

ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ  
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΩΝ  
ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

---

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ ΓΙΑ  
ΤΟ ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2024-2025

---

---

ΟΜΑΔΑ 5380-5226

---

ΙΩΑΝΝΗΣ ΦΙΛΛΗΣ, ΑΜ: 5380

ΚΩΝΣΤΑΝΤΙΝΟΣ ΖΩΗΣ, ΑΜ: 5226

---

ΤΕΛΙΚΗ ΑΝΑΦΟΡΑ

ΜΑΪΟΣ 2025

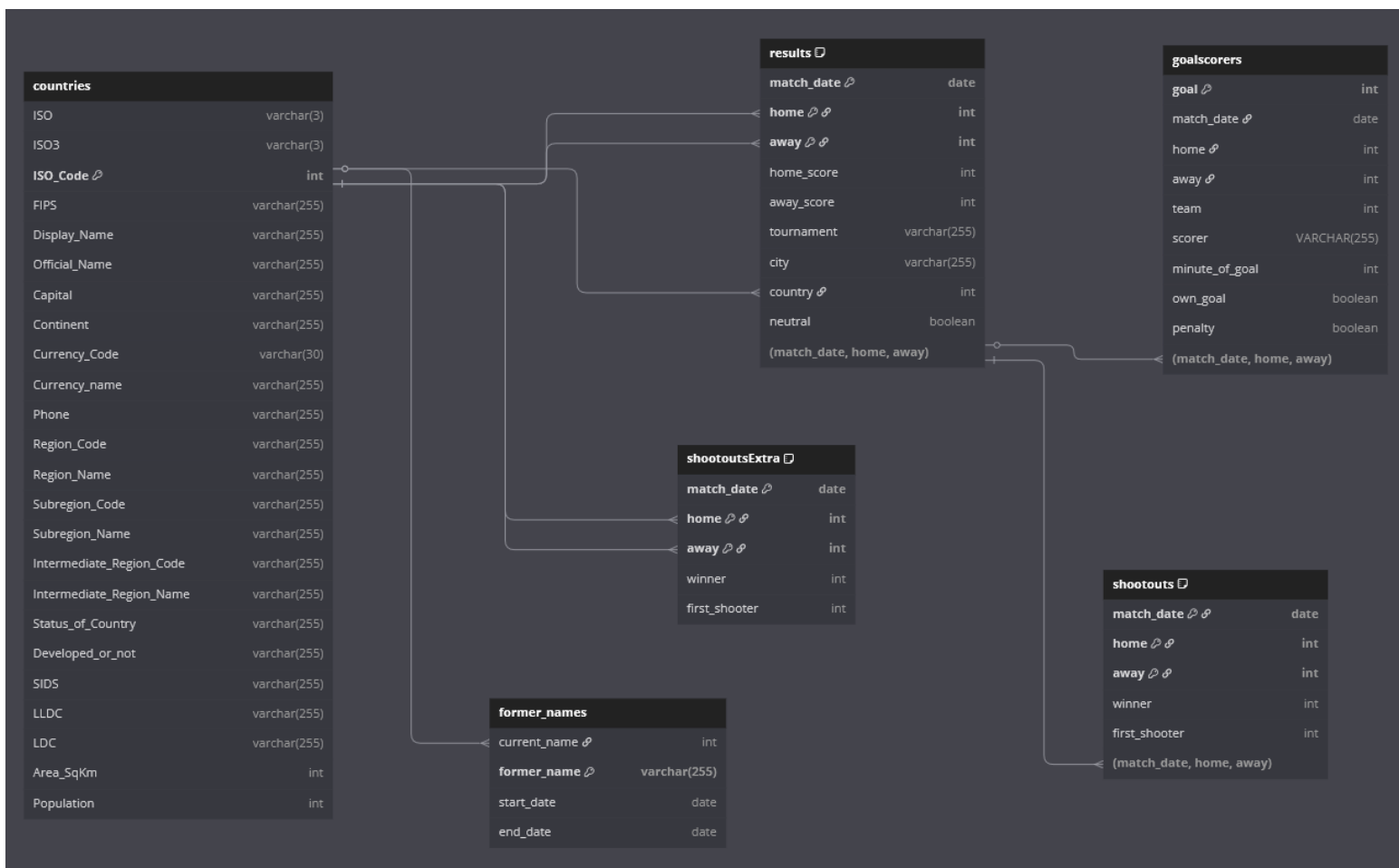
## ΙΣΤΟΡΙΚΟ ΠΡΟΗΓΟΥΜΕΝΩΝ ΕΚΔΟΣΕΩΝ

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
2025/05/22	1.0	Προστέθηκαν οι κύριες πληροφορίες για την σχεδίαση και ρύθμιση της βάσης δεδομένων, καθώς και για την αρχιτεκτονική της εφαρμογής	Κωνσταντίνος Ζώης, Ιωάννης Φίλλης
2025/05/24	1.1	Προστέθηκαν το σχεσιακό σχήμα και τα UML διαγράμματα	Κωνσταντίνος Ζώης, Ιωάννης Φίλλης
2025/05/25	1.2	Προστέθηκε ο σχολιασμός της οργάνωσης της εφαρμογής	Κωνσταντίνος Ζώης, Ιωάννης Φίλλης
2025/05/26	1.3	Προστέθηκαν οι πληροφορίες για την ενότητα 3	Κωνσταντίνος Ζώης, Ιωάννης Φίλλης
2025/05/27	2.0	Τελικές Προσθήκες και διορθώσεις	Κωνσταντίνος Ζώης, Ιωάννης Φίλλης

## 1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Στην παρούσα ενότητα περιγράφονται τα σχήματα της βάσης δεδομένων που χρησιμοποιούνται στο Project.

### 1.1 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΛΟΓΙΚΟ ΕΠΙΠΕΔΟ



Σχήμα 1.1 Σχεσιακό σχήμα της βάσης δεδομένων του συστήματος

Ας δούμε και λίγο πιο αναλυτικά τους πίνακες που έχουμε δημιουργήσει.

```
4  ● create table countries(  
5      ISO varchar(3),  
6      ISO3 varchar(3),  
7      ISO_Code int,  
8      FIPS varchar(255),  
9      Display_Name varchar(255),  
10     Official_Name varchar(255),  
11     Capital varchar(255),  
12     Continent varchar(255),  
13     Currency_Code varchar(30),  
14     Currency_name varchar(255),  
15     Phone varchar(255),  
16     Region_Code varchar(255),  
17     Region_Name varchar(255),  
18     Subregion_Code varchar(255),  
19     Subregion_Name varchar(255),  
20     Intermediate_Region_Code varchar(255),  
