in, slide, skeleton loaders) που προσφέρουν ομαλή εμπειρία κατά τη φόρτωση δεδομένων ή τη μετάβαση μεταξύ σελίδων.

[ΕΙΚΟΝΑ: Παράδειγμα custom SCSS modules, palette, typography, animation]

### 7.4.4 4.4.4 Δομή βασικών σελίδων και επαναχρησιμοποιήσιμα components

Η αρχιτεκτονική του UI βασίζεται σε επαναχρησιμοποιήσιμα Angular components (π.χ. κάρτες ταινιών, sliders, headers, κουμπιά, φόρμες, loaders, banners). Κάθε σελίδα (αρχική, λεπτομέρειες ταινίας, watchlist, αναζήτηση) συντίθεται από τέτοια components, τα οποία παραμετροποιούνται ανάλογα με το περιεχόμενο και το context. Αυτή η προσέγγιση επιτρέπει την ταχεία ανάπτυξη νέων λειτουργιών και τη διατήρηση της συνέπειας σε όλη την εφαρμογή.

[ΕΙΚΟΝΑ: Διάγραμμα ή screenshot με τη σύνθεση μιας σελίδας από components]

### 7.4.5 4.4.5 Προσβασιμότητα και τεχνικές βελτιστοποίησης

Η υλοποίηση του UI στηρίχθηκε σε μια καλά οργανωμένη και επεκτάσιμη αρχιτεκτονική SCSS, με στόχο τη μέγιστη συντηρησιμότητα, επαναχρησιμοποίηση και συνέπεια σε όλη την εφαρμογή. Η δομή των SCSS αρχείων ακολουθεί τη λογική των modules και partials, με κάθε λειτουργικό ή οπτικό κομμάτι να έχει το δικό του αρχείο ή φάκελο. Για παράδειγμα, υπάρχουν ξεχωριστά partials για χρώματα (\_colors.scss), τυπογραφία (\_typography.scss), spacing (\_spacing.scss), animations (\_animations.scss), καθώς και για κάθε βασικό component ή feature (\_button.scss, \_card.scss, \_carousel.scss). Όλα τα partials εισάγονται στο κεντρικό styles.scss μέσω imports, διασφαλίζοντας modular και ευέλικτη οργάνωση.

Κεντρικό ρόλο παίζουν οι μεταβλητές SCSS, οι οποίες ορίζουν την παλέτα χρωμάτων, τα μεγέθη γραμματοσειρών, τα breakpoints για responsive design, τα spacing units και άλλες βασικές παραμέτρους. Έτσι, κάθε αλλαγή στην αισθητική ή τη συμπεριφορά του UI μπορεί να γίνει κεντρικά, χωρίς να απαιτείται αναζήτηση και τροποποίηση σε πολλά σημεία του κώδικα. Για παράδειγμα, η αλλαγή του primary color ή του base font size γίνεται με μία μόνο γραμμή, επηρεάζοντας αυτόματα όλη την εφαρμογή.

Για την αποφυγή επανάληψης και την ενίσχυση της επαναχρησιμοποίησης, αξιοποιήθηκαν mixins και functions. Τα mixins χρησιμοποιούνται για κοινά patterns (π.χ. flexbox layouts, media queries, transitions, shadows), ενώ οι functions επιτρέπουν δυναμικούς υπολογισμούς (π.χ. color manipulation, responsive units). Έτσι, το styling παραμένει DRY (Don’t Repeat Yourself) και ευέλικτο, ενώ διευκολύνεται η προσθήκη νέων features ή η αλλαγή υπάρχοντων.

Η οργάνωση των styles ακολουθεί το separation of concerns: υπάρχουν global styles μόνο για βασικά στοιχεία (body, html, reset), ενώ κάθε component ή feature έχει το δικό του scoped SCSS αρχείο. Αυτό αποτρέπει τα global conflicts και διασφαλίζει ότι οι αλλαγές σε ένα στοιχείο δεν επηρεάζουν απρόβλεπτα άλλα μέρη της εφαρμογής. Για την ονοματολογία των κλάσεων χρησιμοποιήθηκε BEM (Block Element Modifier) convention, που προσφέρει σαφήνεια, προβλεψιμότητα και ευκολία στην αναζήτηση/συντήρηση του κώδικα.

Το responsive design υλοποιήθηκε με χρήση μεταβλητών για breakpoints και mixins για media queries, επιτρέποντας την εύκολη προσαρμογή της διάταξης και των στοιχείων σε διαφορετικές συσκευές. Επιπλέον, δημιουργήθηκαν utility classes για συχνά χρησιμοποιούμενα patterns (π.χ. margin, padding, text alignment), διευκολύνοντας τη γρήγορη ανάπτυξη και τη συνέπεια.

Ιδιαίτερη έμφαση δόθηκε στη δημιουργία custom animations και skeleton loaders, που υλοποιήθηκαν ως ξεχωριστά modules και μπορούν να επαναχρησιμοποιηθούν σε όλη την εφαρμογή. Τα animations (fade-in, slide, scale, shimmer) προσφέρουν ομαλή εμπειρία κατά τη φόρτωση ή τη μετάβαση, ενώ τα skeleton loaders βελτιώνουν την αντιληπτή απόδοση κατά την αναμονή δεδομένων.

Αυτή η αρχιτεκτονική και οι καλές πρακτικές στη χρήση του SCSS διασφαλίζουν ότι το UI της εφαρμογής Watchable παραμένει επεκτάσιμο, ευέλικτο και εύκολο στη συντήρηση, ακόμη και αν προστεθούν νέες λειτουργίες ή αλλάξει η αισθητική στο μέλλον. Παράλληλα, διευκολύνουν τη συνεργασία μεταξύ προγραμματιστών, καθώς κάθε αλλαγή είναι τοπική, προβλέψιμη και τεκμηριωμένη.

## 7.5 4.5 Ενδεικτικά σενάρια χρήσης

Η αξιολόγηση μιας εφαρμογής δεν περιορίζεται μόνο στην τεχνική της υλοποίηση, αλλά επεκτείνεται και στην πραγματική εμπειρία του τελικού χρήστη. Για να αναδειχθούν οι δυνατότητες και η λειτουργικότητα του Watchable, παρουσιάζονται παρακάτω ενδεικτικά σενάρια χρήσης που καλύπτουν τις βασικές και πιο χαρακτηριστικές ροές της εφαρμογής. Κάθε σενάριο περιγράφει τα βήματα που ακολουθεί ο χρήστης, τις αλληλεπιδράσεις με τη διεπαφή και τα αποτελέσματα που λαμβάνει, αναδεικνύοντας τόσο την τεχνική υλοποίηση όσο και την εμπειρία χρήστη (UX).

### 7.5.1 4.5.1 Σενάριο 1: Αναζήτηση και εξερεύνηση ταινιών

Ο χρήστης ανοίγει την αρχική σελίδα του Watchable και χρησιμοποιεί τη γραμμή αναζήτησης για να βρει μια συγκεκριμένη ταινία ή να εξερευνήσει νέες προτάσεις. Καθώς πληκτρολογεί, εμφανίζονται δυναμικά προτεινόμενα αποτελέσματα (autocomplete), με τίτλο, εικόνα και βασικές πληροφορίες. Επιλέγοντας μια ταινία, μεταφέρεται στη σελίδα λεπτομερειών, όπου μπορεί να δει περιγραφή, βαθμολογία, trailer, cast και σχετικές προτάσεις. Η πλοήγηση είναι άμεση, με animations και skeleton loaders κατά τη φόρτωση των δεδομένων.

