ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΛΟΠΟΙΗΣΗ FULL-STACK WEB ΕΦΑΡΜΟΓΗΣ ΑΘΛΗΤΙΚΗΣ ΠΛΗΡΟΦΟΡΗΣΗΣ

Δούκας Παρασκευάς

Επιβλέπων: Φώτιος Βαρτζιώτης

Επίκουρος Καθηγητής

Άρτα, Μάρτιος, 2026

IMPLEMENTATION OF A FULL-STACK WEB APPLICATION FOR SPORTS INFORMATION

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα,16/03/2026

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

   Φώτιος Βαρτζιώτης,

2. Μέλος επιτροπής

   Όνομα Επίθετο,

3. Μέλος επιτροπής

   Όνομα Επίθετο,

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Επίθετο, Όνομα

Υπογραφή

Περίληψη

Η παρούσα εργασία πραγματεύεται τη σχεδίαση και υλοποίηση της διαδικτυακής εφαρμογής "Sport90", μιας σύγχρονης πλατφόρμας παροχής αθλητικών δεδομένων σε πραγματικό χρόνο. Σκοπός του έργου είναι η δημιουργία ενός λειτουργικού συστήματος που επιτρέπει στους χρήστες να παρακολουθούν βαθμολογίες, αποτελέσματα αγώνων, ρόστερ ομάδων και ειδησεογραφία από τα σημαντικότερα ευρωπαϊκά πρωταθλήματα ποδοσφαίρου.

Η αρχιτεκτονική της εφαρμογής βασίζεται στο μοντέλο τριών επιπέδων (3-tier architecture). Για την ανάπτυξη της δυναμικής διεπαφής χρήστη (Frontend) χρησιμοποιήθηκε η βιβλιοθήκη **React**, ενώ για τη δημιουργία ενός διακομιστή μεσολάβησης (Proxy Server) αξιοποιήθηκε το περιβάλλον **Node.js** με το framework **Express**. Ο Proxy Server διαδραματίζει κρίσιμο ρόλο, καθώς διασφαλίζει την ασφαλή επικοινωνία με την εξωτερική πηγή δεδομένων Football-Data.org API, επιλύοντας παράλληλα προβλήματα ασφάλειας και περιορισμών CORS.

Στο επίπεδο της υλοποίησης, δόθηκε ιδιαίτερη έμφαση στη διαχείριση καταστάσεων μέσω των React Hooks (useState, useEffect) για τον αυτόματο συγχρονισμό των δεδομένων, καθώς και στον αποκρινόμενο σχεδιασμό (Responsive Design) μέσω του **Tailwind CSS** για την απρόσκοπτη λειτουργία σε κάθε είδους συσκευή. Τα αποτελέσματα καταδεικνύουν την αποτελεσματικότητα του συνδυασμού αυτών των τεχνολογιών στην κατασκευή γρήγορων και κλιμακώσιμων εφαρμογών, θέτοντας τις βάσεις για μελλοντικές επεκτάσεις όπως η ενσωμάτωση βάσεων δεδομένων.

Λέξεις-κλειδιά: Full-Stack Ανάπτυξη, React, Node.js Proxy Server, Αθλητική Πληροφόρηση, REST API.

Abstract

This thesis explores the design and implementation of "Sport90", a modern web application providing real-time sports data. The objective of this project is to create a functional system that allows users to track league standings, match results, team rosters, and football news from the most prominent European leagues.

The application architecture is based on a three-tier model, utilizing the **React** library for the development of a dynamic frontend and the **Node.js** environment with the **Express** framework for the creation of a **Proxy Server**. The Proxy Server plays a vital role by ensuring secure communication with the external Football-Data.org API, effectively resolving security issues and CORS restrictions.

On the implementation level, significant emphasis was placed on state management using **React Hooks** (useState, useEffect) for automatic data synchronization, as well as on **Responsive Design** using **Tailwind CSS** to ensure seamless operation across desktops and mobile devices. The project results demonstrate the effectiveness of combining these technologies in building fast, secure, and scalable applications, while establishing a foundation for future enhancements such as database integration and real-time updates via WebSockets.

Keywords: Full-Stack Development, React, Node.js Proxy Server, Sports Information, REST API.

Περιεχόμενα

Εισαγωγή

Η ταχεία εξέλιξη των τεχνολογιών του παγκόσμιου ιστού (World Wide Web) έχει μεταβάλει ριζικά τον τρόπο με τον οποίο οι χρήστες καταναλώνουν αθλητική πληροφορία. Στο σύγχρονο ψηφιακό περιβάλλον, η ανάγκη για άμεση, έγκυρη και οπτικά ελκυστική πρόσβαση σε δεδομένα πραγματικού χρόνου είναι πιο επιτακτική από ποτέ. Η παρούσα πτυχιακή εργασία εστιάζει στον σχεδιασμό και την ανάπτυξη μιας ολοκληρωμένης διαδικτυακής πλατφόρμας ποδοσφαιρικών στατιστικών, η οποία συνδυάζει σύγχρονα front-end frameworks με ισχυρές back-end τεχνικές διαμεσολάβησης δεδομένων.

1.1 Αντικείμενο της Πτυχιακής Εργασίας

Το κύριο αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός, η αρχιτεκτονική σύνθεση και η υλοποίηση μιας προηγμένης διαδικτυακής πλατφόρμας με την ονομασία **"Sport90"**. Η εφαρμογή αποτελεί ένα ολοκληρωμένο σύστημα παροχής ποδοσφαιρικών πληροφοριών σε πραγματικό χρόνο, εστιάζοντας στην ορθή διαχείριση και οπτικοποίηση μεγάλου όγκου δεδομένων. Η εφαρμογή έχει αναπτυχθεί ως **Single Page Application (SPA)** χρησιμοποιώντας τη βιβλιοθήκη **React**. Η επιλογή αυτής της αρχιτεκτονικής επιτρέπει την ταχύτατη εναλλαγή περιεχομένου χωρίς την ανάγκη αναφόρτωσης ολόκληρης της σελίδας από τον εξυπηρετητή, προσφέροντας στον χρήστη μια εμπειρία παρόμοια με αυτή μιας

τοπικής εφαρμογής (native app). Η δρομολόγηση των σελίδων (routing) επιτυγχάνεται μέσω του react-router-dom, διασφαλίζοντας τη συνεκτικότητα μεταξύ των διαφορετικών ενοτήτων όπως η αρχική σελίδα, οι σελίδες πρωταθλημάτων και οι λεπτομέρειες των ομάδων.

Ένα από τα βασικά αντικείμενα της εργασίας είναι η ανάλυση και η υλοποίηση μιας **component-based** αρχιτεκτονικής. Η εφαρμογή διασπάται σε αυτόνομα και επαναχρησιμοποιούμενα στοιχεία:

- **Διαχείριση Δεδομένων Πρωταθλήματος**: Υλοποίηση των components DataTable και FullTable για την προβολή βαθμολογιών.
- **Δυναμική Πληροφορία Ομάδας**: Δημιουργία των TeamData, SquadTeam και TeamMatches που συνεργάζονται για την πλήρη απεικόνιση του προφίλ ενός συλλόγου.
- **Στοιχεία Διεπαφής**: Ανάπτυξη των Header και Footer για σταθερή πλοήγηση και επικοινωνία.

Ιδιαίτερη βαρύτητα στην ανάπτυξη δόθηκε στην επίλυση τεχνικών περιορισμών που προκύπτουν από την κατανάλωση εξωτερικών υπηρεσιών (APIs). Το αντικείμενο περιλαμβάνει την υλοποίηση ενός **Proxy Server** με τη χρήση **Node.js** και **Express**. Ο διακομιστής αυτός επιτελεί δύο κρίσιμες λειτουργίες:

1. **Επίλυση προβλημάτων CORS (Cross-Origin Resource Sharing)**: Επιτρέπει την ασφαλή ανταλλαγή δεδομένων μεταξύ του client και του API του Football-Data.org.
2. **Ασφάλεια Δεδομένων**: Διαχειρίζεται το API Key (X-Auth-Token) στην πλευρά του διακομιστή, αποτρέποντας την έκθεσή του στο frontend και την κατάχρησή του από τρίτους.

Τέλος, το αντικείμενο της πτυχιακής καλύπτει την εφαρμογή του **Tailwind CSS** για τη δημιουργία ενός πλήρως αποκρινόμενου σχεδιασμού (Responsive Design). Η εργασία μελετά πώς η πληροφορία προσαρμόζεται δυναμικά από μεγάλες οθόνες desktop σε περιβάλλοντα κινητών τηλεφώνων, διατηρώντας την αναγνωσιμότητα των σύνθετων πινάκων βαθμολογίας και των αγωνιστικών στατιστικών.

1.2 Περιγραφή της Ιδέας

Η κεντρική φιλοσοφία πίσω από τη δημιουργία του **"Sport90"** εστιάζει στην ανάγκη του σύγχρονου φιλάθλου για μια ενοποιημένη και ταχύτατη πηγή πληροφόρησης, η οποία

λειτουργεί ως ένας ψηφιακός **"κόμβος στατιστικών" (Statistics Hub)**. Η ιδέα γεννήθηκε από την παρατήρηση ότι οι περισσότερες υπάρχουσες πλατφόρμες είτε είναι υπερβολικά φορτωμένες με διαφημίσεις και περιττή πληροφορία, είτε υστερούν σε ταχύτητα απόκρισης κατά τη μετάβαση από τα γενικά στατιστικά πρωταθλημάτων στα εξειδικευμένα δεδομένα των συλλόγων.

Σε αντίθεση με τους παραδοσιακούς στατικούς ιστότοπους, το "Sport90" σχεδιάστηκε ως ένα **δυναμικό εργαλείο** το οποίο "ζει" και ανανεώνεται μέσω της συνεχούς ροής δεδομένων. Η εφαρμογή αξιοποιεί την αρχιτεκτονική των Web Services για να αντλεί **ζωντανά δεδομένα (live data)** απευθείας από το API του Football-Data.org. Αυτό επιτρέπει στον χρήστη να εξερευνά την τρέχουσα αγωνιστική εικόνα των κορυφαίων ευρωπαϊκών διοργανώσεων — όπως η Premier League, η La Liga και η Serie A— με την εγγύηση ότι η πληροφορία που βλέπει είναι πάντα επίκαιρη και ακριβής.

Η εμπειρία του χρήστη (User Experience) βρίσκεται στον πυρήνα της ιδέας, προσφέροντας μια **αδιάλειπτη ροή ενημέρωσης**. Μέσα από ένα διαισθητικό σύστημα πλοήγησης, ο χρήστης μπορεί να ξεκινήσει από την αρχική σελίδα με τις γενικές κατατάξεις και να "καταδυθεί" (drill-down) σε βάθος πληροφορίας που περιλαμβάνει:

- Αναλυτικούς πίνακες με στατιστικά επίθεσης και άμυνας.
- Λεπτομερή προφίλ ομάδων με το τρέχον ρόστερ και την ηλικιακή κατανομή των παικτών.
- Πρόσφατα αποτελέσματα και μελλοντικό πρόγραμμα αγώνων, φιλτραρισμένα ώστε να αναδεικνύουν την τρέχουσα αγωνιστική κατάσταση.

Τέλος, η ιδέα συμπληρώνεται από την ανάγκη για **υψηλή ταχύτητα απόκρισης**. Χρησιμοποιώντας τις σύγχρονες δυνατότητες της React, η εφαρμογή ελαχιστοποιεί την αίσθηση της αναμονής, καθώς μόνο τα απαραίτητα τμήματα της σελίδας ενημερώνονται κάθε φορά. Η ενσωμάτωση του Node.js Proxy Server λειτουργεί ως ο "αθέατος" κινητήρας που διασφαλίζει ότι η επικοινωνία με τις πηγές των δεδομένων παραμένει σταθερή και ασφαλής, προσφέροντας μια επαγγελματική λύση στο πρόβλημα της διαχείρισης ζωντανής αθλητικής πληροφορίας.

1.3 Στόχοι και Κοινό

Ο πρωταρχικός στόχος της εφαρμογής **"Sport90"** είναι η ριζική **απλοποίηση της ποδοσφαιρικής πληροφορίας** μέσω μιας ορθολογικής οργάνωσης των δεδομένων. Σε ένα διαδίκτυο κατακλυσμένο από δαιδαλώδεις ιστότοπους που συχνά προκαλούν γνωστική

επιβάρυνση στον χρήστη, το "Sport90" προτείνει μια καθαρή, μινιμαλιστική και χρηστική διεπαφή (UI). Η στόχευση δεν περιορίζεται μόνο στην παροχή των δεδομένων, αλλά στην παροχή τους με τέτοιο τρόπο ώστε να είναι αναγνώσιμα και κατανοητά με την πρώτη ματιά, εξαλείφοντας τον περιττό θόρυβο και εστιάζοντας στην ουσία των στατιστικών.

Η εφαρμογή έχει σχεδιαστεί για να καλύπτει τις ανάγκες ενός ευρέος φάσματος χρηστών, δίνοντας έμφαση στις εξής κατηγορίες:

- **Φίλαθλοι και Οπαδοί**: Απευθύνεται σε χρήστες που αναζητούν άμεση και έγκυρη ενημέρωση για την πορεία της ομάδας τους, τις τρέχουσες βαθμολογίες των εγχώριων και διεθνών πρωταθλημάτων, καθώς και το πρόγραμμα των επόμενων αγωνιστικών. Η δυνατότητα γρήγορης πλοήγησης μεταξύ διαφορετικών λιγκών (π.χ. Premier League, La Liga) εξυπηρετεί τον χρήστη που θέλει μια σφαιρική εικόνα της αγωνιστικής δράσης σε ελάχιστο χρόνο.
- **Αναλυτές και Λάτρεις των Στατιστικών**: Για το πιο εξειδικευμένο κοινό, η εφαρμογή παρέχει βαθιά επίπεδα πληροφορίας, όπως αναλυτικά ρόστερ ομάδων με ηλικίες και εθνικότητες, καθώς και λεπτομερείς πίνακες με τους κορυφαίους σκόρερ, τις ασίστ και τα πέναλτι. Η οργάνωση των δεδομένων επιτρέπει την εξαγωγή συμπερασμάτων για τη φόρμα των παικτών και τη δυναμική των συλλόγων.
- **Χρήστες Φορητών Συσκευών**: Ένας από τους κρισιμότερους στόχους είναι η καθολική προσβασιμότητα μέσω του **αποκρινόμενου σχεδιασμού (Responsive Design)**. Η εφαρμογή υιοθετεί τις αρχές του "Mobile First", διασφαλίζοντας ότι οι σύνθετοι πίνακες και τα γραφικά προσαρμόζονται άψογα σε smartphones και tablets. Αυτό καθιστά το "Sport90" ένα απαραίτητο εργαλείο για τον χρήστη που θέλει να ενημερώνεται "εν κινήσει" (on the go), χωρίς εκπτώσεις στη λειτουργικότητα.

Επιπλέον, η εφαρμογή στοχεύει στην οικοδόμηση εμπιστοσύνης με τον χρήστη μέσω της διαφάνειας στη φόρτωση των δεδομένων. Η ενσωμάτωση μηχανισμών διαχείρισης σφαλμάτων δικτύου και ενημερωτικών μηνυμάτων κατά τη διάρκεια της άντλησης πληροφοριών, διασφαλίζει ότι ο χρήστης γνωρίζει ανά πάσα στιγμή την κατάσταση της εφαρμογής και την εγκυρότητα των προβαλλόμενων αποτελεσμάτων.

1.4 Βασικά Χαρακτηριστικά (Features)

Η αρχιτεκτονική της εφαρμογής **"Sport90"** δομείται γύρω από τέσσερις κεντρικούς πυλώνες λειτουργικότητας, οι οποίοι έχουν σχεδιαστεί για να προσφέρουν μια σφαιρική και λεπτομερή κάλυψη του ποδοσφαιρικού γίγνεσθαι. Κάθε πυλώνας εξυπηρετεί έναν

συγκεκριμένο σκοπό, διασφαλίζοντας ότι ο χρήστης λαμβάνει την πληροφορία που αναζητά με τη μέγιστη δυνατή ακρίβεια και ταχύτητα.

Ο πρώτος πυλώνας αφορά τους **Δυναμικούς Πίνακες Βαθμολογίας (Live Tables)**, οι οποίοι αποτελούν την "καρδιά" της αρχικής σελίδας και των σελίδων πρωταθλημάτων. Οι πίνακες αυτοί προσφέρουν μια πλήρη απεικόνιση της αγωνιστικής κατάταξης, ενσωματώνοντας αυτόματη επεξεργασία και προβολή κρίσιμων στοιχείων, όπως ο αριθμός των αγώνων, οι νίκες, οι ήττες, οι ισοπαλίες, καθώς και η διαφορά τερμάτων (GF/GA). Η χρήση χρωματικών κωδικοποιήσεων (visual cues) επιτρέπει στον χρήστη να αναγνωρίζει άμεσα τις θέσεις που οδηγούν στις ευρωπαϊκές διοργανώσεις ή στη ζώνη του υποβιβασμού.