21     Intermediate_Region_Name varchar(255),  
22     Status_of_Country varchar(255),  
23     Developed_or_not varchar(255),  
24     SIDS varchar(255),  
25     LLDC varchar(255),  
26     LDC varchar(255),  
27     Area_SqKm int,  
28     Population int,  
29     Primary key (ISO_Code)
```

**Εικόνα 1.1: πίνακας countries**

Στον συγκεκριμένο πίνακα βάζουμε ως columns τις στήλες που εμφανίζονται στο csv αρχείο και χρησιμοποιούμε primary key το ISO\_CODE.

```
32  ● create table results(  
33      match_date date,  
34      home int,  
35      away int,  
36      home_score int,  
37      away_score int,  
38      tournament varchar(255),  
39      city varchar(255),  
40      country int,  
41      neutral boolean,  
42      primary key (match_date,home,away),  
43      foreign key(home) references countries(ISO_Code) ON DELETE CASCADE ON UPDATE CASCADE,  
44      foreign key(away) references countries(ISO_Code) ON DELETE CASCADE ON UPDATE CASCADE,  
45      foreign key(country) references countries(ISO_Code) ON DELETE CASCADE ON UPDATE CASCADE  
46  )ENGINE=InnoDB;  
47
```

**Εικόνα 1.2: πίνακας results**

Ο επόμενος πίνακας είναι ο πίνακας των results. Χρησιμοποιούμε ως Primary Key την τριπλέτα match\_date, home, away και έχουμε ξένα κλειδιά στον πίνακα countries.

```
48 • create table shootouts(  
49     match_date date,  
50     home int,  
51     away int,  
52     winner int,  
53     first_shooter int,  
54     primary key (match_date,home,away),  
55     foreign key(match_date,home,away) references results(match_date,home,away) ON DELETE CASCADE ON UPDATE CASCADE  
56 )ENGINE=InnoDB;  
57  
58 • create table shootoutsExtra(  
59     match_date date,  
60     home int,  
61     away int,  
62     winner int,  
63     first_shooter int,  
64     primary key (match_date,home,away),  
65     foreign key(home) references countries(ISO_Code) ON DELETE CASCADE ON UPDATE CASCADE,  
66     foreign key(away) references countries(ISO_Code) ON DELETE CASCADE ON UPDATE CASCADE  
67 )ENGINE=InnoDB;  
68
```