[ΕΙΚΟΝΑ: Screenshot ή διάγραμμα ροής αναζήτησης και εμφάνισης λεπτομερειών ταινίας]

### 7.5.2 4.5.2 Σενάριο 2: Προσθήκη ταινίας στη watchlist

Αφού εντοπίσει μια ενδιαφέρουσα ταινία, ο χρήστης μπορεί να την προσθέσει στη watchlist του με ένα κλικ στο σχετικό εικονίδιο (bookmark). Αν δεν είναι συνδεδεμένος, εμφανίζεται ειδοποίηση που τον προτρέπει να κάνει login μέσω TMDB. Μετά τη σύνδεση, η προσθήκη ολοκληρώνεται και η ταινία εμφανίζεται στη watchlist, με άμεσο οπτικό feedback (animation, snackbar). Ο χρήστης μπορεί να διαχειριστεί τη watchlist του από τη σχετική σελίδα, αφαιρώντας ή προσθέτοντας ταινίες κατά βούληση.

[ΕΙΚΟΝΑ: Screenshot watchlist, ειδοποίηση login, feedback προσθήκης]

### 7.5.3 4.5.3 Σενάριο 3: Αξιολόγηση ταινίας και διαχείριση βαθμολογιών

Ο χρήστης, αφού παρακολουθήσει μια ταινία, μπορεί να την αξιολογήσει μέσω του συστήματος βαθμολόγησης (αστέρια). Η αξιολόγηση αποστέλλεται στο TMDB μέσω του αντίστοιχου API, και η διεπαφή ενημερώνεται άμεσα με το νέο score. Αν ο χρήστης αλλάξει γνώμη, μπορεί να τροποποιήσει ή να αφαιρέσει τη βαθμολογία του. Όλες οι ενέργειες συνοδεύονται από animations και ειδοποιήσεις για την επιτυχή ή αποτυχημένη ολοκλήρωση.

[ΕΙΚΟΝΑ: Παράδειγμα αξιολόγησης ταινίας, feedback, αλλαγή βαθμολογίας]

### 7.5.4 4.5.4 Σενάριο 4: Πλοήγηση σε κατηγορίες και εξερεύνηση περιεχομένου

Ο χρήστης μπορεί να περιηγηθεί σε διάφορες κατηγορίες (δημοφιλείς, επερχόμενες, κορυφαίες ταινίες, σειρές, άτομα) μέσω του βασικού μενού και των φίλτρων. Κάθε κατηγορία εμφανίζεται με καρουζέλ, κάρτες και sliders, επιτρέποντας γρήγορη επισκόπηση και επιλογή. Η responsive σχεδίαση διασφαλίζει άριστη εμπειρία σε όλες τις συσκευές, ενώ τα skeleton loaders και τα animations βελτιώνουν την αντιληπτή απόδοση.

[ΕΙΚΟΝΑ: Screenshot κατηγοριών, καρουζέλ, responsive διάταξη]

### 7.5.5 4.5.5 Σενάριο 5: Σύνδεση και αυθεντικοποίηση μέσω TMDB

Όταν ο χρήστης επιθυμεί να προσθέσει ταινία στη watchlist ή να αξιολογήσει, αλλά δεν είναι συνδεδεμένος, επιλέγει να κάνει login. Η εφαρμογή τον ανακατευθύνει στη σελίδα αυθεντικοποίησης του TMDB, όπου ολοκληρώνει τη διαδικασία με ασφάλεια. Μετά την επιτυχή σύνδεση, επιστρέφει αυτόματα στο Watchable, όπου μπορεί να συνεχίσει τις ενέργειές του χωρίς να χρειάζεται να εισάγει ξανά στοιχεία. Η διαδικασία είναι απλή, ασφαλής και διαφανής για τον χρήστη.

[ΕΙΚΟΝΑ: Διάγραμμα ροής authentication, screenshot login flow]

Τα παραπάνω σενάρια αναδεικνύουν τη ροή και τη λειτουργικότητα της εφαρμογής Watchable, εστιάζοντας τόσο στην τεχνική υλοποίηση όσο και στην εμπειρία του χρήστη. Μέσα από ρεαλιστικές καταστάσεις, φαίνεται πώς η αρχιτεκτονική, η σχεδίαση και οι τεχνολογίες που επιλέχθηκαν συνδυάζονται για να προσφέρουν ένα σύγχρονο, αποδοτικό και ευχάριστο περιβάλλον χρήσης.

# 8 5. Έλεγχος, αξιολόγηση και βελτιστοποίηση

## 8.1 5.1 Μέθοδοι ελέγχου και δοκιμών

Η διασφάλιση της ποιότητας και της αξιοπιστίας της εφαρμογής Watchable αποτέλεσε βασικό στόχο κατά τη διάρκεια της ανάπτυξης. Οι μέθοδοι ελέγχου και δοκιμών που εφαρμόστηκαν είχαν ως σκοπό να εντοπίσουν πιθανά σφάλματα, να διασφαλίσουν την ομαλή λειτουργία των βασικών ροών και να βελτιώσουν τη συνολική εμπειρία του χρήστη.

### 8.1.1 5.1.1 Έλεγχος λειτουργικότητας

Η λειτουργικότητα της εφαρμογής Watchable ελέγχθηκε διεξοδικά μέσω χειροκίνητων δοκιμών που κάλυψαν όλα τα βασικά και δευτερεύοντα σενάρια χρήσης. Εξετάστηκαν επίσης ειδικές περιπτώσεις, όπως η αναζήτηση με κενό ή μη έγκυρο όρο, η προσπάθεια προσθήκης ταινίας χωρίς σύνδεση, η διαχείριση λιστών με μεγάλο αριθμό στοιχείων και η αλλαγή αξιολόγησης σε ήδη βαθμολογημένη ταινία. Όπου εντοπίστηκαν σφάλματα ή ασυνέπειες, καταγράφηκαν και διορθώθηκαν άμεσα, με επαναληπτικό έλεγχο μετά τη διόρθωση. Η διαδικασία αυτή διασφάλισε ότι όλες οι βασικές και κρίσιμες ροές λειτουργούν απρόσκοπτα και σύμφωνα με τις προσδοκίες του χρήστη.