Ο δεύτερος πυλώνας εστιάζει στις **Αναλυτικές Σελίδες Ομάδων (Team Pages)**, οι οποίες παρέχουν ένα βαθύτερο επίπεδο πληροφόρησης για κάθε σύλλογο. Μέσα από μια συγκεντρωτική παρουσίαση, ο χρήστης έχει πρόσβαση σε στοιχεία για τον προπονητή και το συμβόλαιό του, το πλήρες ρόστερ των παικτών με αναλυτικές πληροφορίες ηλικίας και εθνικότητας, καθώς και το σύνολο των διοργανώσεων (running competitions) στις οποίες συμμετέχει η ομάδα. Η λειτουργικότητα αυτή μετατρέπει την εφαρμογή από ένα απλό εργαλείο αποτελεσμάτων σε μια ολοκληρωμένη εγκυκλοπαίδεια ομάδων.

Ο τρίτος πυλώνας περιλαμβάνει το **Πρόγραμμα και τα Αποτελέσματα (Match Results)**, μια λειτουργία ζωτικής σημασίας για την παρακολούθηση της αγωνιστικής φόρμας. Η εφαρμογή καταγράφει με χρονολογική σειρά τα πρόσφατα σκορ, ενώ παράλληλα προβάλλει τις προσεχείς αναμετρήσεις της κάθε ομάδας. Ιδιαίτερη έμφαση δίνεται στην ορθή διαχείριση των ημερομηνιών και των ωρών διεξαγωγής, μετατρέποντας τα δεδομένα από τη μορφή UTC σε αναγνώσιμη τοπική ώρα, διευκολύνοντας έτσι τον προγραμματισμό του φιλάθλου.

Τέλος, ο τέταρτος πυλώνας είναι η **Ενότητα Ειδήσεων (News Section)**, η οποία λειτουργεί συμπληρωματικά προς τα στατιστικά δεδομένα. Πρόκειται για μια σελίδα ενημέρωσης με σύγχρονη διάταξη, όπου παρουσιάζονται σημαντικά γεγονότα, όπως βραβεύσεις παικτών (π.χ. Ballon d'Or) και αναλύσεις αγώνων. Η ενότητα αυτή ενισχύει τον χαρακτήρα της εφαρμογής ως ένα πλήρες δημοσιογραφικό και στατιστικό όργανο, προσφέροντας μια πλούσια εμπειρία περιήγησης που κρατά τον χρήστη ενήμερο για όλες τις εξελίξεις της επικαιρότητας.

2.Μελέτη & Επιλογή Τεχνολογιών

Η επιλογή του κατάλληλου τεχνολογικού στοίβαγματος (tech stack) αποτελεί το κρισιμότερο στάδιο για την επιτυχή υλοποίηση μιας σύγχρονης διαδικτυακής εφαρμογής. Στο κεφάλαιο αυτό, αναλύονται οι τεχνολογίες που επιλέχθηκαν για την ανάπτυξη του "Sport90", λαμβάνοντας υπόψη παραμέτρους όπως η ταχύτητα ανάπτυξης, η επεκτασιμότητα, η απόδοση και η ασφάλεια των δεδομένων. Η μελέτη επικεντρώνεται στον συνδυασμό εργαλείων που επιτρέπουν τη δημιουργία μιας δυναμικής διεπαφής χρήστη, η οποία υποστηρίζεται από έναν σταθερό μηχανισμό διαχείρισης εξωτερικών δεδομένων.

2.1. Frameworks & Βιβλιοθήκες

Η επιλογή των εργαλείων για την ανάπτυξη της διεπαφής δεν έγινε τυχαία, αλλά με γνώμονα τη δημιουργία ενός συστήματος που να είναι επεκτάσιμο, εύκολα συντηρήσιμο και εξαιρετικά αποδοτικό στη διαχείριση της κατάστασης (state management).

2.1.1 React: Αρχιτεκτονική Συστατικών Στοιχείων (Component-Based Architecture)

Η βιβλιοθήκη **React** επιλέχθηκε ως ο θεμέλιος λίθος για την ανάπτυξη του "Sport90", καθώς εισάγει μια επαναστατική προσέγγιση στη δόμηση διεπαφών μέσω της **αρχιτεκτονικής συστατικών στοιχείων (Component-Based Architecture)**. Σύμφωνα με αυτή τη μεθοδολογία, η διεπαφή χρήστη δεν αντιμετωπίζεται ως μια ενιαία, στατική σελίδα HTML, αλλά ως ένα ιεραρχικό δέντρο από αυτόνομες, αυτοπεριεχόμενες και επαναχρησιμοποιήσιμες μονάδες κώδικα, τα λεγόμενα **Components**. Αυτή η προσέγγιση επιτρέπει τον διαχωρισμό των ευθυνών (separation of concerns), όπου κάθε τμήμα της εφαρμογής —όπως το Header, το DataTable ή το TeamMatches— διαχειρίζεται τη δική του λογική, κατάσταση και εμφάνιση ανεξάρτητα από τα υπόλοιπα. Όπως αναφέρεται στη βιβλιογραφία, η ικανότητα της React να συνδυάζει αυτά τα μικρά στοιχεία για τη δημιουργία σύνθετων εφαρμογών ενισχύει δραστικά τη συντηρησιμότητα και την επεκτασιμότητα του κώδικα [1].

Στην υλοποίηση του "Sport90", η δύναμη αυτής της αρχιτεκτονικής αναδεικνύεται μέσω της χρήσης των **Props** και του **State**. Τα Components δέχονται δεδομένα εισόδου (Props), όπως η παράμετρος league στο component Ranking, επιτρέποντας στο ίδιο δομικό στοιχείο να εμφανίζει διαφορετικές πληροφορίες ανάλογα με το πλαίσιο χρήσης του. Παράλληλα, η εσωτερική κατάσταση (State), η οποία διαχειρίζεται μέσω του Hook useState, επιτρέπει στα στοιχεία να αντιδρούν δυναμικά στις αλλαγές των δεδομένων, όπως η φόρτωση των αποτελεσμάτων των αγώνων ή η εμφάνιση μηνυμάτων σφάλματος.

Κεντρικό ρόλο στην απόδοση της εφαρμογής παίζει το **Virtual DOM**, ένας μηχανισμός που επιτρέπει στη React να πραγματοποιεί ταχύτατες ενημερώσεις στο UI. Αντί να ανανεώνεται ολόκληρο το έγγραφο HTML κάθε φορά που αλλάζει μια πληροφορία, η React συγκρίνει την τρέχουσα κατάσταση με την προηγούμενη και εκτελεί τις ελάχιστες δυνατές αλλαγές στο πραγματικό DOM του browser. Αυτή η τεχνική είναι ιδιαίτερα κρίσιμη για το "Sport90", καθώς διασφαλίζει ότι οι δυναμικοί πίνακες βαθμολογίας και τα live στατιστικά εμφανίζονται χωρίς καθυστερήσεις, προσφέροντας μια ομαλή και άμεση εμπειρία πλοήγησης στον χρήστη.

Τέλος, ο κύκλος ζωής των συστατικών ελέγχεται μέσω του Hook useEffect, το οποίο επιτρέπει τον συγχρονισμό της εφαρμογής με το εξωτερικό API του διακομιστή μεσολάβησης (Proxy Server) ακριβώς τη στιγμή που το component εμφανίζεται στην οθόνη.

2.1.2Tailwind CSS: Utility-First Μεθοδολογία και Σχεδιαστική Συνέπεια

Για τον σχεδιασμό και τη μορφοποίηση της διεπαφής του "Sport90", επιλέχθηκε το **Tailwind CSS**, ένα σύγχρονο framework που βασίζεται στη μεθοδολογία **utility-first**. Σε αντίθεση με τα παραδοσιακά CSS frameworks που παρέχουν προκατασκευασμένα συστατικά (όπως κουμπιά ή κάρτες), το Tailwind προσφέρει μια εκτενή συλλογή από χαμηλού επιπέδου κλάσεις χρησιμότητας (utility classes). Αυτές οι κλάσεις —όπως flex, p-4, shadow και bg-gradient-to-r— επιτρέπουν την εφαρμογή στυλ απευθείας μέσα στη δομή της React, διευκολύνοντας την ταχεία πρωτοτυποποίηση και διασφαλίζοντας ότι η σχεδιαστική συνέπεια παραμένει αμετάβλητη σε ολόκληρη την εφαρμογή. Όπως υπογραμμίζεται στη βιβλιογραφία, η utility-first προσέγγιση επιτρέπει στους προγραμματιστές να χτίζουν προσαρμοσμένα interfaces χωρίς να εγκαταλείπουν το έγγραφο HTML/JSX, αποφεύγοντας έτσι το φαινόμενο του υπερβολικού και δυσσυντήρητου κώδικα CSS [2].

Η χρήση του Tailwind CSS στο "Sport90" είναι καθοριστική για την επίτευξη ενός **αποκρινόμενου σχεδιασμού (Responsive Design)**. Μέσω των ενσωματωμένων modifiers (όπως lg:, md:, sm:), η εφαρμογή προσαρμόζει δυναμικά τη διάταξη των στοιχείων της ανάλογα με το μέγεθος της οθόνης. Για παράδειγμα, οι σύνθετοι πίνακες βαθμολογίας και τα δεδομένα των ομάδων μετασχηματίζονται από πολυδιάστατα πλέγματα σε desktop υπολογιστές σε απλές, ευανάγνωστες λίστες σε κινητές συσκευές. Αυτή η ευελιξία είναι απαραίτητη για μια αθλητική εφαρμογή, καθώς ο χρήστης πρέπει να έχει πρόσβαση στην πληροφορία με την ίδια ευκολία είτε βρίσκεται στο σπίτι του είτε "εν κινήσει".

Επιπλέον, το Tailwind CSS συνέβαλε στη δημιουργία μιας ισχυρής **οπτικής ταυτότητας** για το project. Η χρήση διαβαθμίσεων (gradients) από το μπλε στο μαύρο (from-blue-700 to-blue-950) προσδίδει έναν επαγγελματικό και στιβαρό χαρακτήρα που ταιριάζει στον κόσμο του αθλητισμού, ενώ οι κλάσεις για σκιές (shadow) και στρογγυλεμένες γωνίες (rounded-xl) ενισχύουν τη μοντέρνα αισθητική της εφαρμογής. Η σταθερή πλοήγηση (fixed header) και η ομοιόμορφη τυπογραφία σε όλα τα components διασφαλίζουν ότι ο χρήστης παραμένει προσανατολισμένος, ανεξάρτητα από το βάθος της πληροφορίας που εξερευνά. Τέλος, το γεγονός ότι το Tailwind παράγει μόνο τον απαραίτητο κώδικα CSS κατά τη διαδικασία του build (PurgeCSS), βελτιώνει σημαντικά τους χρόνους φόρτωσης της εφαρμογής, κάτι που αποτελεί βασικό ζητούμενο για το "Sport90".

2.2Διαχείριση Δεδομένων

Η αποτελεσματική διαχείριση των δεδομένων αποτελεί τη ραχοκοκαλιά της εφαρμογής, καθώς το περιεχόμενο είναι πλήρως δυναμικό και εξαρτάται από εξωτερικές πηγές πληροφόρησης.

2.2.1Football-Data.org API: Η Πηγή των Δεδομένων

Η κύρια πηγή από την οποία το "Sport90" αντλεί την πληροφορία του είναι το **Football-Data.org API**, μια εξειδικευμένη υπηρεσία παροχής ποδοσφαιρικών δεδομένων που βασίζεται στην αρχιτεκτονική **REST (Representational State Transfer)**. Η χρήση ενός τέτοιου API είναι θεμελιώδης για την εφαρμογή, καθώς επιτρέπει την πρόσβαση σε έναν τεράστιο όγκο πληροφοριών χωρίς την ανάγκη διατήρησης μιας τοπικής, χειροκίνητα ενημερωμένης βάσης δεδομένων. Η RESTful φύση της υπηρεσίας σημαίνει ότι η επικοινωνία γίνεται μέσω τυποποιημένων αιτημάτων HTTP, επιτρέποντας στο "Sport90" να ανακτά συγκεκριμένους πόρους (resources) χρησιμοποιώντας μοναδικές διαδρομές URL για κάθε πρωτάθλημα ή ομάδα. Όπως επισημαίνεται στη βιβλιογραφία, τα REST APIs αποτελούν το πρότυπο για τη σύγχρονη ανταλλαγή δεδομένων στον ιστό, καθώς προσφέρουν ευελιξία, επεκτασιμότητα και ανεξαρτησία μεταξύ του παρόχου των δεδομένων και της εφαρμογής που τα καταναλώνει [3].

Το **πεδίο δεδομένων** που καλύπτει το API είναι ιδιαίτερα ευρύ, προσφέροντας στην εφαρμογή τη δυνατότητα να παρουσιάζει μια ολοκληρωμένη εικόνα του ευρωπαϊκού ποδοσφαίρου. Συγκεκριμένα, το API παρέχει σε πραγματικό χρόνο τις βαθμολογίες των πρωταθλημάτων (standings), αναλυτικά στοιχεία για κάθε σύλλογο (crest, coach, stadium), πλήρη ρόστερ παικτών με λεπτομέρειες όπως η ηλικία και η εθνικότητα, καθώς και το πλήρες ιστορικό και πρόγραμμα αγώνων (matches). Αυτή η πολυεπίπεδη πληροφορία επιτρέπει στο "Sport90" να λειτουργεί ως ένας "κόμβος στατιστικών", προσφέροντας στον χρήστη από τη γενική εικόνα της βαθμολογίας μέχρι την εξειδικευμένη ανάλυση μιας συγκεκριμένης ομάδας.

Η **δομή απόκρισης** των δεδομένων είναι σε μορφή **JSON (JavaScript Object Notation)**, η οποία θεωρείται η ιδανική επιλογή για εφαρμογές που βασίζονται στη JavaScript και τη React. Η ιεραρχική δομή του JSON επιτρέπει την άμεση "χαρτογράφηση" (mapping) των δεδομένων στα components της React, διευκολύνοντας τη δυναμική παραγωγή στοιχείων UI, όπως οι γραμμές ενός πίνακα βαθμολογίας ή οι κάρτες των παικτών. Τέλος, η διαδικασία της **αυθεντικοποίησης** διασφαλίζει την ελεγχόμενη πρόσβαση στις πληροφορίες. Κάθε αίτημα που αποστέλλεται από τον Proxy Server περιλαμβάνει στις κεφαλίδες του (HTTP Headers) ένα μοναδικό κλειδί ασφαλείας, το **API Token** (X-Auth-Token), το οποίο επαληθεύει την ταυτότητα της εφαρμογής και επιτρέπει την επικοινωνία με τον εξυπηρετητή των δεδομένων.

2.2.2Node.js Proxy Server: Επίλυση CORS και Ασφάλεια

Ένα από τα πιο κρίσιμα τεχνικά τμήματα της παρούσας εργασίας είναι η υλοποίηση ενός **Proxy Server** (διακομιστή μεσολάβησης) με τη χρήση του περιβάλλοντος εκτέλεσης **Node.js** και του web framework **Express**. Ο διακομιστής αυτός λειτουργεί ως ένας απαραίτητος ενδιάμεσος μηχανισμός (**middleware**) μεταξύ της React εφαρμογής (Client) και του εξωτερικού API του Football-Data.org. Σύμφωνα με τη βιβλιογραφία, η χρήση ενός διακομιστή μεσολάβησης σε σύγχρονες web εφαρμογές προσφέρει ένα επιπλέον επίπεδο ελέγχου και ασφάλειας, επιτρέποντας στον προγραμματιστή να διαχειρίζεται τα αιτήματα μακριά από το περιβάλλον του προγράμματος περιήγησης, το οποίο είναι εκτεθειμένο στον τελικό χρήστη [4].

Η κύρια ανάγκη υλοποίησης αυτού του διακομιστή προέκυψε για την **επίλυση των προβλημάτων CORS (Cross-Origin Resource Sharing)**. Τα σύγχρονα προγράμματα περιήγησης εφαρμόζουν αυστηρούς κανόνες ασφαλείας που εμποδίζουν μια εφαρμογή (π.χ. το frontend του Sport90 που τρέχει στο localhost:5173) να ζητήσει δεδομένα απευθείας από ένα διαφορετικό domain (π.χ. το api.football-data.org). Ο Proxy Server παρακάμπτει αυτόν τον περιορισμό εκτελώντας τα αιτήματα σε περιβάλλον διακομιστή (**server-side**), όπου οι περιορισμοί CORS δεν υφίστανται, και στη συνέχεια μεταβιβάζει την πληροφορία στο frontend μέσω του πακέτου cors της Node.js.