**Εικόνα 1.3: πίνακες shootouts/shootoutsExtra**

Έπειτα έχουμε τους πίνακες shootouts και shootoutsExtra. Και οι δύο πίνακες έχουν ως Primary Key πάλι την τριάδα match\_date, home, away ωστόσο η διαφορά τους είναι στα ξένα κλειδιά. Το shootouts έχει ξένα κλειδιά στον πίνακα results ενώ το shootoutsExtra έχει στον πίνακα countries. Αυτό συμβαίνει διότι το shootoutsExtra είναι ένας έξτρα πίνακας που δημιουργήσαμε προκειμένου να απομονώσουμε το ματς που υπάρχει στον αρχείο shootouts.csv αλλά όχι στο results.csv.

```
69 • CREATE TABLE goalscorers (  
70     goal int primary key auto_increment,  
71     match_date date,  
72     home int,  
73     away int,  
74     team int,  
75     scorer VARCHAR(255),  
76     minute_of_goal int,  
77     own_goal boolean,  
78     penalty boolean,  
79     foreign key(match_date,home,away) references results(match_date,home,away) ON DELETE CASCADE ON UPDATE CASCADE  
80 )ENGINE=InnoDB;  
81
```

**Εικόνα 1.4: πίνακας goalscorers**

Επόμενος πίνακας είναι ο πίνακας των goalscorers. Αυτός έχει Primary Key με auto increment και ξένα κλειδιά στον πίνακα results την τριάδα match\_date, home, away.

```
82 • create table former_names(  
83     current_name int,  
84     former_name varchar(255),  
85     start_date date,  
86     end_date date,  
87     primary key (former_name),  
88     foreign key(current_name) references countries(ISO_Code)  
89 );  
90
```

**Εικόνα 1.5: πίνακας former\_names**

---

## 1.2 ΣΧΕΣΙΑΚΟ ΣΧΗΜΑ ΣΕ ΦΥΣΙΚΟ ΕΠΙΠΕΔΟ

---

Αφού λοιπόν έχουμε στήσει την βάση μας και έχουμε φορτώσει τα δεδομένα μας, παρακάτω αναφέρουμε κάποιες από τις παραμέτρους που έχουμε εφαρμόσει.

---

### 1.2.1 ΡΥΘΜΙΣΗ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ DBMS

---

Ως storage engine έχουμε θέσει την INNO\_DB.

---

### 1.2.2 ΡΥΘΜΙΣΗ ΤΟΥ ΦΥΣΙΚΟΥ ΣΧΗΜΑΤΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

---

Για να διευκολύνουμε λίγο την υλοποίηση των πολύπλοκων queries που είχαμε να δημιουργήσουμε αποφασίσαμε να χρησιμοποιήσουμε ορισμένα views προκειμένου να γίνει πιο εύκολη η υλοποίηση. Αν και σε ορισμένες εφαρμογές, μπορούν να κάνουν το πρόγραμμα αργό, στην συγκεκριμένη εφαρμογή, ο όγκος πληροφορίας δεν είναι τόσο μεγάλος ώστε να δημιουργεί προβλήματα, έτσι η επιλογή και χρήση των views καθίσταται ιδανική.

Το πρώτο view που χρησιμοποιήσαμε ονομάζεται **country\_home\_away\_summary**. Σε αυτό χρησιμοποιούμε WITH statements για να εξάγουμε τις πληροφορίες από τους διάφορους πίνακες. Χρησιμοποιούμε επίσης και μία συνάρτηση η οποία ονομάζεται COALESCE. Η λειτουργία της είναι να παίρνει ορίσματα και να κρατάει την πρώτη non-null τιμή που θα βρει. Ο λόγος που την χρησιμοποιήσαμε είναι ώστε να μπορούμε να κρατήσουμε την βάση όσο πιο «καθαρή» γίνεται, δηλαδή να μην έχει null τιμές. Στο τέλος, αυτό που κάνουμε είναι να έχουμε ως columns πληροφορίες για την κάθε ομάδα όπως νίκες, ισοπαλίες, ήττες και λοιπά αναλυτικά στατιστικά.