### 8.1.2 5.1.2 Έλεγχος συμβατότητας και προσβασιμότητας

Για τη διασφάλιση της συμβατότητας, η εφαρμογή δοκιμάστηκε σε όλους τους δημοφιλείς browsers (Google Chrome, Mozilla Firefox, Microsoft Edge) και σε διαφορετικά λειτουργικά συστήματα (Windows, macOS, Android). Ελέγχθηκε η ορθή εμφάνιση και λειτουργία του UI σε desktop, tablet και κινητά τηλέφωνα, με ιδιαίτερη προσοχή στη σωστή προσαρμογή του layout και των διαδραστικών στοιχείων. Η προσβασιμότητα αξιολογήθηκε με χρήση εργαλείων όπως το Lighthouse και το axe DevTools, καθώς και με χειροκίνητο έλεγχο: δοκιμάστηκε η επάρκεια του contrast, η ύπαρξη εναλλακτικών κειμένων σε εικόνες και η συμβατότητα με screen readers. Όπου διαπιστώθηκαν ελλείψεις (π.χ. χαμηλό contrast, μη προσβάσιμα κουμπιά), έγιναν οι απαραίτητες διορθώσεις ώστε η εφαρμογή να είναι φιλική και σε άτομα με αναπηρίες, σύμφωνα με τις βασικές αρχές των προτύπων WCAG.

### 8.1.3 5.1.3 Έλεγχος απόδοσης

Η απόδοση της εφαρμογής Watchable αξιολογήθηκε με συστηματικό τρόπο, καθώς αποτελεί κρίσιμο παράγοντα για τη συνολική εμπειρία του χρήστη. Η διαδικασία περιλάμβανε:

* **Έλεγχο ταχύτητας φόρτωσης**: Μετρήθηκε ο χρόνος που απαιτείται για την αρχική φόρτωση της εφαρμογής και την εμφάνιση του περιεχομένου, τόσο σε desktop όσο και σε κινητές συσκευές.
* **Ανταπόκριση διεπαφής**: Εξετάστηκε η ταχύτητα με την οποία ανταποκρίνονται τα κουμπιά, τα μενού και οι βασικές λειτουργίες στις ενέργειες του χρήστη, ακόμη και όταν υπάρχουν πολλαπλές ταυτόχρονες αλληλεπιδράσεις.
* **Διαχείριση μεγάλου όγκου δεδομένων**: Ελέγχθηκε η ομαλή λειτουργία της εφαρμογής κατά την εμφάνιση μεγάλων λιστών (π.χ. δημοφιλείς ταινίες, watchlist), ώστε να διασφαλιστεί ότι δεν παρατηρούνται καθυστερήσεις ή κολλήματα.
* **Απόδοση σε διαφορετικές συσκευές και δίκτυα**: Πραγματοποιήθηκαν δοκιμές τόσο σε σύγχρονους υπολογιστές όσο και σε κινητά τηλέφωνα, καθώς και σε συνθήκες αργής σύνδεσης (π.χ. 3G), για να διαπιστωθεί η ανθεκτικότητα της εφαρμογής.

Για την αντικειμενική αξιολόγηση της απόδοσης χρησιμοποιήθηκε το εργαλείο **Lighthouse** της Google, το οποίο παρέχει αναλυτικές μετρήσεις για την ταχύτητα, τη βελτιστοποίηση και τη συνολική εμπειρία χρήστη. Τα βασικά κριτήρια που εξετάστηκαν ήταν:

* **Performance** (απόδοση): Ταχύτητα φόρτωσης, interactivity, βελτιστοποίηση πόρων.
* **Accessibility** (προσβασιμότητα): Κατά πόσο η εφαρμογή είναι φιλική σε άτομα με αναπηρίες.
* **Best Practices** (βέλτιστες πρακτικές): Τήρηση σύγχρονων προτύπων ανάπτυξης.
* **SEO** (βελτιστοποίηση για μηχανές αναζήτησης): Ορατότητα και προσβασιμότητα του περιεχομένου.

Τα αποτελέσματα του Lighthouse έδειξαν υψηλές επιδόσεις στους παραπάνω τομείς, με ιδιαίτερα θετικές ενδείξεις στην ταχύτητα φόρτωσης και την ανταπόκριση της διεπαφής. Οι όποιες παρατηρήσεις/προτάσεις βελτίωσης που εντοπίστηκαν μέσω του εργαλείου αξιοποιήθηκαν για περαιτέρω βελτιστοποίηση του κώδικα και των ρυθμίσεων της εφαρμογής.

[ΕΙΚΟΝΑ: Αποτελέσματα ελέγχου απόδοσης με Lighthouse (screenshot)]

### 8.1.4 5.1.4 Έλεγχος ποιότητας κώδικα

Η ποιότητα του κώδικα διασφαλίστηκε με τη συστηματική χρήση εργαλείων αυτόματου ελέγχου και μορφοποίησης. Ενσωματώθηκαν τα ESLint, Angular ESLint και Prettier στη ροή ανάπτυξης, με υποχρεωτικό linting και format πριν από κάθε commit. Τα εργαλεία αυτά εντόπισαν συντακτικά λάθη, ασυνέπειες στη μορφοποίηση, αχρησιμοποίητες μεταβλητές και πιθανά προβλήματα απόδοσης ή ασφάλειας. Όλες οι παρατηρήσεις διορθώνονταν άμεσα, ενώ η τεκμηρίωση του κώδικα και η χρήση σαφών ονομάτων για μεταβλητές, συναρτήσεις και components συνέβαλαν στη συντηρησιμότητα και την ευκολία επέκτασης της εφαρμογής. Η συνεχής εφαρμογή αυτών των πρακτικών διασφάλισε ότι ο κώδικας παραμένει καθαρός, ευανάγνωστος και εύκολα διαχειρίσιμος από οποιονδήποτε μελλοντικό συνεισφέροντα.

### 8.1.5 5.1.5 Εσωτερική αξιολόγηση και δοκιμές από χρήστες

Η διαδικασία εσωτερικής αξιολόγησης περιλάμβανε συνεχή αυτοέλεγχο από τον προγραμματιστή, αλλά και τη συμμετοχή τρίτων χρηστών που κλήθηκαν να δοκιμάσουν την εφαρμογή σε πραγματικές συνθήκες. Οι testers κλήθηκαν να εκτελέσουν συγκεκριμένα σενάρια (π.χ. αναζήτηση, προσθήκη/αφαίρεση από watchlist, αξιολόγηση, πλοήγηση σε διαφορετικές σελίδες) και να καταγράψουν τυχόν δυσκολίες, σφάλματα ή προτάσεις βελτίωσης. Ενδεικτικά, προτάθηκαν αλλαγές στη διάταξη των κουμπιών, βελτιώσεις στα μηνύματα ειδοποιήσεων και διόρθωση μικρών ασυνεπειών στη συμπεριφορά της διεπαφής. Το feedback αξιοποιήθηκε άμεσα, με στόχο τη βελτίωση της χρηστικότητας και της συνολικής εμπειρίας πλοήγησης.

### 8.1.6 5.1.6 Περιορισμοί και μελλοντικές βελτιώσεις

Βασικός περιορισμός κατά τη διάρκεια των ελέγχων ήταν ο διαθέσιμος χρόνος, που δεν επέτρεψε την υλοποίηση πλήρους αυτοματοποιημένου ελέγχου (unit tests, end-to-end tests) σε όλο το εύρος της εφαρμογής. Επίσης, η απουσία πραγματικών δεδομένων χρηστών και η χρήση μόνο του TMDB API περιόρισε τη δυνατότητα ελέγχου σε σενάρια με πολύ μεγάλο όγκο δεδομένων ή ακραίες περιπτώσεις χρήσης. Σε μελλοντικές εκδόσεις, προτείνεται η ενσωμάτωση αυτοματοποιημένων tests, η χρήση εργαλείων CI/CD και η διεύρυνση της βάσης testers με άτομα διαφορετικών προφίλ. Παρά τους περιορισμούς, η εφαρμογή των παραπάνω μεθόδων ελέγχου συνέβαλε ουσιαστικά στη βελτίωση της αξιοπιστίας, της απόδοσης και της φιλικότητας της εφαρμογής Watchable.