Παράλληλα, ο Proxy Server διασφαλίζει την **ασφάλεια του API Key** της εφαρμογής. Σε μια τυπική client-side υλοποίηση, το ευαίσθητο κλειδί αυθεντικοποίησης θα ήταν ορατό σε οποιονδήποτε χρήστη χρησιμοποιούσε τα εργαλεία ανάπτυξης του browser (DevTools). Στην περίπτωση του Sport90, το κλειδί αποθηκεύεται και χρησιμοποιείται αποκλειστικά από τον Proxy Server στο αρχείο proxy.js, καθιστώντας το απρόσιτο για τρίτους. Επιπλέον, ο server έχει παραμετροποιηθεί για **Δυναμική Δρομολόγηση**, χρησιμοποιώντας παραμέτρους στο URL (req.params) για να αναγνωρίζει αν ο χρήστης ζητά δεδομένα πρωταθλήματος (league), ομάδας (team), αγώνων (matches) ή σκόρερ (scorers), κατασκευάζοντας αυτόματα το κατάλληλο URL επικοινωνίας με την πηγή.

Τέλος, ο Proxy Server παίζει καθοριστικό ρόλο στη **Διαχείριση Σφαλμάτων**. Μέσω δομών try-catch και του axios, ο διακομιστής παρακολουθεί την απόκριση του εξωτερικού API. Εάν παρουσιαστεί πρόβλημα —όπως υπέρβαση του ορίου αιτημάτων ή προσωρινή διακοπή της υπηρεσίας— ο Proxy Server συλλαμβάνει το σφάλμα και επιστρέφει έναν κατάλληλο κωδικό κατάστασης HTTP (π.χ. 500 Internal Server Error) μαζί με ένα μήνυμα σφάλματος σε μορφή JSON. Αυτό επιτρέπει στο frontend της React να διαχειριστεί την κατάσταση με επαγγελματικό τρόπο, εμφανίζοντας στον χρήστη ενημερωτικά μηνύματα ("Something went wrong") αντί να καταρρεύσει η εφαρμογή.

2.3Εργαλεία Ανάπτυξης

Η επιλογή σύγχρονων εργαλείων ανάπτυξης αποτέλεσε καθοριστικό παράγοντα για την υλοποίηση του "Sport90", καθώς επέτρεψε τη βελτιστοποίηση της ροής εργασίας και τη διατήρηση υψηλών προτύπων στην ποιότητα του πηγαίου κώδικα. Σε ένα οικοσύστημα όπως αυτό της JavaScript, όπου οι απαιτήσεις για ταχύτητα και αξιοπιστία αυξάνονται συνεχώς, η χρήση αυτοματοποιημένων εργαλείων για τη διαχείριση, τον έλεγχο και τη δόμηση της εφαρμογής είναι απαραίτητη για τη διασφάλιση της επεκτασιμότητας του project.

Η στρατηγική ανάπτυξης βασίστηκε σε τρεις κεντρικούς άξονες: την ταχύτητα στην παραγωγή αποτελέσματος μέσω του **Vite**, τη διασφάλιση της συντακτικής ορθότητας μέσω του **ESLint**, και την αυστηρή οργάνωση των εξαρτήσεων μέσω του **NPM**. Τα εργαλεία αυτά δεν λειτουργούν μεμονωμένα, αλλά συνθέτουν ένα ενιαίο περιβάλλον που προστατεύει την εφαρμογή από σφάλματα, βελτιστοποιεί το μέγεθος των αρχείων για τον τελικό χρήστη και επιτρέπει τη γρήγορη ενσωμάτωση νέων λειτουργιών.

2.3.1Vite: Το Εργαλείο Κατασκευής (Build Tool)

Η επιλογή του **Vite** ως του κύριου εργαλείου κατασκευής και ανάπτυξης για το "Sport90" αποτέλεσε στρατηγική απόφαση, καθώς προσφέρει μια νέα γενιά frontend tooling που εστιάζει στην ταχύτητα και την αποδοτικότητα. Σε αντίθεση με παλαιότερα εργαλεία όπως το Webpack, το Vite επαναπροσδιορίζει τον τρόπο με τον οποίο ο κώδικας σερβίρεται κατά τη διάρκεια της ανάπτυξης και πώς βελτιστοποιείται για την παραγωγή. Αυτό επιτυγχάνεται μέσω μιας αρχιτεκτονικής που εκμεταλλεύεται τις σύγχρονες δυνατότητες των browsers, μειώνοντας δραστικά τους χρόνους αναμονής για τον προγραμματιστή.

Ένα από τα κύρια πλεονεκτήματα του Vite είναι η **ασύλληπτη ταχύτητα ανάπτυξης** που προσφέρει. Χρησιμοποιώντας την τεχνολογία **Native ESM (ES Modules)**, το Vite δεν χρειάζεται να κάνει bundle ολόκληρη την εφαρμογή πριν ξεκινήσει ο διακομιστής ανάπτυξης· αντίθετα, σερβίρει τον κώδικα κατευθείαν στον browser, ο οποίος αναλαμβάνει το "δέσιμο" των αρχείων. Αυτό επιτρέπει το σχεδόν ακαριαίο **Hot Module Replacement (HMR)**, όπου οι αλλαγές που πραγματοποιούνται στον κώδικα —για παράδειγμα, μια αλλαγή στο styling ενός πίνακα στο dataTables.jsx— εμφανίζονται σε πραγματικό χρόνο χωρίς να χάνεται η κατάσταση (state) της εφαρμογής [5].

Όσον αφορά τη **βελτιστοποίηση κατά τη διαδικασία του build**, το Vite χρησιμοποιεί το πανίσχυρο εργαλείο **Rollup** για τη δημιουργία της τελικής έκδοσης της εφαρμογής. Κατά την εκτέλεση της εντολής npm run build, το Vite εκτελεί σύνθετες διεργασίες όπως το "tree-shaking" (αφαίρεση κώδικα που δεν χρησιμοποιείται), το "code-splitting" (διάσπαση κώδικα σε μικρότερα τμήματα) και τη συμπίεση των αρχείων JavaScript και CSS. Αυτή η διαδικασία διασφαλίζει ότι το "Sport90" θα έχει το ελάχιστο δυνατό μέγεθος, βελτιστοποιώντας τους χρόνους φόρτωσης για τον τελικό χρήστη και βελτιώνοντας τη συνολική εμπειρία χρήσης (UX).

Τέλος, η **πλήρης υποστήριξη για React και Tailwind CSS** αποτελεί κεντρικό πυλώνα της παραμετροποίησης του εργαλείου. Μέσα από το αρχείο vite.config.js, έχουν ενσωματωθεί τα απαραίτητα plugins (@vitejs/plugin-react και @tailwindcss/vite) που επιτρέπουν στο Vite να αναγνωρίζει και να επεξεργάζεται τη σύνταξη JSX της React καθώς και τις utility classes του Tailwind. Αυτή η στενή ενσωμάτωση διασφαλίζει ότι όλα τα δομικά στοιχεία της εφαρμογής, από το routing του App.jsx έως το responsive styling των CSS αρχείων, συνεργάζονται αρμονικά σε ένα ενιαίο και εξαιρετικά αποδοτικό περιβάλλον εκτέλεσης.

2.3.2ESLint: Διασφάλιση Ποιότητας Κώδικα

Η ποιότητα και η συνεκτικότητα του πηγαίου κώδικα αποτελούν θεμελιώδη χαρακτηριστικά για τη μακροπρόθεσμη σταθερότητα μιας εφαρμογής, και στο project "Sport90" αυτά διασφαλίζονται μέσω του **ESLint**. Το ESLint είναι ένα προηγμένο εργαλείο στατικής ανάλυσης κώδικα (linter) που εξετάζει το αρχείο χωρίς να το εκτελέσει, εντοπίζοντας μοτίβα που δεν συμμορφώνονται με τους προκαθορισμένους κανόνες προγραμματισμού. Η ενσωμάτωσή του στη ροή εργασίας επιτρέπει στον προγραμματιστή να διορθώνει αστοχίες σε πραγματικό χρόνο, αποφεύγοντας τη συσσώρευση "τεχνικού χρέους" (technical debt) και διατηρώντας τον κώδικα ευανάγνωστο και ομοιόμορφο.

Ένα από τα κύρια οφέλη του ESLint είναι η **αυτοματοποιημένη ανίχνευση σφαλμάτων** και κακών πρακτικών πριν η εφαρμογή φτάσει στο στάδιο της εκτέλεσης. Μέσω του αρχείου ρυθμίσεων eslint.config.js, το εργαλείο έχει παραμετροποιηθεί ώστε να εντοπίζει κοινά προγραμματιστικά λάθη, όπως η χρήση μεταβλητών που δεν έχουν οριστεί ή η ύπαρξη μεταβλητών που ορίστηκαν αλλά δεν χρησιμοποιήθηκαν ποτέ (no-unused-vars). Αυτή η προληπτική προσέγγιση είναι κρίσιμη για τη διατήρηση ενός "καθαρού" περιβάλλοντος, ειδικά σε αρχεία με σύνθετη λογική όπως το teamMatches.jsx ή το proxy.js, όπου η διαχείριση των δεδομένων απαιτεί υψηλή ακρίβεια.

Ιδιαίτερη έμφαση έχει δοθεί στην επιβολή των **κανόνων της React**, οι οποίοι είναι ενσωματωμένοι στο project μέσω εξειδικευμένων plugins όπως το eslint-plugin-react-hooks. Οι κανόνες αυτοί διασφαλίζουν την ορθή χρήση των React Hooks, όπως το useState και το useEffect, αποτρέποντας κλασικά προβλήματα στον κύκλο ζωής των components. Για παράδειγμα, το ESLint προειδοποιεί εάν λείπει κάποια εξάρτηση (dependency) από τον πίνακα ελέγχου του useEffect, μια παράλειψη που θα μπορούσε να οδηγήσει σε ατέρμονα loops ή λανθασμένη ανάκτηση δεδομένων από το API [6].

Τέλος, η συστηματική χρήση του ESLint συμβάλλει καθοριστικά στη **συντήρηση και τη μελλοντική επέκταση** της εφαρμογής. Καθώς το "Sport90" μεγαλώνει, η ύπαρξη ενός κοινού κώδικα κανόνων διασφαλίζει ότι νέα components ή σελίδες θα ακολουθούν την ίδια λογική και σύνταξη με τα υπάρχοντα. Αυτό διευκολύνει τη διορθωση σφαλμάτων (debugging), καθώς ο κώδικας είναι προβλέψιμος και απαλλαγμένος από συντακτικές ασυνέπειες, επιτρέποντας στον προγραμματιστή να εστιάσει στην υλοποίηση νέων λειτουργιών αντί για τον καθαρισμό παλαιότερων αστοχιών.

2.3.3Διαχείριση Εκδόσεων και Οργάνωση Project

Η οργάνωση του **"Sport90"** ακολουθεί τα διεθνή πρότυπα ανάπτυξης λογισμικού, διασφαλίζοντας τη σωστή διαχείριση των πόρων, των εξαρτήσεων και των εκδόσεων της εφαρμογής. Η υιοθέτηση μιας δομημένης προσέγγισης στην οργάνωση των αρχείων και των εργαλείων διαχείρισης είναι απαραίτητη για τη διατήρηση της ακεραιότητας του κώδικα, ειδικά σε περιβάλλοντα όπου χρησιμοποιούνται πολλαπλές εξωτερικές βιβλιοθήκες και εργαλεία αυτοματισμού.

Κεντρικό ρόλο σε αυτή τη διαδικασία παίζει ο **NPM (Node Package Manager)**, ο οποίος χρησιμοποιείται ως το κύριο σύστημα διαχείρισης όλων των εξωτερικών βιβλιοθηκών (dependencies) της εφαρμογής. Μέσω του NPM, εγκαταστάθηκαν και ελέγχονται κρίσιμα πακέτα όπως το **Axios** για τη διενέργεια των αιτημάτων HTTP προς τον Proxy Server, το **React Router DOM** για τη διαχείριση της πλοήγησης και το **FontAwesome** για τον οπτικό εμπλουτισμό του UI. Η αυτοματοποιημένη διαχείριση μέσω του NPM επιτρέπει την εύκολη ενημέρωση των βιβλιοθηκών και τη διασφάλιση ότι όλοι οι προγραμματιστές που εργάζονται στο project χρησιμοποιούν τις ίδιες ακριβώς εκδόσεις εργαλείων.

Η "ταυτότητα" και η καρδιά του project εντοπίζεται στο αρχείο **package.json**. Το αρχείο αυτό λειτουργεί ως ένα κεντρικό μητρώο, καταγράφοντας όχι μόνο το όνομα και την έκδοση της εφαρμογής ("sport90", v0.0.0), αλλά και το πλήρες ιστορικό των εξαρτήσεων (dependencies) και των εργαλείων ανάπτυξης (devDependencies). Επιπλέον, περιλαμβάνει τα **scripts**, τα

οποία είναι προκαθορισμένα σενάρια εντολών που απλοποιούν τις καθημερινές εργασίες: η εντολή npm run dev εκκινεί το τοπικό περιβάλλον ανάπτυξης μέσω του Vite, ενώ η εντολή npm run build προετοιμάζει την εφαρμογή για την τελική της δημοσίευση, βελτιστοποιώντας τον κώδικα για μέγιστη απόδοση.

Τέλος, η διαχείριση του πηγαίου κώδικα ενισχύεται από τη χρήση του αρχείου **.gitignore**. Σε ένα επαγγελματικό περιβάλλον ανάπτυξης, είναι κρίσιμο να διαχωρίζεται ο πηγαίος κώδικας από τα αυτόματα παραγόμενα αρχεία ή τις τοπικές ρυθμίσεις. Το αρχείο .gitignore διασφαλίζει ότι ογκώδεις φάκελοι όπως ο node_modules (ο οποίος περιέχει χιλιάδες αρχεία από τις εγκατεστημένες βιβλιοθήκες) ή προσωρινά αρχεία συστήματος δεν περιλαμβάνονται στο αποθετήριο του κώδικα. Αυτή η πρακτική διατηρεί το project "ελαφρύ" και καθαρό, επιτρέποντας τη γρήγορη μεταφορά και τον εύκολο έλεγχο των εκδόσεων (version control), ενώ παράλληλα αποτρέπει την άσκοπη κατανάλωση αποθηκευτικού χώρου και πόρων δικτύου.

3.Μεθοδολογία & Σχεδιασμός Συστήματος

Η επιτυχία μιας web εφαρμογής που βασίζεται σε δεδομένα πραγματικού χρόνου εξαρτάται άμεσα από τη στιβαρότητα της αρχιτεκτονικής της και τον προσεκτικό σχεδιασμό των αλληλεπιδράσεών της. Στο κεφάλαιο αυτό, αναλύεται η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη του "Sport90", εστιάζοντας στον τρόπο με τον οποίο τα διαφορετικά επίπεδα του συστήματος επικοινωνούν μεταξύ τους για να προσφέρουν μια απρόσκοπτη εμπειρία χρήστη. Μέσω του αρχιτεκτονικού σχεδιασμού και της χρήσης προτύπων UML, αποσαφηνίζεται η ροή της πληροφορίας και οι λειτουργικές απαιτήσεις που διέπουν το σύστημα.

3.1Αρχιτεκτονική Εφαρμογής: Η επικοινωνία Client-Server-API

Η αρχιτεκτονική του **"Sport90"** βασίζεται σε ένα μοντέλο τριών επιπέδων (3-tier architecture), το οποίο έχει σχεδιαστεί για να διασφαλίζει τη μέγιστη δυνατή απόδοση, επεκτασιμότητα και ασφάλεια κατά τη διαχείριση αθλητικών δεδομένων σε πραγματικό χρόνο. Η επιλογή αυτής της δομής επιτρέπει τον πλήρη διαχωρισμό των αρμοδιοτήτων (Separation of Concerns), όπου κάθε επίπεδο επιτελεί έναν εξειδικευμένο ρόλο: το frontend επικεντρώνεται στην παρουσίαση και την αλληλεπίδραση, ο Proxy Server στη διαμεσολάβηση και την ασφάλεια, και το εξωτερικό API στην παροχή της πρωτογενούς πληροφορίας.

Η ροή της πληροφορίας ξεκινά από το αίτημα του χρήστη στη διεπαφή και ολοκληρώνεται με την επιστροφή των δεδομένων JSON, περνώντας μέσα από έναν ελεγχόμενο δίαυλο επικοινωνίας που ελαχιστοποιεί την έκθεση ευαίσθητων δεδομένων και βελτιστοποιεί την ταχύτητα απόκρισης. Η συγκεκριμένη αρχιτεκτονική προσέγγιση είναι απαραίτητη για την αντιμετώπιση προκλήσεων όπως οι περιορισμοί ασφαλείας των περιηγητών (CORS) και η προστασία των κλειδιών πρόσβασης (API Tokens).

3.1.1Το Επίπεδο της Διεπαφής (Client Side)

Το επίπεδο της διεπαφής (Frontend) της εφαρμογής "Sport90" αποτελεί το κρίσιμο σημείο επαφής μεταξύ του τελικού χρήστη και των δεδομένων, και έχει υλοποιηθεί εξολοκλήρου με τη χρήση της βιβλιοθήκης **React**. Η επιλογή αυτή επιτρέπει στην εφαρμογή να λειτουργεί ως μια Μονοσελιδική Εφαρμογή (Single Page Application - SPA), όπου η πλοήγηση και η αλληλεπίδραση πραγματοποιούνται χωρίς την ανάγκη ανανέωσης του φυλλομετρητή, προσφέροντας μια εξαιρετικά ομαλή εμπειρία χρήσης. Το Frontend δεν περιορίζεται μόνο στην οπτικοποίηση, αλλά αναλαμβάνει την πλήρη διαχείριση του κύκλου ζωής των αιτημάτων και τον συγχρονισμό του UI με την τρέχουσα κατάσταση των δεδομένων.