Το επόμενο view είναι το **team\_yearly\_results**. Αυτό, πάλι με την χρήση διαφόρων with statements, κρατάει πληροφορία για κάθε χώρα και κάθε χρονιά. Ενδεικτικά, πληροφορίες που κρατάει είναι (ανά χρονιά) αποτελέσματα αγώνων, αποτελέσματα σπασμένα ανάλογα με το που έπαιξαν γκολ που σκόραραν, που δέχτηκαν κτλ.

Το επόμενο view που χρησιμοποιήθηκε είναι το **match\_information**. Αυτό, για κάθε ματς μας δίνει ότι πληροφορία μπορεί να είναι χρήσιμη, όπως ποιες ομάδες αγωνίστηκαν, που πραγματοποιήθηκε ο αγώνας, πληροφορίες για τις ομάδες που αγωνίστηκαν, πληροφορίες για τον τόπο διεξαγωγής του αγώνα και λοιπές πληροφορίες σχετικά με τον αγώνα.

Άλλο ένα view που χρησιμοποιήθηκε είναι το **order\_by\_points** το οποίο έχει μία κατάταξη με βάση τους πόντους που έχει συλλέξει η κάθε ομάδα. Τρεις πόντους παίρνει για κάθε

νίκη και ένα για κάθε ισοπαλία. Επίσης έχουμε δημιουργήσει και το ***order\_by\_points\_per\_year*** το οποίο έχει πάλι κατάταξη ανά χρονιά τώρα.

Άλλο ένα view που χρησιμοποιήσαμε είναι το ***scatterplot\_country\_stats*** το οποίο χρησιμοποιούμε για να εξάγουμε την πληροφορία που χρειαζόμαστε για να φτιάξουμε τα scatterplots.

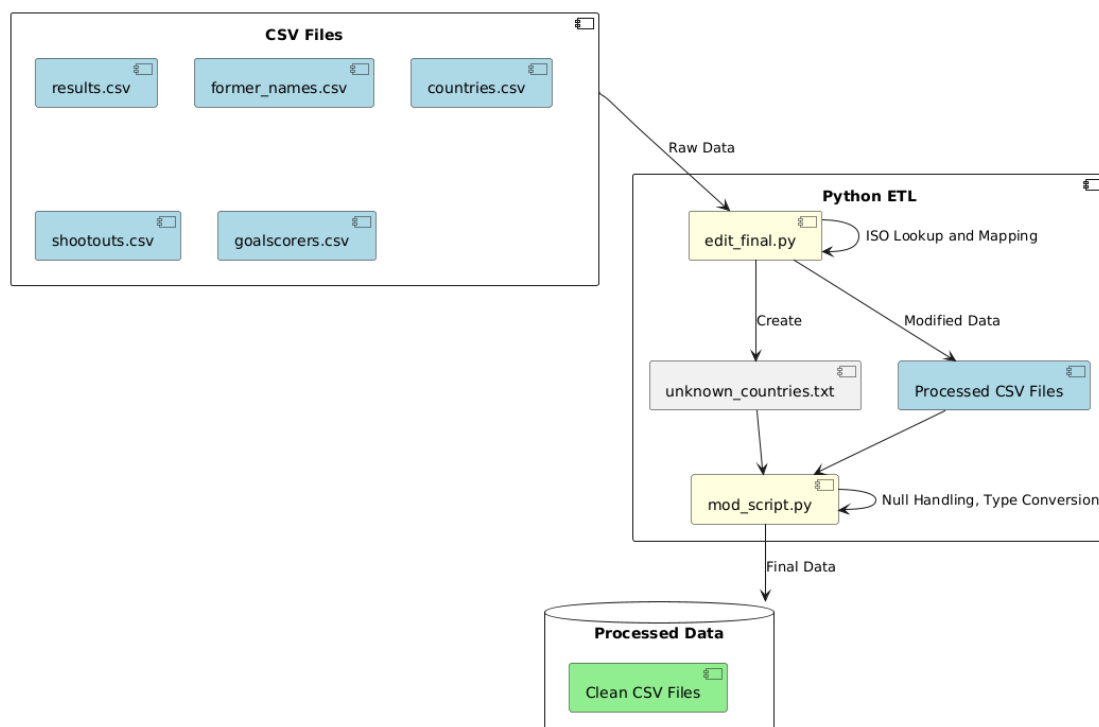
Τέλος, φτιάξαμε το ***main\_info\_of\_player*** που κρατά τις κύριες πληροφορίες για έναν παίκτη.