## 8.2 5.2 Προτάσεις για μελλοντικές βελτιώσεις

Η συνεχής βελτίωση μιας εφαρμογής αποτελεί βασικό παράγοντα για τη διατήρηση της αξίας και της ανταγωνιστικότητάς της. Με βάση τα αποτελέσματα των ελέγχων, τις παρατηρήσεις των χρηστών και τις σύγχρονες τάσεις στην ανάπτυξη web εφαρμογών, προτείνονται οι παρακάτω κατευθύνσεις για τη μελλοντική εξέλιξη του Watchable:

### 8.2.1 5.2.1 Ενίσχυση αυτοματοποίησης ελέγχων και ποιότητας

* **Υλοποίηση unit tests και end-to-end tests:** Η προσθήκη αυτοματοποιημένων δοκιμών (π.χ. με Jasmine, Karma, Cypress) θα διασφαλίσει τη σταθερότητα του κώδικα σε κάθε νέα αλλαγή, θα μειώσει τα σφάλματα και θα επιτρέψει την ταχύτερη ανάπτυξη νέων λειτουργιών.
* **Ενσωμάτωση εργαλείων CI/CD:** Η χρήση πλατφορμών CI/CD (π.χ. GitHub Actions, GitLab CI) θα αυτοματοποιήσει τη διαδικασία build, test και deployment, διασφαλίζοντας ότι κάθε έκδοση είναι αξιόπιστη και έτοιμη για παραγωγή.
* **Αυτοματοποιημένοι έλεγχοι προσβασιμότητας:** Ενσωμάτωση εργαλείων όπως το axe-core για συνεχή έλεγχο WCAG compliance.

### 8.2.2 5.2.2 Βελτιστοποίηση απόδοσης και τεχνικής υποδομής

* **Progressive Web App (PWA) χαρακτηριστικά:** Υλοποίηση offline λειτουργίας, caching και εγκατάστασης ως native app για βελτιωμένη εμπειρία σε κινητές συσκευές.
* **Βελτιστοποίηση φόρτωσης δεδομένων:** Εφαρμογή τεχνικών όπως lazy loading, code splitting και βελτιστοποίηση εικόνων για ακόμη ταχύτερη απόκριση.
* **Αναβάθμιση στην τελευταία έκδοση Angular και βιβλιοθηκών:** Διατήρηση της εφαρμογής ενημερωμένης για λόγους ασφάλειας, απόδοσης και νέων δυνατοτήτων.
* **Υποστήριξη dark/light theme:** Ενσωμάτωση δυνατότητας εναλλαγής μεταξύ σκοτεινού και φωτεινού θέματος, ώστε ο χρήστης να επιλέγει το περιβάλλον που τον εξυπηρετεί καλύτερα, βελτιώνοντας την προσβασιμότητα και την εμπειρία χρήσης σε διαφορετικές συνθήκες φωτισμού.
* **Χρήση design tokens:** Υιοθέτηση design tokens για την κεντρική διαχείριση χρωμάτων, τυπογραφίας, spacing και άλλων παραμέτρων UI, διευκολύνοντας τη συνέπεια, την παραμετροποίηση και τη συντήρηση του οπτικού σχεδιασμού.

### 8.2.3 5.2.3 Εμπλουτισμός λειτουργιών και εμπειρίας χρήστη

* **Προσωποποιημένες προτάσεις περιεχομένου:** Ανάπτυξη αλγορίθμων που θα προτείνουν ταινίες/σειρές βάσει του ιστορικού και των προτιμήσεων του χρήστη, αξιοποιώντας machine learning ή advanced filtering.
* **Επέκταση κοινωνικών λειτουργιών:** Δυνατότητα αλληλεπίδρασης μεταξύ χρηστών (π.χ. σχόλια, ανταλλαγή watchlists, likes σε αξιολογήσεις).
* **Υποστήριξη πολλαπλών γλωσσών (i18n):** Ενσωμάτωση συστήματος μεταφράσεων για να γίνει η εφαρμογή προσβάσιμη σε ευρύτερο κοινό.
* **Ειδοποιήσεις (notifications):** Προσθήκη push notifications για υπενθυμίσεις, νέες κυκλοφορίες ή δραστηριότητα στη watchlist.
* **Δημιουργία component library:** Ανάπτυξη βιβλιοθήκης επαναχρησιμοποιήσιμων UI components, ώστε να διασφαλίζεται συνέπεια, επεκτασιμότητα και ευκολία συντήρησης σε όλο το project ή και σε μελλοντικά projects.
* **Ενσωμάτωση Storybook:** Χρήση του Storybook για την τεκμηρίωση, οπτικοποίηση και απομόνωση των components, διευκολύνοντας τη συνεργασία μεταξύ σχεδιαστών και προγραμματιστών και τη δοκιμή UI στοιχείων ανεξάρτητα από το υπόλοιπο σύστημα.

### 8.2.4 5.2.4 Επέκταση διασύνδεσης και υποδομής

* **Υποστήριξη και άλλων APIs:** Ενσωμάτωση δεδομένων από επιπλέον πηγές (π.χ. IMDb, Rotten Tomatoes) για πλουσιότερο περιεχόμενο και αξιολογήσεις.
* **Εξαγωγή/εισαγωγή watchlist:** Δυνατότητα μεταφοράς λιστών από/προς άλλες πλατφόρμες για ευκολότερη μετάβαση χρηστών.
* **Ανάπτυξη backend server:** Δημιουργία backend server για την υποστήριξη επιπλέον λειτουργιών, όπως αποθήκευση προτιμήσεων, διαχείριση χρηστών, custom notifications ή analytics, προσφέροντας μεγαλύτερη ευελιξία και δυνατότητα επέκτασης της εφαρμογής πέρα από τα όρια του TMDB API.

### 8.2.5 5.2.5 Τεκμηρίωση και κοινότητα

* **Αναλυτική τεκμηρίωση χρήσης και ανάπτυξης:** Δημιουργία πλήρους οδηγού για χρήστες και developers, με παραδείγματα, screenshots και συχνές ερωτήσεις.
* **Δημιουργία κοινότητας χρηστών:** Ενθάρρυνση συμμετοχής μέσω forum, feedback forms ή social media για συνεχή βελτίωση της εφαρμογής.

**Συνοψίζοντας**, οι παραπάνω προτάσεις στοχεύουν στη διαρκή αναβάθμιση της εφαρμογής Watchable, τόσο σε τεχνικό όσο και σε λειτουργικό επίπεδο. Η υλοποίησή τους θα ενισχύσει την αξιοπιστία, την απόδοση, τη φιλικότητα και τη διαχρονικότητα της εφαρμογής, προσφέροντας μια ολοκληρωμένη εμπειρία σε ακόμη περισσότερους χρήστες.