Η **υποβολή των αιτημάτων** αποτελεί την πρώτη βασική λειτουργία αυτού του επιπέδου. Η εφαρμογή αξιοποιεί τη βιβλιοθήκη **Axios** για τη διενέργεια ασύγχρονων αιτημάτων HTTP προς τον τοπικό Proxy Server (localhost:5000). Η ασύγχρονη φύση αυτών των αιτημάτων, που υλοποιείται μέσω της σύνταξης async/await, είναι καθοριστική, καθώς επιτρέπει στη ροή της πλοήγησης να παραμένει ενεργή. Έτσι, ενώ τα δεδομένα ανακτώνται στο παρασκήνιο, ο χρήστης μπορεί να συνεχίσει να αλληλεπιδρά με το μενού ή άλλα τμήματα της σελίδας, ενώ

παράλληλα ενημερώνεται για την εξέλιξη της διαδικασίας μέσω ενδείξεων φόρτωσης (loading indicators).

Μόλις ολοκληρωθεί η λήψη, ξεκινά η φάση της **επεξεργασίας και προβολής**. Τα δεδομένα που επιστρέφονται από τον Proxy Server σε μορφή JSON υφίστανται άμεσο "parsing" από τα εξειδικευμένα React Components. Για παράδειγμα, το component DataTable επεξεργάζεται τη δομή data.standings[0].table για να παράγει δυναμικά τις γραμμές του πίνακα βαθμολογίας, ενώ το SquadTeam μετατρέπει τη λίστα παικτών σε μια δομημένη παρουσίαση με ονόματα, ηλικίες και εθνικότητες. Η δυναμική αυτή δημιουργία στοιχείων UI διασφαλίζει ότι η εφαρμογή μπορεί να προβάλει οποιοδήποτε πρωτάθλημα ή ομάδα υποστηρίζεται από το API, χωρίς να απαιτείται αλλαγή στη δομή του κώδικα.

Τέλος, η **διαχείριση της κατάστασης (State Management)** μέσω των React Hooks διασφαλίζει την αποδοτικότητα του συστήματος. Χρησιμοποιώντας το useState, το Frontend διατηρεί μια "στιγμιαία εικόνα" των δεδομένων στην τοπική μνήμη του browser. Η React παρακολουθεί συνεχώς αυτή την κατάσταση και προβαίνει σε επανασχεδιασμό (re-render) της οθόνης μόνο όταν εντοπίσει πραγματικές αλλαγές στα δεδομένα. Αυτή η προσέγγιση μειώνει δραστικά την κατανάλωση υπολογιστικών πόρων και την άσκοπη επεξεργασία του DOM, εξασφαλίζοντας ότι η προβολή των αποτελεσμάτων και των στατιστικών γίνεται με τη μέγιστη δυνατή ταχύτητα και ακρίβεια.

3.1.2Ο Διακομιστής Μεσολάβησης (Server Side Proxy)

Ο **Proxy Server** του "Sport90", ο οποίος έχει αναπτυχθεί με τη χρήση του περιβάλλοντος εκτέλεσης **Node.js** και του framework **Express**, αποτελεί τον "αόρατο" αλλά κρισιμότερο κρίκο στην αλυσίδα επικοινωνίας της εφαρμογής. Λειτουργώντας ως ενδιάμεσο λογισμικό (**middleware**), ο διακομιστής αυτός αναλαμβάνει να γεφυρώσει το χάσμα μεταξύ της διεπαφής χρήστη (Client) και της απομακρυσμένης πηγής δεδομένων (Football-Data API). Η ύπαρξή του είναι απαραίτητη όχι μόνο για τη λειτουργικότητα, αλλά και για τη συμμόρφωση με τα σύγχρονα πρότυπα ασφάλειας του διαδικτύου, μετατρέποντας τα απλά αιτήματα του frontend σε έγκυρες και ασφαλείς συναλλαγές δεδομένων.

Η λειτουργία της **ασφαλούς διαμεσολάβησης** είναι το πρώτο βασικό καθήκον του server. Σε μια τυπική ροή, ο διακομιστής δέχεται το αίτημα από την εφαρμογή React, η οποία στερείται της δυνατότητας να επικοινωνήσει απευθείας με το API λόγω της ανάγκης για κρυφή αυθεντικοποίηση. Ο Proxy Server "συμπληρώνει" το αίτημα ενσωματώνοντας στις κεφαλίδες (headers) το κρυφό κλειδί **X-Auth-Token**. Με αυτόν τον τρόπο, το ευαίσθητο API Key παραμένει απομονωμένο στο ελεγχόμενο περιβάλλον του διακομιστή και δεν εκτίθεται ποτέ στον κώδικα του frontend ή στην κίνηση του δικτύου που είναι ορατή από τον browser του τελικού χρήστη, αποτρέποντας έτσι πιθανές κακόβουλες χρήσεις.

Η **επίλυση των προβλημάτων CORS (Cross-Origin Resource Sharing)** αποτελεί τη δεύτερη κρίσιμη αποστολή του διακομιστή. Οι σύγχρονοι περιηγητές επιβάλλουν αυστηρούς περιορισμούς που απαγορεύουν σε μια ιστοσελίδα να αντλεί δεδομένα από ένα διαφορετικό domain για την αποφυγή επιθέσεων. Επειδή ο Proxy Server εκτελεί τα αιτήματα στην πλευρά του διακομιστή (**server-side**), δεν υπόκειται στους περιορισμούς αυτούς. Χρησιμοποιώντας το πακέτο cors της Node.js, ο server επιτρέπει ελεγχόμενα στη React εφαρμογή να λαμβάνει

τα δεδομένα που χρειάζεται, λειτουργώντας ως "νόμιμος εκπρόσωπος" που φέρνει την πληροφορία από το εξωτερικό API και την παραδίδει με ασφάλεια στο frontend.

Τέλος, ο διακομιστής προσφέρει ευελιξία μέσω της **δυναμικής δρομολόγησης (Dynamic Routing)**. Ο κώδικας του proxy.js είναι σχεδιασμένος με παραμετροποιημένες διαδρομές (endpoints), οι οποίες αναλύουν τις παραμέτρους του εισερχόμενου αιτήματος μέσω του αντικειμένου req.params. Είτε ο χρήστης επιθυμεί να δει τα αποτελέσματα μιας συγκεκριμένης λίγκας, είτε τα στατιστικά μιας συγκεκριμένης ομάδας, ο διακομιστής αναγνωρίζει την πρόθεση, κατασκευάζει δυναμικά το αντίστοιχο URL για το Football-Data API και επιστρέφει την ακριβή πληροφορία. Αυτή η αρχιτεκτονική καθιστά τον Proxy Server έναν κλιμακώσιμο μηχανισμό που μπορεί να εξυπηρετήσει πληθώρα διαφορετικών αιτημάτων χωρίς την ανάγκη για πολλαπλές, στατικές ρυθμίσεις.

3.1.3Η Εξωτερική Πηγή Δεδομένων (External API)

Το τρίτο και τελικό επίπεδο της αρχιτεκτονικής είναι το **Football-Data.org API**, το οποίο λειτουργεί ως η κεντρική δεξαμενή πληροφοριών και παρέχει τα πρωτογενή δεδομένα που απαιτούνται για τη λειτουργία της εφαρμογής **"Sport90"**. Πρόκειται για μια εξειδικευμένη Cloud υπηρεσία που συγκεντρώνει και οργανώνει ποδοσφαιρικά στατιστικά από τα σημαντικότερα πρωταθλήματα παγκοσμίως, εκθέτοντάς τα μέσω μιας διεπαφής προγραμματισμού εφαρμογών (API). Η αξιοπιστία και η ταχύτητα απόκρισης αυτού του επιπέδου είναι καθοριστικής σημασίας, καθώς αποτελεί τη μοναδική πηγή αλήθειας (source of truth) για τις βαθμολογίες, τα αποτελέσματα και τα ρόστερ που βλέπει ο τελικός χρήστης.

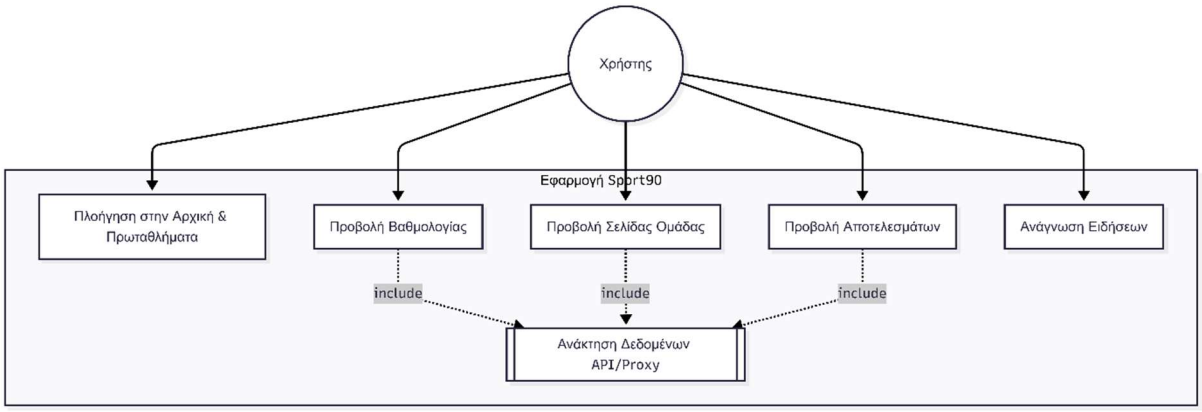
Η επικοινωνία σε αυτό το στάδιο είναι **αυστηρά δομημένη** και ακολουθεί το πρωτόκολλο HTTP/S, διασφαλίζοντας ότι η ανταλλαγή πληροφοριών γίνεται με ασφάλεια και τυποποίηση. Ο Proxy Server του "Sport90" αναλαμβάνει τον ρόλο του "πελάτη" για το API, κατασκευάζοντας δυναμικά URLs με βάση τις ανάγκες του frontend. Για παράδειγμα, όταν ένας χρήστης επιλέγει να δει τη βαθμολογία της Premier League, ο διακομιστής συνθέτει τη διαδρομή /v4/competitions/PL/standings, ενώ για τις λεπτομέρειες μιας ομάδας χρησιμοποιεί το endpoint /v4/teams/{id}. Αυτή η **δυναμική δόμηση των αιτημάτων** επιτρέπει στην εφαρμογή να ανακτά με χειρουργική ακρίβεια μόνο την πληροφορία που είναι απαραίτητη κάθε στιγμή, βελτιστοποιώντας τη χρήση του εύρους ζώνης (bandwidth) και μειώνοντας τον χρόνο φόρτωσης.

Τα δεδομένα που επιστρέφει το API είναι οργανωμένα σε **ιεραρχική μορφή JSON**, γεγονός που επιτρέπει τη βαθιά διασύνδεση των πληροφοριών. Κάθε απόκριση περιέχει μεταδεδομένα (όπως ημερομηνίες ενημέρωσης και στοιχεία διοργάνωσης) καθώς και τα κύρια δεδομένα (όπως πίνακες βαθμολογίας ή λίστες παικτών). Η δομή αυτή επιτρέπει στο "Sport90" να "διαβάζει" σύνθετες σχέσεις, όπως το ποιοι αγώνες ανήκουν σε ποια αγωνιστική ή ποιοι παίκτες ανήκουν σε μια συγκεκριμένη ομάδα, και να τις οπτικοποιεί με τρόπο που είναι εύκολα κατανοητός από τον φίλαθλο. Τέλος, το API επιβάλλει συγκεκριμένους κανόνες χρήσης (rate limits), τους οποίους ο Proxy Server διαχειρίζεται προσεκτικά, διασφαλίζοντας τη συνεχή και απρόσκοπτη παροχή δεδομένων στην εφαρμογή.

3.2Διαγράμματα UML

Η χρήση των διαγραμμάτων UML στο στάδιο του σχεδιασμού διασφαλίζει ότι όλες οι λειτουργικές απαιτήσεις της εφαρμογής έχουν προβλεφθεί και οργανωθεί σωστά πριν από την τελική υλοποίηση. Για το "Sport90", επιλέχθηκαν δύο τύποι διαγραμμάτων που καλύπτουν διαφορετικές πτυχές του συστήματος: το **Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram)**, που εστιάζει στην εξωτερική συμπεριφορά και τις δυνατότητες που παρέχονται στον χρήστη, και το **Διάγραμμα Δραστηριοτήτων (Activity Diagram)**, το οποίο αναλύει τη δυναμική ροή των εργασιών και των αιτημάτων στο παρασκήνιο.

3.2.1Use Case Diagram: Αλληλεπίδραση χρήστη με τις σελίδες και τα δεδομένα



Αναλυτική Περιγραφή και Σχολιασμός Διαγράμματος

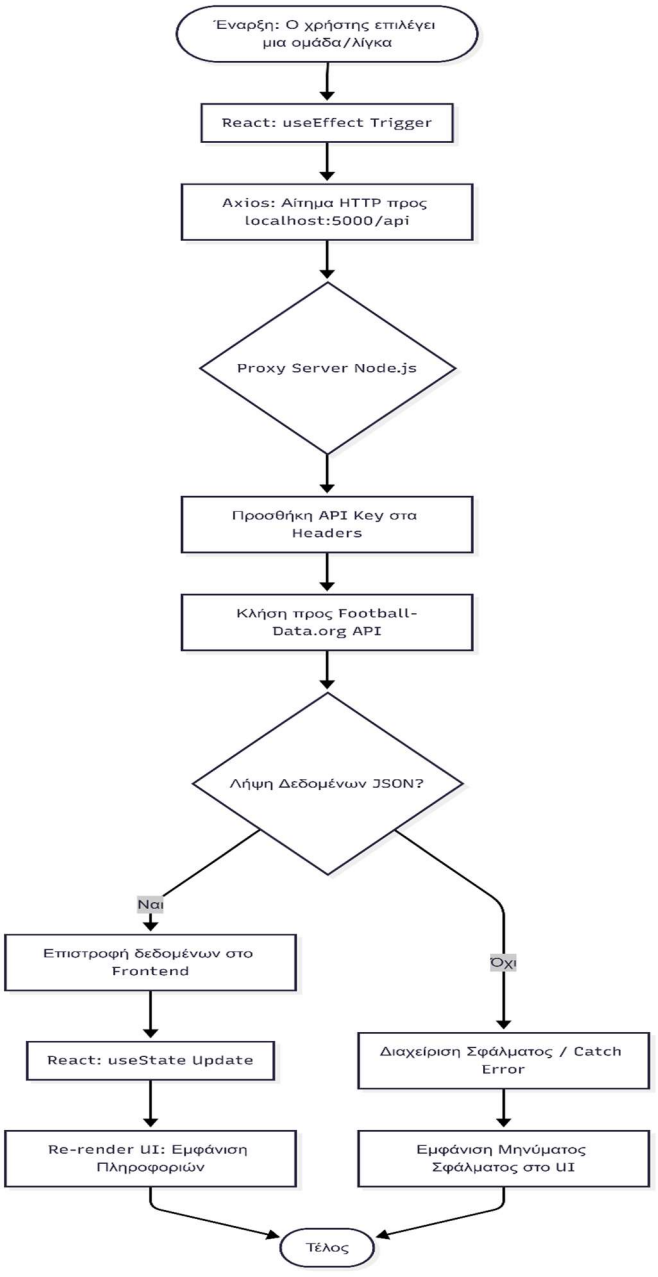
Το Διάγραμμα Περιπτώσεων Χρήσης αποτυπώνει τις λειτουργικές απαιτήσεις του συστήματος από τη σκοπιά του τελικού χρήστη, ο οποίος αποτελεί τον κύριο "Δρώντα" (Actor) της εφαρμογής **"Sport90"**.

- **Κεντρικές Λειτουργίες (Core Use Cases)**: Ο χρήστης αλληλεπιδρά με το σύστημα μέσω πέντε βασικών λειτουργιών. Η **Πλοήγηση** αποτελεί τη βάση της εμπειρίας, επιτρέποντας τη μετακίνηση μεταξύ των διαφορετικών ενοτήτων. Η **Προβολή Βαθμολογίας** και η **Προβολή Σελίδας Ομάδας** αποτελούν τις πιο σύνθετες περιπτώσεις χρήσης, καθώς απαιτούν τη δυναμική δόμηση της διεπαφής με βάση τις επιλογές του χρήστη.
- **Σχέσεις Εξάρτησης (Include)**: Στο διάγραμμα παρατηρείται η σχέση <<include>> μεταξύ των περιπτώσεων προβολής δεδομένων και της **Ανάκτησης Δεδομένων (UC6)**. Αυτό υποδηλώνει ότι για να ολοκληρωθεί η προβολή μιας βαθμολογίας ή ενός ρόστερ, το σύστημα υποχρεωτικά εκτελεί τη λειτουργία ανάκτησης πληροφοριών από το API μέσω του Proxy Server.

- **Αποτέλεσμα Αλληλεπίδρασης**: Ο σχεδιασμός αυτός αναδεικνύει τη **Δυναμική Ενημέρωση** του συστήματος. Κάθε ενέργεια του χρήστη (όπως το κλικ σε μια ομάδα) δεν είναι μια απλή αλλαγή σελίδας, αλλά μια πυροδότηση (trigger) μιας αλυσίδας γεγονότων που καταλήγει στην παρουσίαση φρέσκιας πληροφορίας στην οθόνη.

Ο σχολιασμός αυτός τεκμηριώνει ότι η εφαρμογή έχει σχεδιαστεί με επίκεντρο τον χρήστη, διασφαλίζοντας ότι όλες οι διαθέσιμες πληροφορίες είναι προσβάσιμες με ελάχιστα βήματα (clicks), ενώ η τεχνική πολυπλοκότητα της ανάκτησης των δεδομένων παραμένει διαφανής σε αυτόν.

3.2.2Activity Diagram: Ροή αιτημάτων από το frontend στο proxy και πίσω



Αναλυτική Περιγραφή και Σχολιασμός Διαγράμματος

Το Διάγραμμα Δραστηριοτήτων απεικονίζει τη διαδρομή που ακολουθεί η πληροφορία από τη στιγμή της αλληλεπίδρασης του χρήστη έως την τελική προβολή των δεδομένων στην οθόνη.

- **Η Φάση του Αιτήματος**: Η δραστηριότητα ξεκινά από το Frontend, όπου τα Hooks της React (useEffect) αναλαμβάνουν να "πυροδοτήσουν" την επικοινωνία αμέσως μόλις φορτωθεί ένα Component.
- **Ο Ρόλος του Ενδιάμεσου**: Στο κέντρο της δραστηριότητας βρίσκεται ο Proxy Server. Ο διαχωρισμός της προσθήκης του API Key σε αυτό το επίπεδο (όπως φαίνεται στο διάγραμμα) αναδεικνύει τη στρατηγική ασφαλείας που ακολουθήθηκε, κρατώντας τα διαπιστευτήρια μακριά από τον Client.
- **Λήψη και Διακλάδωση**: Το διάγραμμα περιλαμβάνει ένα κρίσιμο σημείο απόφασης (Decision Point) σχετικά με την επιτυχία της λήψης των δεδομένων. Η ύπαρξη της διαδρομής "Catch Error" αποδεικνύει ότι η εφαρμογή έχει σχεδιαστεί για να είναι ανθεκτική (robust), διαχειριζόμενη περιπτώσεις απώλειας σύνδεσης ή αποτυχίας του εξωτερικού API.
- **Ολοκλήρωση**: Η ροή καταλήγει στην ενημέρωση της κατάστασης (State Update), η οποία προκαλεί τον επανασχεδιασμό του UI, μετατρέποντας τα ακατέργαστα δεδομένα JSON σε μορφοποιημένη πληροφορία για τον χρήστη.

3.3Διαχείριση Καταστάσεων (State Management)

Η διαχείριση της κατάστασης αποτελεί την πιο κρίσιμη πτυχή της ανάπτυξης με React, καθώς καθορίζει τη συμπεριφορά και την εμφάνιση των συστατικών (components) σε κάθε χρονική στιγμή. Στο "Sport90", η προσέγγιση που ακολουθήθηκε βασίζεται στα **React Hooks**, τα οποία επιτρέπουν τη χρήση λειτουργιών της React χωρίς τη συγγραφή παραδοσιακών κλάσεων (classes). Η μεθοδολογία αυτή επιλέχθηκε για να διασφαλιστεί ότι η εφαρμογή παραμένει ελαφριά και αποδοτική, αποφεύγοντας τη χρήση σύνθετων εξωτερικών βιβλιοθηκών διαχείρισης κατάστασης (όπως το Redux) που θα πρόσθεταν περιττή πολυπλοκότητα σε ένα project αυτού του βεληνεκούς.

Μέσω της συνδυαστικής χρήσης των Hooks, η εφαρμογή επιτυγχάνει έναν πλήρη κύκλο διαχείρισης πληροφορίας: από την αρχική δήλωση κενών μεταβλητών, στην αυτόματη ανάκτηση δεδομένων μέσω του δικτύου, και τελικά στην ασφαλή αποθήκευση και προβολή τους. Η στρατηγική αυτή δεν περιορίζεται μόνο στην επιτυχή λήψη των δεδομένων, αλλά επεκτείνεται και στην πρόβλεψη δυσλειτουργιών. Η διαχείριση της κατάστασης περιλαμβάνει ειδικούς μηχανισμούς για τον έλεγχο των σφαλμάτων και της φάσης αναμονής

(loading state), διασφαλίζοντας ότι ο χρήστης έχει πάντα μια σαφή εικόνα για το τι συμβαίνει στο παρασκήνιο, ακόμα και όταν οι συνθήκες του δικτύου δεν είναι ευνοϊκές.

3.3.1Χρήση του useState για την Αποθήκευση Δεδομένων

Το Hook **useState** αποτελεί το βασικό εργαλείο για τη δήλωση και τη διαχείριση της τοπικής κατάστασης μέσα στα λειτουργικά συστατικά (functional components) του "Sport90". Η λειτουργία του βασίζεται στην επιστροφή ενός ζεύγους στοιχείων: της τρέχουσας τιμής της κατάστασης και μιας συνάρτησης που επιτρέπει την ενημέρωσή της. Στο project μας, η χρήση του useState είναι καθοριστική για τη διατήρηση της ακεραιότητας των δεδομένων, καθώς κάθε φορά που η συνάρτηση ενημέρωσης καλείται με νέα δεδομένα, η React προβαίνει αυτόματα σε επανασχεδιασμό (re-render) του component, διασφαλίζοντας ότι το UI ταυτίζεται πάντα με την τρέχουσα πληροφορία.

Στην πρακτική εφαρμογή του "Sport90", το useState χρησιμοποιείται για τρεις διακριτούς σκοπούς που αφορούν τη διαχείριση των αθλητικών δεδομένων. Πρώτον, χρησιμοποιείται για την **αποθήκευση των πρωτογενών δεδομένων** που επιστρέφονται από τον Proxy Server, όπως για παράδειγμα ο πίνακας clubs στο component της βαθμολογίας ή το αντικείμενο teamDetails στη σελίδα της ομάδας. Δεύτερον, χρησιμοποιείται για τη διαχείριση της **κατάστασης φόρτωσης (loading state)** μέσω της μεταβλητής loadStatus. Αυτή η boolean τιμή επιτρέπει στην εφαρμογή να γνωρίζει πότε μια διαδικασία ανάκτησης βρίσκεται σε εξέλιξη, ώστε να προβάλλει στον χρήστη δυναμικά στοιχεία όπως "Data is Loading..." αντί για μια κενή ή ημιτελή σελίδα.

Τέλος, το useState παίζει κεντρικό ρόλο στον **έλεγχο της ροής των σφαλμάτων**, διατηρώντας την κατάσταση errorStatus. Η στρατηγική αυτή επιτρέπει στο component να "θυμάται" αν η τελευταία προσπάθεια επικοινωνίας με το API ήταν επιτυχής ή όχι. Εάν προκύψει κάποιο πρόβλημα στο δίκτυο ή στον διακομιστή, η κατάσταση αυτή ενημερώνεται άμεσα, ενεργοποιώντας την προβολή εναλλακτικού περιεχομένου (conditional rendering). Με αυτόν τον τρόπο, το useState μετατρέπει τα στατικά αρχεία JavaScript σε ζωντανά δομικά στοιχεία που αντιλαμβάνονται το περιβάλλον τους και προσαρμόζουν την εμφάνισή τους με βάση τα δεδομένα που διαχειρίζονται.

3.3.2Χρήση του useEffect για τον Συγχρονισμό Δεδομένων

Το Hook **useEffect** λειτουργεί ως ο κεντρικός ενορχηστρωτής των "παρενεργειών" στην εφαρμογή **"Sport90"**, επιτρέποντας στα συστατικά στοιχεία να συγχρονίζονται με εξωτερικά συστήματα, όπως ο Proxy Server και το Football-Data API. Στην αρχιτεκτονική της React, τα components πρέπει να παραμένουν "καθαρά" όσον αφορά την οπτικοποίηση, επομένως οποιαδήποτε ενέργεια απαιτεί επικοινωνία με το δίκτυο ή μεταβολή δεδομένων εκτός του τοπικού ελέγχου του component, πρέπει να περικλείεται μέσα στο useEffect. Αυτό διασφαλίζει ότι η ανάκτηση των δεδομένων δεν θα εμποδίσει το αρχικό rendering της σελίδας, διατηρώντας την εφαρμογή γρήγορη και αποκρινόμενη.

Στο project μας, η χρήση του useEffect υλοποιείται με δύο βασικές στρατηγικές. Η πρώτη αφορά την **αυτόματη ανάκτηση δεδομένων κατά τη φόρτωση** (Component Mount). Μέσω ενός κενού πίνακα εξαρτήσεων ([]), το Hook διασφαλίζει ότι η συνάρτηση λήψης δεδομένων

θα εκτελεστεί ακριβώς μία φορά όταν ο χρήστης εισέρχεται σε μια σελίδα, όπως η αρχική ή η σελίδα των ειδήσεων. Αυτή η προσέγγιση είναι ιδανική για σταθερά δεδομένα που δεν αλλάζουν κατά τη διάρκεια της παραμονής του χρήστη στην ίδια οθόνη, μειώνοντας τις περιττές κλήσεις προς τον διακομιστή και εξοικονομώντας πόρους.

Η δεύτερη και πιο σύνθετη στρατηγική αφορά τον **συγχρονισμό βάσει παραμέτρων** (Dependency Array). Σε δυναμικές σελίδες, όπως το TeamInfo ή το Ranking, το useEffect παρακολουθεί συγκεκριμένες μεταβλητές, όπως το αναγνωριστικό της ομάδας (id) ή το όνομα του πρωταθλήματος. Εάν ο χρήστης πλοηγηθεί από το προφίλ μιας ομάδας σε μια άλλη, το Hook ανιχνεύει την αλλαγή στην παράμετρο και πυροδοτεί εκ νέου τη διαδικασία ανάκτησης δεδομένων για τη νέα οντότητα. Αυτή η δυναμική συμπεριφορά επιτρέπει στο "Sport90" να λειτουργεί ως μια αληθινή Μονοσελιδική Εφαρμογή (SPA), όπου το περιεχόμενο ανανεώνεται "έξυπνα" και στοχευμένα, προσφέροντας μια συνεχή και αδιάλειπτη εμπειρία πλοήγησης στον φίλαθλο.

3.3.3Διαχείριση Σφαλμάτων Δικτύου (Error Handling)

Η διαχείριση σφαλμάτων δικτύου αποτελεί κρίσιμο κομμάτι της μεθοδολογίας του **"Sport90"**, καθώς διασφαλίζει ότι η εφαρμογή παραμένει λειτουργική και φιλική προς τον χρήστη ακόμη και όταν προκύπτουν τεχνικά προβλήματα στην επικοινωνία με το API. Σε περιβάλλοντα που βασίζονται σε εξωτερικές πηγές δεδομένων, οι αστοχίες είναι αναπόφευκτες —είτε λόγω απώλειας της σύνδεσης στο διαδίκτυο, είτε λόγω υπέρβασης των ορίων αιτημάτων (rate limits) του Football-Data API, είτε λόγω προσωρινής δυσλειτουργίας του Proxy Server. Η στρατηγική που ακολουθήθηκε στοχεύει στην αποφυγή της κατάρρευσης της εφαρμογής και στην παροχή σαφούς πληροφόρησης στον χρήστη.

Η τεχνική υλοποίηση βασίζεται στη χρήση των δομών **try...catch** μέσα στις ασύγχρονες συναρτήσεις ανάκτησης δεδομένων. Όταν ένα αίτημα αποστέλλεται μέσω του Axios, η εφαρμογή "δοκιμάζει" (try) να λάβει την πληροφορία. Εάν ο διακομιστής επιστρέψει κωδικό σφάλματος (π.χ. 429 Too Many Requests ή 500 Internal Server Error), ο έλεγχος περνά αμέσως στο block "catch". Εκεί, η εφαρμογή ενημερώνει την κατάσταση **errorStatus** σε true, μια ενέργεια που πυροδοτεί αυτόματα την αλλαγή της διεπαφής. Αυτό το μοτίβο διασφαλίζει ότι ο κώδικας δεν θα προσπαθήσει να επεξεργαστεί ανύπαρκτα δεδομένα, κάτι που θα οδηγούσε σε σφάλματα τύπου "undefined" και ολική κατάρρευση της σελίδας.

Στο επίπεδο της διεπαφής (UI), η διαχείριση αυτή μεταφράζεται σε **υπό συνθήκη προβολή (conditional rendering)**. Αντί για μια κενή οθόνη ή ένα σφάλμα συστήματος, ο χρήστης βλέπει ένα ειδικά σχεδιασμένο μήνυμα που τον ενημερώνει για το πρόβλημα, προτρέποντάς τον να ελέγξει τη σύνδεσή του ή να προσπαθήσει ξανά αργότερα. Επιπλέον, ο Proxy Server έχει ρυθμιστεί ώστε να καταγράφει (log) τα σφάλματα στην πλευρά του διακομιστή, διευκολύνοντας τον εντοπισμό και τη διόρθωση προβλημάτων κατά τη διάρκεια της ανάπτυξης. Αυτή η πολυεπίπεδη προσέγγιση —από τον διακομιστή έως την οθόνη του χρήστη— προσδίδει στο "Sport90" έναν επαγγελματικό και αξιόπιστο χαρακτήρα, διατηρώντας την εμπιστοσύνη του χρήστη ακόμη και σε συνθήκες τεχνικής αστάθειας.

4.Υλοποίηση & Εγχειρίδιο Χρήσης

Η ολοκλήρωση της ανάπτυξης μιας web εφαρμογής σηματοδοτείται από τη μετάβαση από τον κώδικα σε ένα πλήρως λειτουργικό και φιλικό προς τον χρήστη περιβάλλον. Στο κεφάλαιο αυτό παρουσιάζεται η τελική υλοποίηση του "Sport90", αναλύοντας τη δομή των σελίδων, τον τρόπο πλοήγησης και τις εξειδικευμένες λειτουργίες που προσφέρονται στον φίλαθλο. Η παρουσίαση δεν περιορίζεται μόνο στην τεχνική περιγραφή, αλλά λειτουργεί και ως ένας οδηγός που αναδεικνύει την προστιθέμενη αξία της εφαρμογής στην παροχή αθλητικής πληροφόρησης.

Η υλοποίηση βασίστηκε στην αρχή της «πληροφορίας με ένα κλικ», διασφαλίζοντας ότι ο χρήστης μπορεί να μεταβεί από τη γενική εικόνα ενός πρωταθλήματος στις λεπτομέρειες μιας ομάδας με απόλυτη φυσικότητα. Μέσα από την ανάλυση του routing και των επιμέρους ενοτήτων —από τους δυναμικούς πίνακες βαθμολογίας έως τα αναλυτικά προφίλ των συλλόγων— τεκμηριώνεται η ορθή εφαρμογή των τεχνολογιών που επιλέχθηκαν στα προηγούμενα κεφάλαια.

Ιδιαίτερη έμφαση δίνεται στην οπτική επικοινωνία και την εμπειρία χρήστη (UX). Η παρουσίαση των λειτουργιών συνοδεύεται από οπτικό υλικό (screenshots) που αποδεικνύει την αρτιότητα του interface, ενώ η ενότητα του Responsive Design επιβεβαιώνει ότι το "Sport90" ανταποκρίνεται στις σύγχρονες απαιτήσεις για καθολική πρόσβαση από κάθε είδους συσκευή, είτε πρόκειται για σταθερό υπολογιστή είτε για κινητό τηλέφωνο.

4.1Δομή Εφαρμογής: Ανάλυση του Routing και των Pages

Η αρχιτεκτονική του "Sport90" σχεδιάστηκε με γνώμονα τη modularity (αρθρωτικότητα) και την ευκολία στη συντήρηση. Ο διαχωρισμός των λειτουργιών σε διακριτά αρχεία και η χρήση ενός κεντρικού συστήματος δρομολόγησης διασφαλίζουν ότι η εφαρμογή παραμένει οργανωμένη καθώς επεκτείνεται.

4.1.1Οργάνωση Φακέλων και Αρχείων

Η αρχιτεκτονική των αρχείων στο project "Sport90" έχει δομηθεί με βάση την αρχή του διαχωρισμού των αρμοδιοτήτων (**Separation of Concerns**), επιτρέποντας τη διακριτή διαχείριση της επιχειρησιακής λογικής (logic) και της οπτικής παρουσίασης (presentation). Αυτή η προσέγγιση είναι θεμελιώδης για την ανάπτυξη με React, καθώς επιτρέπει στον προγραμματιστή να εντοπίζει και να διορθώνει σφάλματα σε συγκεκριμένα τμήματα της εφαρμογής χωρίς να επηρεάζει την υπόλοιπη δομή. Η οργάνωση αυτή αντικατοπτρίζεται στον κατάλογο src, ο οποίος λειτουργεί ως ο κεντρικός πυρήνας της εφαρμογής.