# 9 6. Αποτελέσματα και παρατηρήσεις

## 9.1 6.1 Αποτελέσματα ως προς τη λειτουργικότητα

Η εφαρμογή ανταποκρίθηκε με επιτυχία στα βασικά σενάρια χρήσης που είχαν τεθεί ως στόχοι: αναζήτηση και προβολή ταινιών/σειρών, προσθήκη και διαχείριση watchlist, αξιολόγηση περιεχομένου, πλοήγηση σε λεπτομέρειες και διαχείριση λιστών. Όλες οι βασικές ροές λειτούργησαν απρόσκοπτα, χωρίς σφάλματα ή ασυνέπειες. Ειδικές περιπτώσεις, όπως η αναζήτηση με κενό ή μη έγκυρο όρο, η προσπάθεια προσθήκης χωρίς σύνδεση και η διαχείριση μεγάλων λιστών, αντιμετωπίστηκαν με κατάλληλα μηνύματα και ομαλή συμπεριφορά της διεπαφής. Οι testers ανέφεραν ότι η εφαρμογή είναι εύχρηστη και κατανοητή, με σαφή διαχωρισμό λειτουργιών και άμεση απόκριση στις ενέργειες του χρήστη.

## 9.2 6.2 Αποτελέσματα ως προς την απόδοση

Οι μετρήσεις από το εργαλείο Lighthouse έδειξαν πολύ υψηλές επιδόσεις στους τομείς της ταχύτητας φόρτωσης, της διαδραστικότητας και της βελτιστοποίησης πόρων. Σε δοκιμές τόσο σε desktop όσο και σε κινητές συσκευές, ο χρόνος αρχικής φόρτωσης κυμάνθηκε σε πολύ χαμηλά επίπεδα (<2 δευτερόλεπτα σε γρήγορη σύνδεση), ενώ η διεπαφή παρέμεινε άμεση και ομαλή ακόμη και σε συνθήκες αυξημένου φόρτου δεδομένων. Οι όποιες παρατηρήσεις του Lighthouse (π.χ. μικρές βελτιώσεις σε εικόνες ή caching) αξιοποιήθηκαν για περαιτέρω βελτιστοποίηση. Η εφαρμογή διατήρησε υψηλή απόδοση και σε αργές συνδέσεις, με ομαλή εμπειρία χρήστη.

[ΕΙΚΟΝΑ: Αποτελέσματα Lighthouse με επιδόσεις]

## 9.3 6.3 Αποτελέσματα ως προς τη συμβατότητα και προσβασιμότητα

Η εφαρμογή λειτούργησε ομαλά σε όλους τους βασικούς browsers (Chrome, Firefox, Edge) και σε διαφορετικά λειτουργικά συστήματα (Windows, macOS, Android). Το responsive design διασφάλισε την ορθή εμφάνιση και λειτουργία σε desktop, tablet και κινητά. Η προσβασιμότητα αξιολογήθηκε θετικά, με επαρκές contrast και εναλλακτικά κείμενα σε εικόνες. Όπου εντοπίστηκαν ελλείψεις (π.χ. χαμηλό contrast σε δευτερεύοντα κουμπιά), έγιναν διορθώσεις ώστε να καλύπτονται οι βασικές αρχές των προτύπων WCAG. Οι testers με διαφορετικό επίπεδο εξοικείωσης ανέφεραν ότι η εφαρμογή είναι φιλική και προσβάσιμη.

[ΕΙΚΟΝΑ: Screenshot responsive εμφάνισης σε κινητό και desktop]

## 9.4 6.4 Αποτελέσματα ως προς την εμπειρία χρήστη (UX)

Η συνολική εμπειρία χρήστη αξιολογήθηκε ως ιδιαίτερα θετική. Οι testers επισήμαναν την καθαρότητα της διεπαφής, την ευκολία πλοήγησης και την άμεση απόκριση των λειτουργιών. Ιδιαίτερα θετικά σχολιάστηκαν τα animations, τα feedback μηνύματα και η συνοχή του οπτικού σχεδιασμού. Μικρές δυσκολίες που εντοπίστηκαν (π.χ. ασαφή μηνύματα σε σπάνιες περιπτώσεις, μικρές καθυστερήσεις σε πολύ μεγάλες λίστες) αντιμετωπίστηκαν με βελτιώσεις στη διατύπωση και στη διαχείριση των δεδομένων. Η εφαρμογή θεωρήθηκε κατάλληλη τόσο για αρχάριους όσο και για πιο έμπειρους χρήστες.

## 9.5 6.5 Αποτελέσματα ως προς την ποιότητα κώδικα

Η χρήση εργαλείων linting και μορφοποίησης (ESLint, Angular ESLint, Prettier) συνέβαλε στη διατήρηση καθαρού, ευανάγνωστου και συντηρήσιμου κώδικα. Εντοπίστηκαν και διορθώθηκαν συντακτικά λάθη, αχρησιμοποίητες μεταβλητές και ασυνέπειες στη μορφοποίηση. Η τεκμηρίωση και η σαφής ονοματολογία διευκολύναν τη συντήρηση και την επέκταση της εφαρμογής. Η συνολική ποιότητα του κώδικα θεωρείται υψηλή, με δυνατότητα εύκολης μελλοντικής εξέλιξης.

## 9.6 6.6 Συνοπτική αποτίμηση

Τα αποτελέσματα των ελέγχων και οι παρατηρήσεις από τη χρήση της εφαρμογής Watchable τεκμηριώνουν την επιτυχία της ως προς τους βασικούς στόχους: αξιοπιστία, απόδοση, φιλικότητα και προσβασιμότητα. Παρά τους περιορισμούς (π.χ. απουσία αυτοματοποιημένων tests, χρήση μόνο του TMDB API), η εφαρμογή ανταποκρίθηκε στις απαιτήσεις και προσφέρει μια ολοκληρωμένη εμπειρία στον τελικό χρήστη. Οι παρατηρήσεις που καταγράφηκαν αποτελούν πολύτιμο οδηγό για μελλοντικές βελτιώσεις και επέκταση των δυνατοτήτων της εφαρμογής.

# 10 7. Συμπεράσματα

## 10.1 7.1 Ανακεφαλαίωση βασικών σημείων

Η παρούσα πτυχιακή εργασία είχε ως βασικό στόχο τη μελέτη, σχεδίαση και υλοποίηση της διαδικτυακής εφαρμογής Watchable, με σκοπό να προσφέρει μια σύγχρονη, φιλική και λειτουργική πλατφόρμα για την αναζήτηση, οργάνωση και αξιολόγηση πολυμεσικού περιεχομένου, με έμφαση στις ταινίες και τις τηλεοπτικές σειρές.

Αρχικά, παρουσιάστηκαν τα κίνητρα και η σημασία του θέματος, αναδεικνύοντας την ανάγκη για εργαλεία που διευκολύνουν τους χρήστες στη διαχείριση του αυξανόμενου όγκου πολυμέσικου υλικού. Μέσα από τη μελέτη των υφιστάμενων λύσεων και ανταγωνιστικών πλατφορμών, εντοπίστηκαν τα πλεονεκτήματα και οι αδυναμίες τους, τα οποία λήφθηκαν υπόψη στον σχεδιασμό του Watchable, με στόχο τη βελτίωση της εμπειρίας χρήστη και την κάλυψη πραγματικών αναγκών.