Στον φάκελο **Components** φιλοξενούνται όλα τα επαναχρησιμοποιήσιμα δομικά στοιχεία της διεπαφής. Στοιχεία όπως το Navbar.jsx, το Footer.jsx και το Sidebar.jsx λειτουργούν ως σταθερά πλαίσια (layouts) που πλαισιώνουν το περιεχόμενο, διατηρώντας την αισθητική ομοιομορφία σε όλο το εύρος της εφαρμογής. Η χρήση αυτόνομων components επιτρέπει την «ατομική» σχεδίαση (atomic design), όπου μικρά κομμάτια κώδικα —όπως το

dataTables.jsx για τους πίνακες— μπορούν να ενσωματωθούν σε διαφορετικές σελίδες, μειώνοντας την επανάληψη κώδικα και διευκολύνοντας τις καθολικές αλλαγές στο styling.

Αντίστοιχα, ο φάκελος **Pages** περιλαμβάνει τα κύρια "Views" ή τις σελίδες που βλέπει ο χρήστης ως ολοκληρωμένες ενότητες. Αρχεία όπως το Home.jsx, το Ranking.jsx και το TeamInfo.jsx λειτουργούν ως ενορχηστρωτές: καλούν τα απαραίτητα components, εκτελούν τις εξειδικευμένες κλήσεις προς τον Proxy Server μέσω των Hooks και διαχειρίζονται τη ροή των δεδομένων για τη συγκεκριμένη οθόνη. Με αυτόν τον τρόπο, μια "σελίδα" στη React δεν είναι ένα στατικό αρχείο HTML, αλλά ένας δυναμικός συνδυασμός components που αντιδρούν σε πραγματικό χρόνο στις παραμέτρους του URL.

Τέλος, η διαχείριση των στατικών πόρων και της εμφάνισης συγκεντρώνεται στον φάκελο **Assets & CSS**. Η επιλογή να οργανωθούν τα αρχεία στυλ ανά ενότητα (π.χ. newsResponsive.css) και να συνδυαστούν με το framework **Tailwind CSS** προσφέρει μεγάλη ευελιξία. Αυτή η οργάνωση είναι ιδιαίτερα κρίσιμη για τον **responsive σχεδιασμό**, καθώς επιτρέπει την απομόνωση των κανόνων που αφορούν τη συμπεριφορά της εφαρμογής σε κινητά και tablets. Έτσι, οι εικόνες, τα εικονίδια και οι κανόνες μορφοποίησης παραμένουν ταξινομημένοι, διασφαλίζοντας ότι η αισθητική ταυτότητα του "Sport90" παραμένει αναλλοίωτη ανεξάρτητα από την πολυπλοκότητα του περιεχομένου.

4.1.2Δυναμικό Routing με React Router

Η πλοήγηση στην εφαρμογή "Sport90" βασίζεται στη βιβλιοθήκη **React Router DOM**, η οποία αποτελεί το πρότυπο για τη διαχείριση διαδρομών σε σύγχρονες εφαρμογές React. Μέσω αυτού του εργαλείου, επιτυγχάνεται η δημιουργία μιας εμπειρίας **Single Page Application (SPA)**, όπου η μετάβαση από τη μία ενότητα στην άλλη γίνεται ακαριαία στο επίπεδο του πελάτη (client-side), χωρίς να απαιτείται η παραδοσιακή επαναφόρτωση ολόκληρης της σελίδας από τον διακομιστή. Στο κεντρικό αρχείο App.jsx, ορίζεται ο "χάρτης" της εφαρμογής, ο οποίος καθορίζει ποιο συστατικό (component) θα προβληθεί με βάση τη διεύθυνση URL που πληκτρολογεί ή επιλέγει ο χρήστης.

Οι **στατικές διαδρομές** αποτελούν τους σταθερούς πυλώνες του "Sport90". Διαδρομές όπως το / για την αρχική σελίδα (Home) ή το /news για την ενότητα των αθλητικών νέων, οδηγούν σε συγκεκριμένα, προκαθορισμένα περιβάλλοντα. Αυτές οι διαδρομές εξυπηρετούν την ανάγκη για γρήγορη πρόσβαση στις βασικές λειτουργίες της εφαρμογής, παρέχοντας ένα οικείο σημείο αναφοράς για τον χρήστη. Παρά τη στατικότητά τους ως προς το path, το περιεχόμενο που προβάλλουν παραμένει δυναμικό, καθώς τα αντίστοιχα components ανανεώνουν τις πληροφορίες τους μέσω του API κάθε φορά που φορτώνονται.

Η πραγματική ισχύς του συστήματος, ωστόσο, αναδεικνύεται μέσω των **δυναμικών διαδρομών (Dynamic Routes)**. Το "Sport90" αξιοποιεί τις παραμέτρους (URL parameters) στα paths, όπως το /ranking/:id ή το /team/:id, για να δημιουργήσει ευέλικτα components. Με αυτή την προσέγγιση, ένα μόνο αρχείο, όπως το Ranking.jsx, μπορεί να εξυπηρετήσει δεκάδες διαφορετικά πρωταθλήματα. Όταν ο χρήστης επιλέγει την Premier League, το ID PL περνάει μέσω του URL στο component, το οποίο με τη σειρά του "καταλαβαίνει" ποια δεδομένα πρέπει να ζητήσει από τον Proxy Server. Αυτή η αρχιτεκτονική μειώνει δραστικά τον όγκο του κώδικα και απλοποιεί τη συντήρηση της εφαρμογής.

Η **σύνδεση με το περιβάλλον χρήστη (UI)** πραγματοποιείται μέσω του component Link της React Router, το οποίο αντικαθιστά τους παραδοσιακούς συνδέσμους <a>. Η χρήση του Link είναι καθοριστική για την απόδοση: αντί ο browser να ζητήσει ένα νέο αρχείο HTML από τον server, η React Router απλώς ενημερώνει τη διεύθυνση στον φυλλομετρητή και αλλάζει το περιεχόμενο της οθόνης επεξεργαζόμενη μόνο τα απαραίτητα components. Αυτή η διαδικασία, γνωστή ως "client-side routing", διασφαλίζει ότι η μετάβαση από τον πίνακα βαθμολογίας στις λεπτομέρειες μιας ομάδας (π.χ. στο TeamInfo.jsx) γίνεται ομαλά, διατηρώντας την αίσθηση μιας ενιαίας, στιβαρής εφαρμογής desktop.

Παρακάτω βλέπουμε αναλυτικά στον Πίνακα τις διαδρομές:

Διαδρομή (Path)	Component	Τύπος	Περιγραφή Λειτουργίας
/	Home	Στατική	Η κεντρική σελίδα της εφαρμογής με επισκόπηση των κορυφαίων πρωταθλημάτων.
/news	NewsPage	Στατική	Εμφάνιση αθλητικής ειδησεογραφίας και πρόσφατων νέων.
/ranking/:id	Ranking	**Δυναμική**	Προβολή βαθμολογίας για το πρωτάθλημα με το συγκεκριμένο :id (π.χ. PL, PD).
/team/:id	TeamInfo	**Δυναμική**	Αναλυτικές πληροφορίες ομάδας, ρόστερ και τρέχουσες διοργανώσεις βάσει του :id.
/full-table/:id	FullTable	**Δυναμική**	Εκτενής πίνακας στατιστικών (νίκες, ήττες, γκολ) για το επιλεγμένο πρωτάθλημα.
/matches/:id	Matches	**Δυναμική**	Εμφάνιση πρόσφατων αποτελεσμάτων και μελλοντικού προγράμματος αγώνων.

4.2Παρουσίαση Λειτουργιών

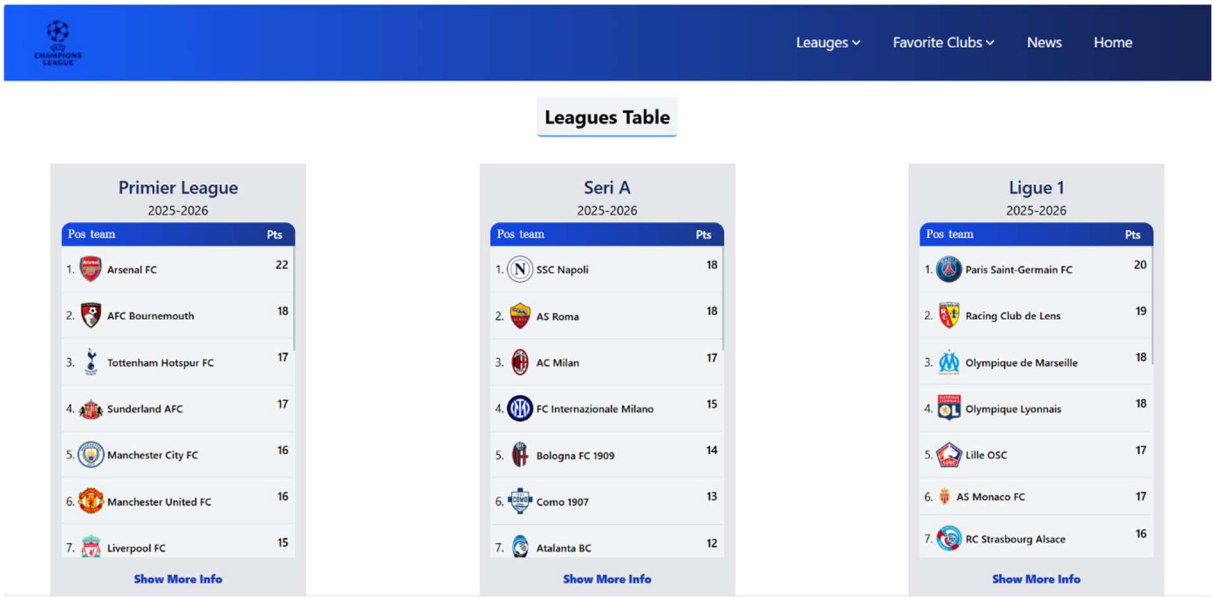
Η διεπαφή του "Sport90" έχει σχεδιαστεί με γνώμονα την καθαρότητα και την αμεσότητα. Οι λειτουργίες χωρίζονται σε τρεις κύριους άξονες που καλύπτουν όλο το φάσμα της αθλητικής πληροφόρησης.

4.2.1Leagues Table & Full Table: Βαθμολογίες και Στατιστικά

Η οπτικοποίηση των δεδομένων βαθμολογίας αποτελεί τον δομικό πυρήνα του "Sport90" και έχει σχεδιαστεί ώστε να εξυπηρετεί τόσο τον χρήστη που επιθυμεί μια γρήγορη ενημέρωση όσο και τον απαιτητικό αναλυτή των στατιστικών. Η εφαρμογή υιοθετεί μια ιεραρχική προσέγγιση στην προβολή της πληροφορίας, η οποία υλοποιείται μέσω δύο διακριτών επιπέδων, διασφαλίζοντας ότι το περιβάλλον εργασίας παραμένει καθαρό και λειτουργικό χωρίς να κατακλύζει τον χρήστη με περιττές λεπτομέρειες στην αρχική επαφή.

Στο πρώτο επίπεδο, αυτό των **Συνοπτικών Πινάκων (Home)**, η εφαρμογή λειτουργεί ως ένα κεντρικό ταμπλό ελέγχου (dashboard). Όπως παρατηρείται και στην Εικόνα 4.2.1.1, η αρχική σελίδα οργανώνεται σε αυτόνομες κάρτες για κάθε μεγάλο ευρωπαϊκό πρωτάθλημα, όπως η Premier League, η Serie A και η Ligue 1. Σε αυτό το στάδιο, προβάλλονται οι πρώτες επτά ομάδες της κατάταξης με τις βασικές πληροφορίες: τη θέση τους, το λογότυπο, το όνομα της ομάδας και τους συνολικούς πόντους (Pts). Η χρήση του κουμπιού **"Show More Info"** στο

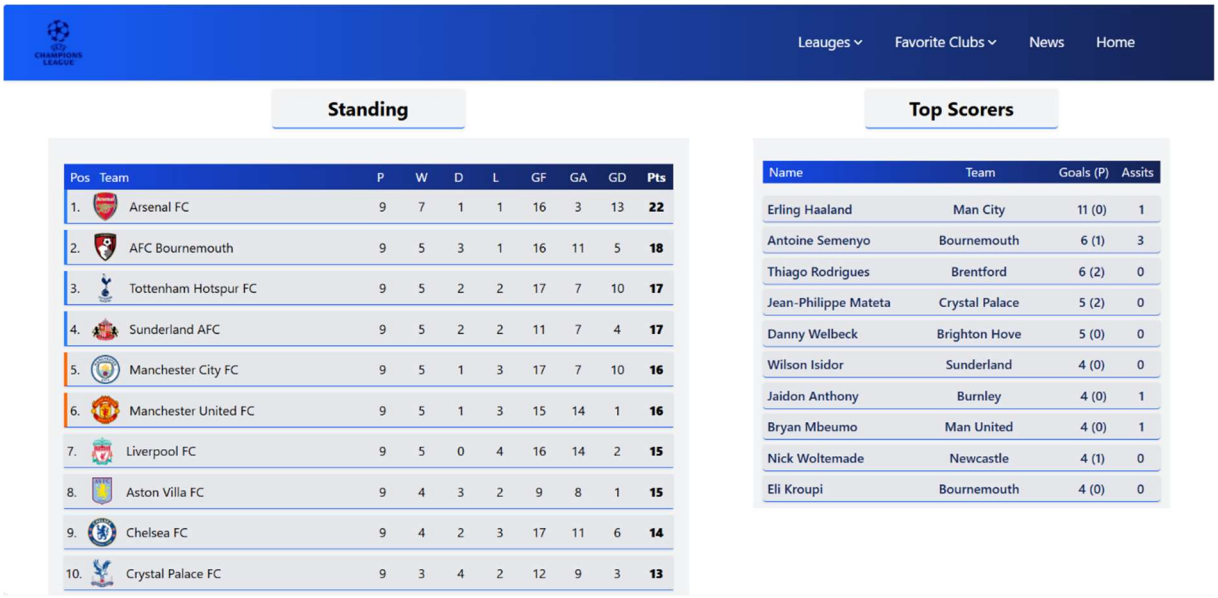
κάτω μέρος κάθε κάρτας λειτουργεί ως πύλη εισόδου, επιτρέποντας στον χρήστη να μεταβεί μέσω δυναμικού routing στο επόμενο επίπεδο ανάλυσης.



Εικόνα 4.2.1.1 Αρχική Σελίδα

Το δεύτερο επίπεδο αφορά την **Αναλυτική Βαθμολογία (Standing)**, η οποία προσφέρει μια πλήρη ακτινογραφία της αγωνιστικής κατάστασης του πρωταθλήματος. Στη σελίδα αυτή, ο πίνακας επεκτείνεται οριζόντια για να συμπεριλάβει όλες τις κρίσιμες μεταβλητές που καθορίζουν την πορεία των ομάδων. Ο χρήστης μπορεί να αναλύσει την απόδοση κάθε συλλόγου μέσω των στηλών για τους συνολικούς αγώνες (P), τις νίκες (W), τις ισοπαλίες (D) και τις ήττες (L). Επιπλέον, παρέχονται εξειδικευμένα δεδομένα για την παραγωγικότητα και την αμυντική ευστάθεια, όπως τα γκολ υπέρ (GF), τα γκολ κατά (GA) και η διαφορά τερμάτων (GD), η οποία συχνά αποτελεί το κλειδί για την ισοβαθμία.

Παράλληλα με την ομαδική κατάταξη, η εφαρμογή ενσωματώνει έναν αυτόνομο **Πίνακα Σκόρερ (Top Scorers)**, ενισχύοντας τη σφαιρική εικόνα της διοργάνωσης. Όπως φαίνεται και στην Εικόνα 4.2.1.2, η ενότητα αυτή προβάλλει τους κορυφαίους ποδοσφαιριστές, συνδέοντας τα ονόματά τους με τις αντίστοιχες ομάδες τους. Η ανάλυση εδώ γίνεται σε ατομικό επίπεδο, παρουσιάζοντας τον αριθμό των γκολ, τον αριθμό των πέναλτι που έχουν εκτελεστεί (σημειώνονται σε παρένθεση) καθώς και τις ασίστ (Assits) που έχουν μοιράσει. Αυτή η συνδυαστική προβολή βαθμολογίας και ατομικών επιδόσεων επιτρέπει στον φίλαθλο να έχει μια πλήρη και εμπεριστατωμένη εικόνα για την εξέλιξη του πρωταθλήματος μέσα από μία μόνο οθόνη.