Στη συνέχεια, πραγματοποιήθηκε αναλυτική καταγραφή των απαιτήσεων των χρηστών, τόσο σε λειτουργικό όσο και σε ποιοτικό επίπεδο. Η εργασία εστίασε στην ευκολία χρήσης, την προσβασιμότητα, την αξιοπιστία και την εξατομίκευση της εμπειρίας, ενώ δόθηκε ιδιαίτερη έμφαση στη διασύνδεση με αξιόπιστες πηγές δεδομένων, όπως το TMDB API. Ο σχεδιασμός του συστήματος βασίστηκε σε αρχές μινιμαλισμού, modularity και responsive design, ώστε η εφαρμογή να είναι προσιτή και λειτουργική σε κάθε συσκευή και για κάθε κατηγορία χρήστη.

Η υλοποίηση πραγματοποιήθηκε εξ ολοκλήρου με το framework Angular, αξιοποιώντας σύγχρονες τεχνολογίες και εργαλεία, όπως TypeScript, RxJS, SCSS, FontAwesome και Swiper. Η αρχιτεκτονική του κώδικα οργανώθηκε σε διακριτές ενότητες (features, shared, assets, environments), με στόχο τη συντηρησιμότητα, την επεκτασιμότητα και την ευκολία ανάπτυξης νέων λειτουργιών. Η επικοινωνία με το TMDB API υλοποιήθηκε μέσω ειδικών services και gateways, διασφαλίζοντας την άμεση και αξιόπιστη άντληση δεδομένων (Angular, 2025; TypeScript, 2025; RxJS, 2025; SCSS, 2025; FontAwesome, 2025; Swiper, 2025).

Η εφαρμογή υποβλήθηκε σε εκτενείς ελέγχους λειτουργικότητας, απόδοσης, συμβατότητας και προσβασιμότητας, με χρήση εργαλείων όπως το Lighthouse και το axe DevTools, αλλά και μέσω δοκιμών από πραγματικούς χρήστες. Τα αποτελέσματα των ελέγχων ανέδειξαν την αξιοπιστία, τη φιλικότητα και την υψηλή απόδοση της εφαρμογής, ενώ καταγράφηκαν προτάσεις για μελλοντικές βελτιώσεις, όπως η ενίσχυση των αυτοματοποιημένων tests, η υποστήριξη PWA χαρακτηριστικών και η περαιτέρω εξατομίκευση της εμπειρίας χρήστη (Lighthouse, 2025).

Συνολικά, η εργασία απέδειξε ότι ο συνδυασμός σύγχρονων τεχνολογιών, σωστής ανάλυσης απαιτήσεων και προσεκτικού σχεδιασμού μπορεί να οδηγήσει στη δημιουργία μιας ολοκληρωμένης και ποιοτικής εφαρμογής, που ανταποκρίνεται στις ανάγκες του σύγχρονου χρήστη και θέτει τις βάσεις για περαιτέρω εξέλιξη και εμπλουτισμό λειτουργιών στο μέλλον.

## 10.2 7.2 Προοπτικές αποτίμησης και εμπειρίες

Η ενασχόληση με τη μελέτη, τον σχεδιασμό και την υλοποίηση της εφαρμογής Watchable αποτέλεσε μια ιδιαίτερα σημαντική και πολυδιάστατη εμπειρία, τόσο σε τεχνικό όσο και σε προσωπικό επίπεδο. Η διαδικασία αυτή έδωσε τη δυνατότητα εμβάθυνσης σε σύγχρονες τεχνολογίες, ανάπτυξης δεξιοτήτων επίλυσης προβλημάτων και βαθύτερης κατανόησης των απαιτήσεων που συνοδεύουν την ανάπτυξη μιας ολοκληρωμένης διαδικτυακής εφαρμογής.

### 10.2.1 7.2.1 Εμβάθυνση σε σύγχρονες τεχνολογίες και μεθοδολογίες

Η επιλογή της Angular ως βασικού framework για το frontend οδήγησε σε εις βάθος μελέτη της φιλοσοφίας του, των αρχών του component-based σχεδιασμού, της διαχείρισης κατάστασης με RxJS, καθώς και της χρήσης TypeScript για ασφαλέστερο και πιο δομημένο κώδικα. Η υλοποίηση της επικοινωνίας με το TMDB API, η διαχείριση ασύγχρονων ροών δεδομένων και η οργάνωση του κώδικα σε modules, services και components, συνέβαλαν στην κατανόηση της σημασίας της modular αρχιτεκτονικής και της επεκτασιμότητας. Παράλληλα, η ενασχόληση με εργαλεία όπως το SCSS, το FontAwesome και το Swiper, ενίσχυσε τις γνώσεις στον σχεδιασμό φιλικών και προσβάσιμων διεπαφών χρήστη.

### 10.2.2 7.2.2 Προκλήσεις και τρόποι αντιμετώπισης

Κατά τη διάρκεια της εργασίας, προέκυψαν ποικίλες προκλήσεις, όπως η διαχείριση μεγάλου όγκου δεδομένων, η διασφάλιση της απόδοσης και της προσβασιμότητας, καθώς και η ανάγκη για καθαρό και τεκμηριωμένο κώδικα. Η απουσία backend και η εξάρτηση αποκλειστικά από το TMDB API απαίτησε ιδιαίτερη προσοχή στη διαχείριση των αιτημάτων και στην ασφάλεια των δεδομένων. Επιπλέον, η προσπάθεια διατήρησης της εφαρμογής απλής και φιλικής, χωρίς να θυσιάζεται η λειτουργικότητα, αποτέλεσε μια συνεχή άσκηση ισορροπίας. Μέσω της διαδικασίας επίλυσης αυτών των προβλημάτων, αναπτύχθηκαν δεξιότητες αναζήτησης λύσεων, αυτονομίας και προσαρμοστικότητας.

### 10.2.3 7.2.3 Απόκτηση δεξιοτήτων και προσωπική εξέλιξη

Η ολοκλήρωση της εργασίας συνέβαλε σημαντικά στην ανάπτυξη τόσο τεχνικών όσο και οριζόντιων δεξιοτήτων. Βελτιώθηκε η ικανότητα διαχείρισης χρόνου, ο σχεδιασμός και η υλοποίηση πολύπλοκων έργων, καθώς και η συνεργασία με εργαλεία version control και testing. Η ανάγκη για συνεχή έλεγχο ποιότητας, τεκμηρίωση και συμμόρφωση με πρότυπα προσβασιμότητας ενίσχυσε την προσοχή και τη μεθοδικότητα κατά την εκτέλεση των εργασιών. Παράλληλα, η ενασχόληση με πραγματικά σενάρια χρήσης και η λήψη feedback από testers συνέβαλαν στην καλύτερη κατανόηση των αναγκών των τελικών χρηστών και στην προσαρμογή των λύσεων ανάλογα.