Εικόνα 4.2.1.2 Σελίδα Scorers

4.2.2Team Pages: Πλήρες Προφίλ Ομάδας

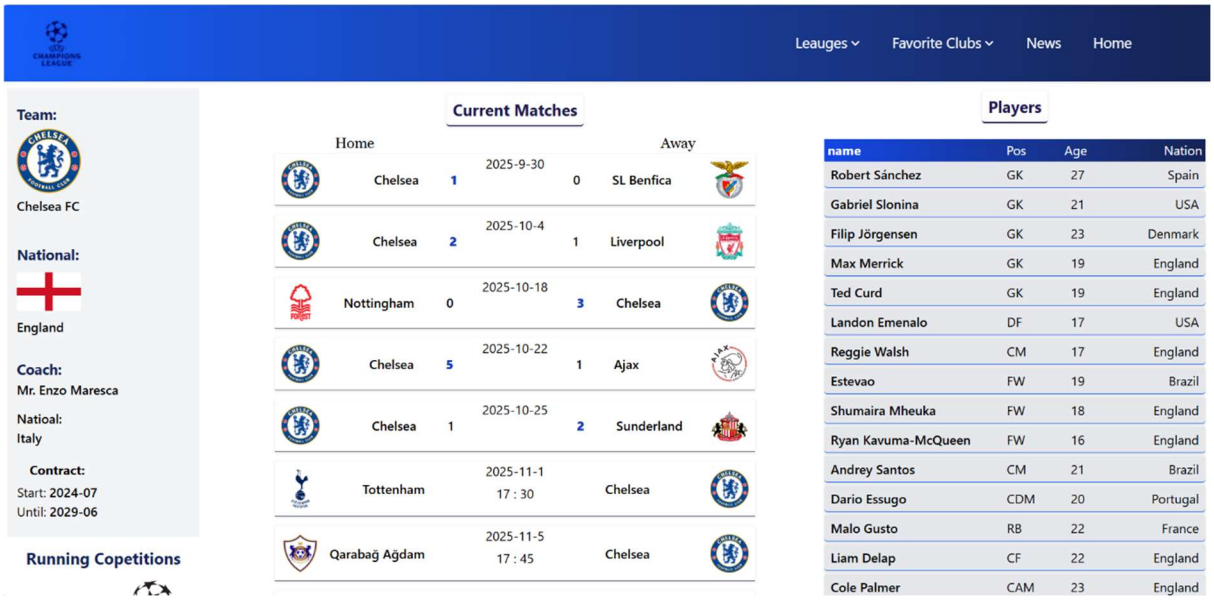
Η σελίδα της ομάδας στο "Sport90" αποτελεί έναν από τους πιο σύνθετους λειτουργικούς κόμβους της εφαρμογής, καθώς συνδυάζει δεδομένα από διαφορετικά επίπεδα πληροφόρησης (διοικητικά, αγωνιστικά και στελέχωσης) σε μία ενιαία και εύχρηστη διεπαφή. Όπως τεκμηριώνεται στην Εικόνα 4.2.2.1, ο σχεδιασμός ακολουθεί μια δομή τριών στηλών, η οποία επιτρέπει στον χρήστη να προσλαμβάνει ταυτόχρονα την ταυτότητα της ομάδας, το τρέχον πρόγραμμα και τη σύνθεση του ρόστερ.

Στην **Πλευρική Μπάρα Πληροφοριών (Sidebar)**, η εφαρμογή προβάλλει τα θεμελιώδη στοιχεία του συλλόγου, λειτουργώντας ως μια γρήγορη "ταυτότητα". Εκτός από το λογότυπο της ομάδας (π.χ. Chelsea FC) και το εθνόσημο της χώρας προέλευσης, δίνεται ιδιαίτερη έμφαση στην τεχνική ηγεσία. Ο χρήστης μπορεί να δει άμεσα το όνομα του προπονητή, την εθνικότητά του, καθώς και τις ακριβείς ημερομηνίες έναρξης και λήξης του συμβολαίου του, προσφέροντας μια επαγγελματική διάσταση στην παρουσίαση των δεδομένων. Επιπλέον, στο κάτω μέρος της μπάρας εμφανίζονται τα λογότυπα των διοργανώσεων (Running Competitions) στις οποίες συμμετέχει ο σύλλογος, υποδηλώνοντας το αγωνιστικό του βεληνεκές.

Στο κεντρικό τμήμα της σελίδας δεσπόζει η ενότητα **Τρέχοντες Αγώνες (Current Matches)**, η οποία προσφέρει μια χρονολογική επισκόπηση της αγωνιστικής φόρμας. Η παρουσίαση είναι εξαιρετικά καθαρή: κάθε αγώνας περιλαμβάνει τα λογότυπα και τα ονόματα των αντιπάλων, το τελικό σκορ με έντονη γραφή για τα γκολ, καθώς και την ημερομηνία διεξαγωγής. Η διάταξη αυτή επιτρέπει στον φίλαθλο να ξεχωρίζει με μια ματιά τα αποτελέσματα των εντός και εκτός έδρας αναμετρήσεων, ενώ παράλληλα ενημερώνεται για επερχόμενους αγώνες στους οποίους αναγράφεται η ώρα έναρξης αντί για σκορ.

Η δεξιά πλευρά της σελίδας καταλαμβάνεται από τη **Λίστα Παικτών (Players)**, η οποία υλοποιείται μέσω ενός δυναμικού και οργανωμένου πίνακα. Το ρόστερ παρουσιάζεται με

πλήρη στατιστικά στοιχεία για κάθε αθλητή, περιλαμβάνοντας το ονοματεπώνυμο, τη θέση στο γήπεδο (Pos), την ηλικία (Age) και την εθνική του ταυτότητα (Nation). Η ταξινόμηση αυτή βοηθά στην κατανόηση της δομής της ομάδας, ενώ η χρήση συντομογραφιών (π.χ. GK, DF, CM, FW) ακολουθεί τα διεθνή αθλητικά πρότυπα, καθιστώντας την πληροφορία άμεσα αναγνώσιμη.



Εικόνα 4.2.2.1 Προφίλ Ομάδας

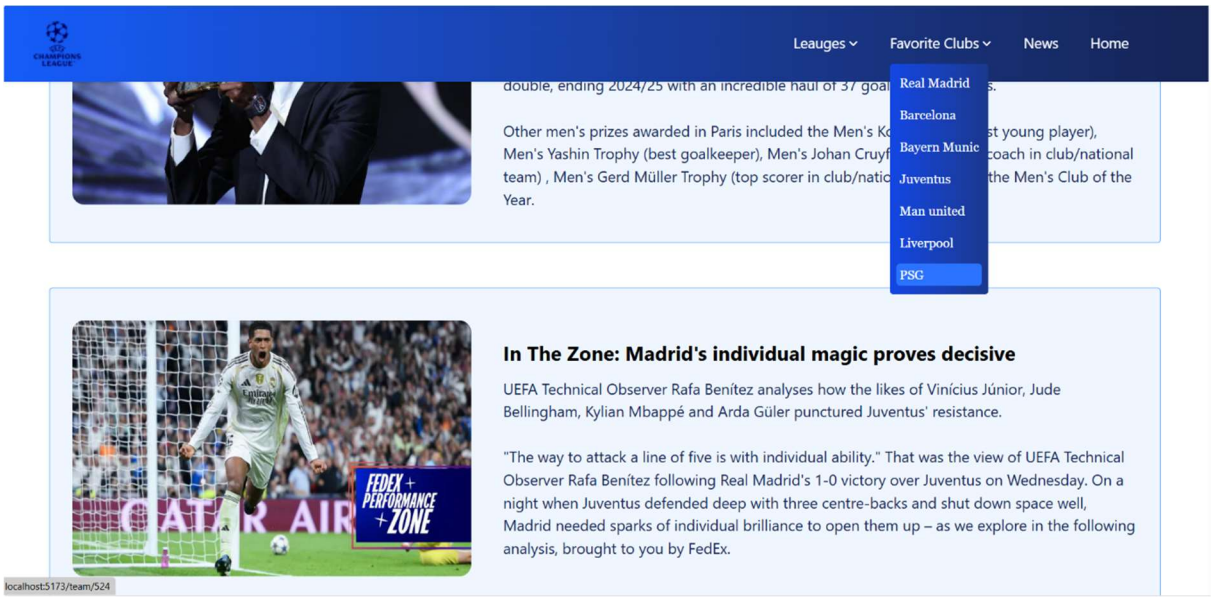
4.2.3 Ειδησεογραφία και Πλοήγηση

Η εμπειρία του χρήστη στο "Sport90" ολοκληρώνεται μέσα από την **Ενότητα Νέων (News)**, η οποία μετατρέπει την εφαρμογή από ένα απλό εργαλείο στατιστικών σε μια ολοκληρωμένη πλατφόρμα αθλητικής ενημέρωσης. Όπως τεκμηριώνεται στην Εικόνα 4.2.3.1, η σελίδα των ειδήσεων έχει σχεδιαστεί με έμφαση στην υψηλή αισθητική και την αναγνωσιμότητα, ενσωματώνοντας άρθρα που συνοδεύονται από πλούσιο οπτικό υλικό. Το περιεχόμενο περιλαμβάνει εις βάθος αναλύσεις, όπως οι τεχνικές παρατηρήσεις της UEFA (UEFA Technical Observer analyses), προσφέροντας προσθετή αξία στον φίλαθλο που αναζητά κάτι περισσότερο από ένα σκορ. Η διάταξη των ειδήσεων σε κάρτες με καθαρούς τίτλους και περιλήψεις επιτρέπει τη γρήγορη σάρωση της πληροφορίας, ενώ η χρήση του CSS διασφαλίζει ότι η παρουσίαση παραμένει ελκυστική σε κάθε ανάλυση οθόνης.

Η πλοήγηση στην εφαρμογή διευκολύνεται από ένα **Διαδραστικό Μενού (Navbar)** και ένα πλήρως οργανωμένο **Footer**, τα οποία λειτουργούν ως οι σταθεροί οδηγοί του χρήστη. Το Navbar διαθέτει έξυπνα drop-down μενού, όπως η ενότητα "Favorite Clubs", που επιτρέπει την άμεση μετάβαση σε κορυφαίους συλλόγους όπως η Real Madrid, η Barcelona και η Bayern Munich χωρίς περιττά κλικ. Αυτή η δυνατότητα προσφέρει ένα επίπεδο εξατομίκευσης, δίνοντας προτεραιότητα στις ομάδες με το μεγαλύτερο ενδιαφέρον.

Στο κάτω μέρος της εφαρμογής, το **Footer** λειτουργεί ως ένας πλήρης χάρτης του ιστοτόπου (sitemap), οργανωμένος σε δύο κεντρικές στήλες: "Leagues" και "Favorite Teams". Ο χρήστης μπορεί να βρει γρήγορους συνδέσμους για όλα τα υποστηριζόμενα πρωταθλήματα, από την Premier League έως τη Ligue 1, καθώς και για τις δημοφιλέστερες ομάδες. Οπτικά, το Footer

πλαισιώνεται από το εμβληματικό λογότυπο του UEFA Champions League, το οποίο ενισχύει την αθλητική ταυτότητα του project και προσδίδει έναν επαγγελματικό χαρακτήρα στο τελικό αποτέλεσμα. Η χρήση του Footer διασφαλίζει ότι ακόμα και στις πιο βαθιές σελίδες της εφαρμογής, ο χρήστης διατηρεί πάντα την επαφή με τις κύριες κατηγορίες δεδομένων.



Εικόνα 4.2.3.1 Νέα

4.3Responsive Design: Καθολική Πρόσβαση

Στη σύγχρονη εποχή του διαδικτύου, η προσαρμοστικότητα μιας εφαρμογής σε διαφορετικά μεγέθη οθόνης δεν αποτελεί απλώς επιλογή, αλλά βασική προϋπόθεση ποιότητας. Η εφαρμογή **"Sport90"** έχει σχεδιαστεί με τη φιλοσοφία **Mobile-First**, δίνοντας προτεραιότητα στη βέλτιστη εμφάνιση στις φορητές συσκευές και επεκτείνοντας τη λειτουργικότητα καθώς αυξάνεται ο διαθέσιμος χώρος. Η προσέγγιση αυτή διασφαλίζει ότι ο χρήστης έχει καθολική πρόσβαση στα αθλητικά δεδομένα, είτε χρησιμοποιεί smartphone κατά τη διάρκεια ενός αγώνα, είτε tablet, είτε σταθερό υπολογιστή για βαθύτερη στατιστική ανάλυση.

Η υλοποίηση του **Mobile View** αποτελεί πρόκληση για εφαρμογές που διαχειρίζονται μεγάλους πίνακες δεδομένων, ωστόσο στο "Sport90" αυτό επιλύθηκε μέσω έξυπνης σύμπτυξης πληροφοριών. Όπως παρατηρούμε στην Εικόνα 4.3.1, ο πίνακας βαθμολογίας προσαρμόζεται δυναμικά στις μικρές οθόνες. Οι δευτερεύουσες στήλες αποκρύπτονται αυτόματα, ενώ παραμένουν ορατά τα πλέον κρίσιμα δεδομένα: η θέση (Pos), το λογότυπο, το όνομα της ομάδας, οι αγώνες (P), η διαφορά τερμάτων (GD) και οι συνολικοί πόντοι (Pts). Παράλληλα, το σύστημα πλοήγησης (Navbar) μετατρέπεται σε ένα λειτουργικό drop-down σύστημα, εξοικονομώντας πολύτιμο χώρο και διευκολύνοντας τη χρήση με το ένα χέρι.

Στις μεγαλύτερες οθόνες, όπως αυτές των **Desktop & Tablet**, η εφαρμογή αλλάζει ριζικά τη διάταξή της για να αξιοποιήσει το επιπλέον εύρος οριζόντιου χώρου. Αντί για τη γραμμική στοίχιση των κινητών, το interface υιοθετεί μια πολυεπίπεδη διάταξη (layout) που προσφέρει μια αίσθηση **"dashboard"** στον χρήστη. Όπως είδαμε και στις προηγούμενες εικόνες, η εφαρμογή προβάλλει ταυτόχρονα πολλαπλές ενότητες πληροφορίας. Για παράδειγμα, στη σελίδα της ομάδας, η πλευρική μπάρα πληροφοριών, το πρόγραμμα των αγώνων και το

ρόστερ εμφανίζονται σε παράλληλες στήλες, επιτρέποντας τη γρήγορη σύγκριση δεδομένων χωρίς την ανάγκη για συνεχές scrolling.

Η τεχνική υλοποίηση αυτής της προσαρμοστικότητας βασίστηκε στον συνδυασμό των **Media Queries** του CSS και των δυνατοτήτων του framework **Tailwind CSS**. Μέσα από εξειδικευμένα αρχεία στυλ, όπως το newsResponsive.css και το footer-responsive.css, ορίστηκαν συγκεκριμένα "breakpoints" που καθορίζουν πότε το περιεχόμενο θα αναδιατάσσεται. Αυτή η προσεκτική διαχείριση του responsive σχεδιασμού εγγυάται ότι η αισθητική ταυτότητα και η λειτουργική αρτιότητα του "Sport90" παραμένουν αναλλοίωτες, προσφέροντας μια επαγγελματική εμπειρία χρήστη σε κάθε ψηφιακό περιβάλλον.

Εικόνα 4.3.1

4.4Οδηγός Εγκατάστασης και Εκτέλεσης

Η επιτυχής εγκατάσταση και λειτουργία της εφαρμογής **"Sport90"** προϋποθέτει την ύπαρξη του περιβάλλοντος εκτέλεσης **Node.js** στο σύστημα του χρήστη. Η διαδικασία ακολουθεί μια δομημένη πορεία τριών σταδίων, ξεκινώντας από την προετοιμασία των αρχείων, συνεχίζοντας με την παραμετροποίηση των εξαρτήσεων και καταλήγοντας στην ταυτόχρονη εκτέλεση της αρχιτεκτονικής δύο επιπέδων (Client και Proxy). Η μεθοδολογία αυτή διασφαλίζει ότι όλες οι απαραίτητες βιβλιοθήκες, όπως η **React** για το frontend και η **Express** για τον proxy, θα συνεργαστούν αρμονικά στο τοπικό περιβάλλον ανάπτυξης.

Στο στάδιο της **προετοιμασίας και εγκατάστασης εξαρτήσεων**, ο χρήστης αποκτά πρόσβαση στον πηγαίο κώδικα και πλοηγείται στον κεντρικό κατάλογο του project μέσω του τερματικού. Η εγκατάσταση του συνόλου των βιβλιοθηκών, που περιλαμβάνει κρίσιμα εργαλεία όπως το **Axios** για τις κλήσεις API και το **Tailwind CSS** για τη μορφοποίηση, πραγματοποιείται αυτόματα με την εντολή npm install. Η εντολή αυτή αναλύει το αρχείο package.json και ανακτά τις ακριβείς εκδόσεις των πακέτων που απαιτούνται. Πριν από την εκκίνηση, είναι θεμελιώδες να πραγματοποιηθεί η **ρύθμιση του API Key**. Ο χρήστης πρέπει να εισάγει το προσωπικό του κλειδί από το Football-Data.org στο καθορισμένο πεδίο εντός του αρχείου proxy.js, καθώς χωρίς αυτό η επικοινωνία με την εξωτερική πηγή δεδομένων θα απορριφθεί για λόγους ασφαλείας.