### 10.2.4 7.2.4 Συμπεράσματα και μελλοντικές προοπτικές

Η συμμετοχή στην ανάπτυξη του Watchable αποτέλεσε μια ολοκληρωμένη μαθησιακή εμπειρία, η οποία εφοδίασε με πολύτιμες γνώσεις και αυτοπεποίθηση για μελλοντικά έργα. Η διαδικασία αυτή ανέδειξε τη σημασία της σωστής ανάλυσης απαιτήσεων, του προσεκτικού σχεδιασμού και της συνεχούς βελτίωσης. Τα εφόδια που αποκτήθηκαν εκτιμάται ότι θα αποτελέσουν ισχυρή βάση για την επαγγελματική εξέλιξη στον χώρο της ανάπτυξης λογισμικού, ενώ η εμπειρία της υλοποίησης μιας πραγματικής εφαρμογής με σύγχρονες τεχνολογίες αναμένεται να συμβάλει στην αποτελεσματική ανταπόκριση σε μελλοντικές προκλήσεις.

# 11 8. Βιβλιογραφία

### 11.0.1 8.1 Εισαγωγή

Η βιβλιογραφία αποτελεί θεμέλιο λίθο κάθε επιστημονικής εργασίας, καθώς τεκμηριώνει το θεωρητικό και τεχνολογικό υπόβαθρο, διασφαλίζει τη διαφάνεια και επιτρέπει την επαλήθευση των πληροφοριών και των μεθοδολογιών που χρησιμοποιήθηκαν. Στην παρούσα εργασία, η επιλογή των πηγών έγινε με κριτήριο την αξιοπιστία, την επικαιρότητα και τη συνάφεια με το αντικείμενο της ανάπτυξης web εφαρμογών με έμφαση στην εμπειρία χρήστη και τη χρήση σύγχρονων τεχνολογιών όπως η Angular.

### 11.0.2 8.2 Βιβλία & Επιστημονικά Άρθρα

1. Martin, R. C. (2008). *Clean code: A handbook of agile software craftsmanship*. Prentice Hall.

### 11.0.3 8.3 Διαδικτυακές Πηγές & Τεκμηρίωση

1. TMDB. (2025). *The Movie Database (TMDB) API Documentation*. https://developer.themoviedb.org/docs
2. IMDb. (2025). *IMDb API Documentation*. https://developer.imdb.com/
3. Letterboxd. (2025). *Letterboxd API Documentation*. https://letterboxd.com/api-beta/
4. GoodWatch. (2025). *GoodWatch Platform*. https://goodwatch.app/
5. RxJS. (2025). *RxJS Documentation*. https://rxjs.dev/guide/overview
6. Sass. (2025). *Sass: Syntactically Awesome Style Sheets*. https://sass-lang.com/guide
7. FontAwesome. (2025). *Font Awesome Documentation*. https://fontawesome.com/docs
8. Swiper. (2025). *Swiper Documentation*. https://swiperjs.com/angular
9. Jasmine. (2025). *Jasmine Documentation*. https://jasmine.github.io/
10. Karma. (2025). *Karma - Spectacular Test Runner for JavaScript*. https://karma-runner.github.io/
11. Cypress. (2025). *Cypress Documentation*. https://docs.cypress.io/
12. W3C. (2025). *Web Content Accessibility Guidelines (WCAG) 2.1*. https://www.w3.org/WAI/standards-guidelines/wcag/
13. Google. (2025). *Google Fonts*. https://fonts.google.com/
14. Google. (2025). *Lighthouse: Open-Source, Automated Tool for Improving the Quality of Web Pages*. https://developer.chrome.com/docs/lighthouse/overview/
15. Placehold. (2025). *Placehold.co - A simple image placeholder service*. https://placehold.co/
16. Angular. (2025). *Angular - The modern web developer’s platform*. https://angular.io/
17. Microsoft. (2025). *TypeScript: JavaScript With Syntax For Types*. https://www.typescriptlang.org/

### 11.0.4 8.4 Παρατηρήσεις για τη Βιβλιογραφία

* Όλες οι πηγές επιλέχθηκαν με βάση την αξιοπιστία, την επικαιρότητα και τη συνάφεια με το αντικείμενο της εργασίας.
* Δόθηκε προτεραιότητα σε επίσημη τεκμηρίωση, επιστημονικά συγγράμματα και αναγνωρισμένες πλατφόρμες.
* Η βιβλιογραφία καλύπτει τόσο το θεωρητικό υπόβαθρο όσο και τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη της εφαρμογής Watchable.

# 12 9. Παραρτήματα

## 12.1 9.1 Οδηγίες εγκατάστασης και χρήσης

Η ενότητα αυτή παρέχει αναλυτικές οδηγίες για την εγκατάσταση και τη χρήση της εφαρμογής Watchable, ώστε κάθε χρήστης ή αξιολογητής να μπορεί να εκτελέσει και να αξιολογήσει την εφαρμογή με επιτυχία. Οι οδηγίες καλύπτουν τόσο το περιβάλλον ανάπτυξης όσο και τη διαδικασία εκκίνησης και βασικής χρήσης της εφαρμογής. Η εφαρμογή έχει υλοποιηθεί με το framework Angular και απαιτεί βασικές γνώσεις χειρισμού γραμμής εντολών και εγκατάστασης λογισμικού.

**Εναλλακτικά, η εφαρμογή είναι διαθέσιμη και online στη διεύθυνση** [**https://watchable-frontend.netlify.app/**](https://watchable-frontend.netlify.app/) **για άμεση δοκιμή χωρίς τοπική εγκατάσταση.**

### 12.1.1 9.1.1 Προαπαιτούμενα

* **Node.js** (έκδοση 18.x ή νεότερη)
* **npm** (έκδοση 9.x ή νεότερη)
* **Git** (για λήψη του πηγαίου κώδικα)
* **Σύγχρονος browser** (π.χ. Google Chrome, Mozilla Firefox, Microsoft Edge)

### 12.1.2 9.1.2 Λήψη του πηγαίου κώδικα

1. Ανοίξτε τη γραμμή εντολών (Command Prompt, Terminal, PowerShell).
2. Εκτελέστε την εντολή: bash git clone https://github.com/angel/watchable.frontend.git
3. Μεταβείτε στον φάκελο του έργου: bash cd watchable.frontend

### 12.1.3 9.1.3 Εγκατάσταση εξαρτήσεων

1. Βεβαιωθείτε ότι βρίσκεστε στον φάκελο του έργου.
2. Εκτελέστε την εντολή: bash npm install Η εντολή αυτή θα εγκαταστήσει αυτόματα όλες τις απαραίτητες βιβλιοθήκες και εξαρτήσεις που απαιτούνται για τη λειτουργία της εφαρμογής.

### 12.1.4 9.1.4 Εκκίνηση της εφαρμογής σε περιβάλλον ανάπτυξης

1. Εκτελέστε την εντολή: bash npm start
2. Μετά από λίγα δευτερόλεπτα, η εφαρμογή θα είναι διαθέσιμη στη διεύθυνση: http://localhost:4200
3. Ανοίξτε τον browser και μεταβείτε στη διεύθυνση αυτή για να δείτε την εφαρμογή σε λειτουργία.

### 12.1.5 9.1.5 Βασικές οδηγίες χρήσης

* **Αναζήτηση ταινιών/σειρών:** Χρησιμοποιήστε τη γραμμή αναζήτησης στην αρχική σελίδα για να βρείτε τίτλους.
* **Περιήγηση σε κατηγορίες:** Επιλέξτε από το μενού τις κατηγορίες (δημοφιλείς, επερχόμενες, κορυφαίες ταινίες/σειρές).
* **Προσθήκη σε watchlist:** Κάντε κλικ στο εικονίδιο «bookmark» για να προσθέσετε μια ταινία/σειρά στη λίστα σας (απαιτείται σύνδεση μέσω TMDB).
* **Αξιολόγηση:** Βαθμολογήστε ταινίες/σειρές μέσω του συστήματος αστεριών (απαιτείται σύνδεση).
* **Διαχείριση λογαριασμού:** Συνδεθείτε μέσω TMDB για να διαχειριστείτε τις προσωπικές σας λίστες και αξιολογήσεις.

### 12.1.6 9.1.6 Συχνά προβλήματα και λύσεις

* **Το npm install αποτυγχάνει:** Ελέγξτε ότι έχετε εγκατεστημένη τη σωστή έκδοση Node.js και npm.
* **Η εφαρμογή δεν ανοίγει στο browser:** Βεβαιωθείτε ότι το npm start εκτελείται χωρίς σφάλματα και ότι δεν χρησιμοποιείται ήδη η θύρα 4200.
* **Σφάλματα κατά τη σύνδεση με TMDB:** Ελέγξτε τη σύνδεση στο διαδίκτυο και τα δικαιώματα του λογαριασμού TMDB.

Ακολουθώντας τα παραπάνω βήματα, μπορείτε να εγκαταστήσετε και να χρησιμοποιήσετε την εφαρμογή Watchable τοπικά στον υπολογιστή σας. Για οποιοδήποτε πρόβλημα ή απορία, συμβουλευτείτε το αρχείο README του έργου ή επικοινωνήστε με τον υπεύθυνο ανάπτυξης.

## 12.2 9.2 Πρόσθετα σχήματα, screenshots, διαγράμματα

Η ενότητα αυτή περιλαμβάνει πρόσθετα σχήματα, screenshots και διαγράμματα που υποστηρίζουν και επεξηγούν τη λειτουργικότητα, τη δομή και τη χρήση της εφαρμογής Watchable. Κάθε οπτικό υλικό συνοδεύεται από αναλυτική περιγραφή και επεξήγηση της σημασίας του, ώστε να διευκολύνεται η κατανόηση της αρχιτεκτονικής, της διεπαφής και των βασικών ροών της εφαρμογής.

#### 12.2.0.1 9.2.1 Διάγραμμα αρχιτεκτονικής εφαρμογής

[ΕΙΚΟΝΑ: Διάγραμμα που απεικονίζει τη ροή δεδομένων μεταξύ του frontend (Angular), του TMDB API και του χρήστη. Να φαίνεται η χρήση drivers, gateways, facades και η απουσία backend.]

**Περιγραφή:** Το διάγραμμα αυτό παρουσιάζει τη συνολική αρχιτεκτονική της εφαρμογής Watchable. Ο χρήστης αλληλεπιδρά με το frontend, το οποίο είναι υλοποιημένο με Angular. Όλα τα δεδομένα αντλούνται απευθείας από το TMDB API μέσω ειδικών drivers και gateways, χωρίς τη μεσολάβηση backend. Η χρήση facades απλοποιεί την επικοινωνία μεταξύ των components και των υπηρεσιών δεδομένων.

#### 12.2.0.2 9.2.2 Screenshot αρχικής σελίδας εφαρμογής

[ΕΙΚΟΝΑ: Screenshot του browser με την αρχική σελίδα της εφαρμογής Watchable, όπως εμφανίζεται στο http://localhost:4200 ή στο https://watchable-frontend.netlify.app/]

**Περιγραφή:** Η αρχική σελίδα της εφαρμογής προσφέρει άμεση πρόσβαση στις βασικές λειτουργίες: αναζήτηση, περιήγηση σε κατηγορίες, πρόσβαση σε watchlist και αξιολογήσεις. Η αισθητική είναι μοντέρνα, με σκοτεινό θέμα, ευδιάκριτα κουμπιά και responsive διάταξη.

#### 12.2.0.3 9.2.3 Διάγραμμα δομής φακέλων και αρχείων

[ΕΙΚΟΝΑ: Διάγραμμα ή screenshot που απεικονίζει τη δομή των φακέλων και των αρχείων του έργου (features, shared, assets, environments, κ.λπ.)]

**Περιγραφή:** Το διάγραμμα αυτό βοηθά στην κατανόηση της οργάνωσης του πηγαίου κώδικα. Κάθε φάκελος έχει συγκεκριμένο ρόλο (features για λειτουργικές ενότητες, shared για κοινόχρηστα στοιχεία, assets για στατικά αρχεία, environments για ρυθμίσεις). Η δομή αυτή διευκολύνει τη συντήρηση και την επέκταση της εφαρμογής.

#### 12.2.0.4 9.2.4 Screenshot λειτουργίας αναζήτησης

[ΕΙΚΟΝΑ: Screenshot που δείχνει τη γραμμή αναζήτησης σε χρήση, με προτεινόμενα αποτελέσματα να εμφανίζονται καθώς ο χρήστης πληκτρολογεί.]

**Περιγραφή:** Η λειτουργία αναζήτησης επιτρέπει στον χρήστη να βρίσκει γρήγορα ταινίες και σειρές. Τα αποτελέσματα εμφανίζονται δυναμικά, με εικόνες και βασικές πληροφορίες, βελτιώνοντας την εμπειρία χρήστη.

#### 12.2.0.5 9.2.5 Screenshot watchlist και αξιολόγησης

[ΕΙΚΟΝΑ: Screenshot που απεικονίζει τη σελίδα watchlist με προσθήκη/αφαίρεση τίτλων και τη λειτουργία αξιολόγησης με αστέρια.]

**Περιγραφή:** Η watchlist επιτρέπει στον χρήστη να οργανώνει τις ταινίες και σειρές που θέλει να παρακολουθήσει. Η αξιολόγηση γίνεται εύκολα μέσω συστήματος αστεριών, με άμεση ενημέρωση της διεπαφής.

#### 12.2.0.6 9.2.6 Screenshot responsive σχεδίασης

[ΕΙΚΟΝΑ: Screenshot της εφαρμογής σε κινητό ή tablet, που δείχνει την προσαρμογή της διάταξης και των στοιχείων UI σε μικρή οθόνη.]

**Περιγραφή:** Η εφαρμογή Watchable έχει σχεδιαστεί με mobile-first λογική και responsive layout, ώστε να λειτουργεί άψογα σε όλες τις συσκευές. Τα στοιχεία της διεπαφής προσαρμόζονται αυτόματα για βέλτιστη εμπειρία χρήστη.

Τα παραπάνω σχήματα, screenshots και διαγράμματα αποτελούν πολύτιμο υλικό για την κατανόηση της λειτουργίας, της αρχιτεκτονικής και της εμπειρίας χρήστη της εφαρμογής Watchable. Συνιστάται η ενσωμάτωσή τους στα αντίστοιχα σημεία της εργασίας ή η αναφορά τους κατά την παρουσίαση/αξιολόγηση του έργου.