Η **εκτέλεση της εφαρμογής** απαιτεί τον συντονισμό δύο διαφορετικών διεργασιών λόγω της διαχωρισμένης φύσης του frontend και του διακομιστή μεσολάβησης. Αρχικά, ενεργοποιείται ο **Proxy Server** με την εντολή node proxy.js (ή μέσω εναλλακτικού script), ο οποίος καταλαμβάνει τη θύρα **5000** και αναλαμβάνει την ασφαλή διαμεσολάβηση των αιτημάτων. Στη συνέχεια, εκκινείται το **Frontend** μέσω του εργαλείου **Vite** με την εντολή npm run dev. Η διεπαφή χρήστη γίνεται προσβάσιμη στη διεύθυνση http://localhost:5173, επιτρέποντας στον browser να επικοινωνεί με τον proxy server και να λαμβάνει τα απαραίτητα δεδομένα.

Τέλος, η **επιβεβαίωση λειτουργίας** πραγματοποιείται με το άνοιγμα του περιηγητή στη διεύθυνση του frontend, όπου η εφαρμογή εκτελεί το αρχικό αίτημα προς τον Proxy. Σε μια επιτυχημένη εγκατάσταση, ο διακομιστής μεσολάβησης τροφοδοτεί άμεσα τη React με τα δεδομένα των πρωταθλημάτων, γεμίζοντας τους πίνακες και τα προφίλ των ομάδων. Σε περίπτωση που προκύψει κάποια αστοχία, το τερματικό λειτουργεί ως εργαλείο **αποσφαλμάτωσης (debugging)**, προβάλλοντας συγκεκριμένα μηνύματα σφάλματος που υποδεικνύουν εάν το πρόβλημα εντοπίζεται στη σύνδεση με το API, στην έλλειψη του API Key ή σε κάποια δέσμευση θύρας.

5.Συμπεράσματα & Μελλοντικές Επεκτάσεις

Στο τελευταίο κεφάλαιο της παρούσας εργασίας, πραγματοποιείται μια συνολική ανασκόπηση και αποτίμηση της πορείας ανάπτυξης της εφαρμογής **"Sport90"**. Η ολοκλήρωση του project δεν αποτελεί μόνο την παράδοση ενός λειτουργικού λογισμικού, αλλά και το επιστέγασμα μιας σύνθετης μαθησιακής διαδικασίας που περιλαμβάνει τη σύζευξη σύγχρονων τεχνολογιών frontend και backend. Μέσα από την αξιολόγηση των εργαλείων που χρησιμοποιήθηκαν, όπως η **React** και το **Node.js**, αναλύονται τα οφέλη της αρχιτεκτονικής τριών επιπέδων και η αποτελεσματικότητα των μηχανισμών διαχείρισης σφαλμάτων που θωρακίζουν την εφαρμογή. Παράλληλα, εξετάζονται οι προκλήσεις που αντιμετωπίστηκαν κατά την επικοινωνία με εξωτερικές πηγές δεδομένων και προτείνονται συγκεκριμένες κατευθύνσεις για τη μελλοντική εξέλιξη του συστήματος, θέτοντας τις βάσεις για τη μετατροπή του "Sport90" σε μια ακόμα πιο ολοκληρωμένη και κλιμακώσιμη αθλητική πλατφόρμα.

5.1Αξιολόγηση Έργου: Τι κέρδισα από την υλοποίηση σε React και Node.js.

Η αξιολόγηση της ανάπτυξης του **"Sport90"** αναδεικνύει τα σημαντικά πλεονεκτήματα που προσφέρει ο συνδυασμός της **React** και του **Node.js** στη δημιουργία σύγχρονων, δυναμικών εφαρμογών. Η υλοποίηση αυτή επέτρεψε τη βαθιά κατανόηση της full-stack αρχιτεκτονικής, αποδεικνύοντας ότι η επιλογή των κατάλληλων τεχνολογικών εργαλείων είναι καθοριστική για την ταχύτητα ανάπτυξης, την απόδοση του συστήματος και την τελική εμπειρία του χρήστη.

Η χρήση της **React** στο frontend προσέφερε μια νέα οπτική στον τρόπο δόμησης της διεπαφής χρήστη μέσω των **components**. Η δυνατότητα διαχωρισμού της εφαρμογής σε αυτόνομα, επαναχρησιμοποιήσιμα τμήματα διευκόλυνε τη διαχείριση του κώδικα και επέτρεψε την ταχεία υλοποίηση σύνθετων λειτουργιών, όπως οι δυναμικοί πίνακες βαθμολογίας και τα προφίλ ομάδων. Επιπλέον, η αξιοποίηση των **React Hooks** (useState, useEffect) κατέδειξε την ισχύ του δηλωτικού προγραμματισμού, καθώς η εφαρμογή αντιδρά αυτόματα στις αλλαγές των δεδομένων, προσφέροντας μια συνεχή και ομαλή ροή χωρίς την ανάγκη για παραδοσιακή ανανέωση της σελίδας.

Στο επίπεδο του backend, η υλοποίηση του **Node.js Proxy Server** με τη χρήση του **Express** αποτέλεσε ένα από τα σημαντικότερα μαθησιακά κέρδη του project. Η επίλυση κρίσιμων ζητημάτων, όπως οι περιορισμοί **CORS** και η ασφαλής διαχείριση των **API Keys**, ανέδειξε την αναγκαιότητα ενός ενδιάμεσου επιπέδου (middleware) στην επικοινωνία με εξωτερικές πηγές δεδομένων. Η χρήση της JavaScript τόσο στο frontend όσο και στο backend (Unified Language approach) μείωσε το γνωστικό φορτίο και επέτρεψε την ταχύτερη ανάπτυξη, καθώς η δομή των δεδομένων (JSON) παρέμεινε ενιαία σε όλη τη διαδρομή από το API έως την οθόνη του χρήστη.

Τελικά, η ενασχόληση με αυτό το τεχνολογικό σχήμα καλλιέργησε την ικανότητα επίλυσης προβλημάτων σε πραγματικό χρόνο. Από τη διαχείριση ασύγχρονων κλήσεων με το **Axios** έως τον σχεδιασμό μιας αποκρινόμενης (responsive) διεπαφής με το **Tailwind CSS**, το έργο "Sport90" λειτούργησε ως μια ολοκληρωμένη πρακτική άσκηση στις βέλτιστες πρακτικές του

σύγχρονου web development. Η κατανόηση του πώς τα δεδομένα ταξιδεύουν, μετασχηματίζονται και τελικά παρουσιάζονται, αποτελεί το σημαντικότερο εφόδιο που αποκομίστηκε από τη συγκεκριμένη υλοποίηση.

5.2Διαχείριση Σφαλμάτων (Error Handling): Πώς αντιμετωπίζονται τα προβλήματα σύνδεσης ή τα όρια του API

Η διαχείριση σφαλμάτων στο "Sport90" δεν αποτελεί απλώς έναν μηχανισμό αποσφαλμάτωσης, αλλά μια ολοκληρωμένη στρατηγική διατήρησης της εμπιστοσύνης του χρήστη. Στο περιβάλλον των σύγχρονων web εφαρμογών, η εξάρτηση από εξωτερικές πηγές δεδομένων, όπως το **Football-Data.org API**, εγκυμονεί κινδύνους που σχετίζονται με την αστάθεια του δικτύου ή τους περιορισμούς των παρόχων. Η εφαρμογή αντιμετωπίζει αυτά τα ζητήματα μέσω ενός πολυεπίπεδου συστήματος ελέγχου που ξεκινά από τον διακομιστή μεσολάβησης και καταλήγει στη διεπαφή χρήστη, διασφαλίζοντας ότι καμία αποτυχία σύνδεσης δεν θα οδηγήσει σε ολοκληρωτική κατάρρευση του συστήματος.

Μία από τις μεγαλύτερες προκλήσεις που αντιμετωπίστηκαν είναι τα **όρια του API (Rate Limits)**, καθώς η δωρεάν βαθμίδα της υπηρεσίας επιτρέπει περιορισμένο αριθμό κλήσεων ανά λεπτό. Ο Proxy Server λειτουργεί ως ο πρώτος αμυντικός μηχανισμός, καταγράφοντας τα σφάλματα τύπου "429 Too Many Requests" στο τερματικό. Αντί η εφαρμογή να "παγώσει", ο διακομιστής επιστρέφει ένα ελεγχόμενο μήνυμα σφάλματος στο frontend. Στην πλευρά του πελάτη, η χρήση των δομών **try...catch** επιτρέπει στην React να "συλλάβει" αυτές τις αστοχίες και να ενημερώσει την κατάσταση errorStatus. Αυτό αποτρέπει την εκτέλεση κώδικα που θα προσπαθούσε να επεξεργαστεί δεδομένα που δεν έφτασαν ποτέ, προστατεύοντας τη σταθερότητα του component.

Η οπτική επικοινωνία του σφάλματος προς τον χρήστη υλοποιείται μέσω της **υπό συνθήκη προβολής (conditional rendering)**. Όταν εντοπιστεί πρόβλημα σύνδεσης ή υπέρβαση ορίων, η εφαρμογή αντικαθιστά αυτόματα τον πίνακα δεδομένων με ένα φιλικό μήνυμα ενημέρωσης, το οποίο εξηγεί την κατάσταση και προτείνει στον χρήστη να ανανεώσει τη σελίδα μετά από λίγο. Αυτή η προσέγγιση μετατρέπει μια τεχνική αποτυχία σε μια καθοδηγούμενη εμπειρία, αποφεύγοντας τις κενές σελίδες ή τα ακατανόητα μηνύματα του browser. Με αυτόν τον τρόπο, το "Sport90" επιδεικνύει υψηλό επίπεδο ωριμότητας στον σχεδιασμό του, αντιμετωπίζοντας τα σφάλματα ως αναπόσπαστο μέρος του κύκλου ζωής μιας εφαρμογής.

5.3Πλεονεκτήματα & Περιορισμοί της Εφαρμογής

Η εφαρμογή **"Sport90"** διακρίνεται για μια σειρά από **πλεονεκτήματα** που την καθιστούν ένα σύγχρονο εργαλείο αθλητικής πληροφόρησης. Το σημαντικότερο όφελος είναι η **ταχύτητα και η απόκριση** της διεπαφής, καθώς η αρχιτεκτονική Single Page Application (SPA) επιτρέπει την ακαριαία εναλλαγή μεταξύ βαθμολογιών και προφίλ ομάδων χωρίς αναμονή για φόρτωση νέων σελίδων. Επιπλέον, η χρήση του **Proxy Server** προσφέρει ένα ισχυρό επίπεδο ασφάλειας, προστατεύοντας τα ευαίσθητα κλειδιά API και επιλύοντας οριστικά τα προβλήματα συμβατότητας CORS που συχνά εμποδίζουν τις web εφαρμογές από το να αντλούν δεδομένα απευθείας από εξωτερικές πηγές. Τέλος, ο **Responsive σχεδιασμός** διασφαλίζει ότι η πληροφορία παραμένει προσβάσιμη και ευανάγνωστη σε κάθε τύπο συσκευής, προσφέροντας μια dashboard εμπειρία σε desktop και μια συμπαγή, λειτουργική προβολή σε κινητά.

Ωστόσο, η εφαρμογή υπόκειται σε συγκεκριμένους **τεχνικούς περιορισμούς**, οι οποίοι απορρέουν κυρίως από την εξάρτηση από το Football-Data.org API. Ο κυριότερος περιορισμός αφορά τα **όρια ρυθμού αιτημάτων (Rate Limits)** της δωρεάν βαθμίδας χρήσης, γεγονός που σημαίνει ότι σε περιπτώσεις πολύ συχνών ανανεώσεων από τον χρήστη, η εφαρμογή ενδέχεται να εμφανίσει προσωρινή αδυναμία ανάκτησης νέων δεδομένων. Επίσης, η **στατικότητα των δεδομένων** εξαρτάται από τον ρυθμό ενημέρωσης της εξωτερικής πηγής· εάν το API καθυστερήσει να ενημερώσει ένα σκορ, η εφαρμογή θα εμφανίσει την τελευταία διαθέσιμη πληροφορία, χωρίς να έχει τη δυνατότητα αυτόνομης επαλήθευσης.

Ένας επιπλέον περιορισμός αφορά την **έλλειψη μόνιμης αποθήκευσης (Persistence)** στην πλευρά του client. Καθώς η εφαρμογή βασίζεται αποκλειστικά στην τρέχουσα κατάσταση (state) της React, κάθε πλήρης ανανέωση της σελίδας (hard refresh) αναγκάζει το σύστημα να εκτελέσει εκ νέου όλα τα αιτήματα προς τον Proxy Server, αυξάνοντας την κατανάλωση πόρων. Παρά τους περιορισμούς αυτούς, το "Sport90" επιτυγχάνει τον βασικό του στόχο: να αποτελέσει μια αξιόπιστη, ασφαλής και αισθητικά άρτια πλατφόρμα παρουσίασης αθλητικών δεδομένων, θέτοντας ταυτόχρονα τις βάσεις για μελλοντικές βελτιώσεις που θα αμβλύνουν αυτά τα σημεία.

5.4Προτάσεις για Μελλοντική Βελτίωση

Η πρώτη και πιο σημαντική κατεύθυνση για τη μελλοντική αναβάθμιση του "Sport90" αφορά την **ενσωμάτωση συστήματος μόνιμης αποθήκευσης (Database Integration)**. Η χρήση μιας βάσης δεδομένων, όπως η **MongoDB** ή η **PostgreSQL**, θα επέτρεπε την υλοποίηση συστήματος εγγραφής χρηστών (User Accounts). Με αυτόν τον τρόπο, οι φίλαθλοι θα μπορούσαν να δημιουργούν το δικό τους προφίλ, να αποθηκεύουν τις αγαπημένες τους ομάδες και να λαμβάνουν εξατομικευμένες ειδοποιήσεις για τα αποτελέσματα των αγώνων που τους ενδιαφέρουν, μετατρέποντας την εφαρμογή από ένα εργαλείο αναζήτησης σε έναν προσωπικό αθλητικό βοηθό.

Μια δεύτερη τεχνική βελτίωση εστιάζει στη **βελτιστοποίηση της απόδοσης μέσω Caching**. Λόγω των περιορισμών (rate limits) του Football-Data API, η υλοποίηση ενός μηχανισμού προσωρινής αποθήκευσης στον Proxy Server (π.χ. με τη χρήση **Redis**) θα επέτρεπε στην εφαρμογή να σερβίρει τα δεδομένα των βαθμολογιών ταχύτερα, μειώνοντας τις κλήσεις προς το εξωτερικό API. Αυτό όχι μόνο θα καθιστούσε το "Sport90" πιο "ανθεκτικό" στις περιπτώσεις υψηλής κίνησης, αλλά θα διασφάλιζε και την απρόσκοπτη λειτουργία του ακόμη και αν η δωρεάν βαθμίδα του API εξαντληθεί προσωρινά.

Τέλος, η εμπειρία χρήστη μπορεί να αναβαθμιστεί μέσω της **χρήσης WebSockets για ζωντανή ενημέρωση (Real-time updates)**. Στην τρέχουσα έκδοση, ο χρήστης πρέπει να ανανεώσει τη σελίδα ή να πλοηγηθεί εκ νέου για να δει την αλλαγή σε ένα σκορ. Με την υιοθέτηση τεχνολογιών όπως το **Socket.io**, η εφαρμογή θα μπορούσε να "σπρώχνει" (push) τις αλλαγές των σκορ σε πραγματικό χρόνο στην οθόνη του χρήστη, προσφέροντας την ένταση της live παρακολούθησης. Σε συνδυασμό με την προσθήκη **περισσότερων πηγών δεδομένων** (π.χ. για μεταγραφικά νέα ή αναλυτικά στατιστικά παικτών), το "Sport90" θα μπορούσε να εξελιχθεί σε έναν πλήρη κόμβο αθλητικής πληροφόρησης που θα ανταγωνίζεται επαγγελματικές πλατφόρμες του χώρου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] React Documentation (Official). "Thinking in React: Component-Based Architecture and State Management." Meta Open Source. https://react.dev/learn/thinking-in-react

[2] Tailwind CSS Documentation. "Utility-First Fundamentals and Responsive Design Patterns." Tailwind Labs Inc. https://tailwindcss.com/docs/utility-first

[3] [3] Fielding, R. T., & Taylor, R. N. "Principled Design of the Modern Web Architecture: RESTful API Development and JSON Data Exchange." Διαθέσιμο στο επίσημο documentation του HTTP/REST.

[4] Node.js Foundation. "Express Middleware and Security Best Practices: Handling CORS and Protecting API Credentials." https://nodejs.org/en/learn/getting-started/introduction-to-nodejs

[5] Vite Guide. "Why Vite? Next Generation Frontend Tooling." https://vitejs.dev/guide/why.html

[6] ESLint Documentation. "Getting Started with ESLint: Static Code Analysis for JavaScript." https://eslint.org/docs/latest/use/getting-started