### 1.2.3 ΡΥΘΜΙΣΗ ΑΣΦΑΛΕΙΑΣ

Δεν έχουμε κάνει κάποια ειδική ρύθμιση σε σχέση με την ασφάλεια.

## 2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

### 2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΔΟΜΗ ETL



Σχήμα 2.1 UML component diagram

Όσον αφορά το κομμάτι της επεξεργασίας των δεδομένων, χρησιμοποιήσαμε την γλώσσα προγραμματισμού `python` προκειμένου να φτιάξουμε τα δεδομένα στην μορφή που θέλουμε. Ένα βασικό πρόβλημα το οποίο αντιμετωπίσαμε ήταν το γεγονός πως πολλές χώρες έχουν αλλάξει όνομα κατά την πάροδο του χρόνου οπότε οι ίδιες χώρες εμφανίζονταν με διαφορετικό όνομα. Στο `dataset` που μας δόθηκε μας δίνονταν χώρες με τα παλιά ονόματά τους.

Ξεκινώντας λοιπόν, αρχικά δημιουργήσαμε το αρχείο **`edit_final.py`**. Εκεί, αρχικά αυτό που κάναμε ήταν να φορτώσουμε τα αρχεία μας σε `dataframes` και να δημιουργήσουμε ένα `map` με τα ονόματα των χωρών ως κλειδιά και τα `ISO_CODE` ως τιμές. Επίσης, δημιουργήσαμε το `country_replacements` το οποίο περιέχει ως κλειδιά ονόματα χωρών (τα παλιά ονόματα τους) και ως τιμές τα `ISO_CODE` τα οποία πρέπει να έχουν. Τα ονόματα προέκυψαν από το αρχείο `former_names.csv` καθώς και από ένα γρήγορο πέρασμα που κάναμε στο αρχείο `results.csv`. Τέλος προσθέτουμε και αυτά στο αρχικό `map` με τις χώρες που δημιουργήσαμε. Φυσικά, θα πρέπει να πάμε και στο αρχείο `countries.csv` και να μετατρέψουμε την κωδικοποίηση σε `utf-8`. Αυτό έχει σαν αποτέλεσμα να μην αναγνωρίζει κάποιους χαρακτήρες οπότε θα πρέπει να πάμε να τους αλλάξουμε όπου χρειάζεται.

Εφόσον έχουμε κάνει όλα αυτά, πάμε και αλλάζουμε τα ονόματα των χωρών στα `dataframes` με την `.replace()`. Έπειτα ξεκινάει μία διαδικασία η οποία θα γίνει σε κάθε `dataframe`. Αυτό που κάνουμε είναι να ψάξουμε και να βρούμε ποιες χώρες στα `dataframes` δεν περιέχονται στο `countries.csv`. Για δικιά μας διευκόλυνση, τους αναθέτουμε ένα προσωρινό `iso_code`. Κάθε μία που βρίσκουμε, την προσθέτουμε στο `map` που δημιουργήσαμε. Τέλος αυτό που κάνουμε είναι ότι πάμε να αναθέσουμε αυτές τις τιμές στα αρχεία ώστε να έχουν όλες οι χώρες ένα `iso_code`, έστω και προσωρινό για αρχή. Αυτή την διαδικασία την κάνουμε και για τα `goalscorers`, `shootouts` ξεχωριστά και ο λόγος είναι για να εξαλείψουμε την πιθανότητα να υπάρχει κάποια ομάδα εκεί που για κάποιο λόγο δεν εμφανίζεται στο `results`. Αφού ολοκληρώσουμε και με αυτό αποθηκεύουμε τα `dataframes` στα αντίστοιχα `csv files`. Ένα σημαντικό πράγμα που κάνουμε είναι ότι γράφουμε σε ένα `txt` αρχείο τις χώρες και τα προσωρινά `iso_codes` έτσι ώστε να μπορούμε αργότερα να τα διορθώσουμε.

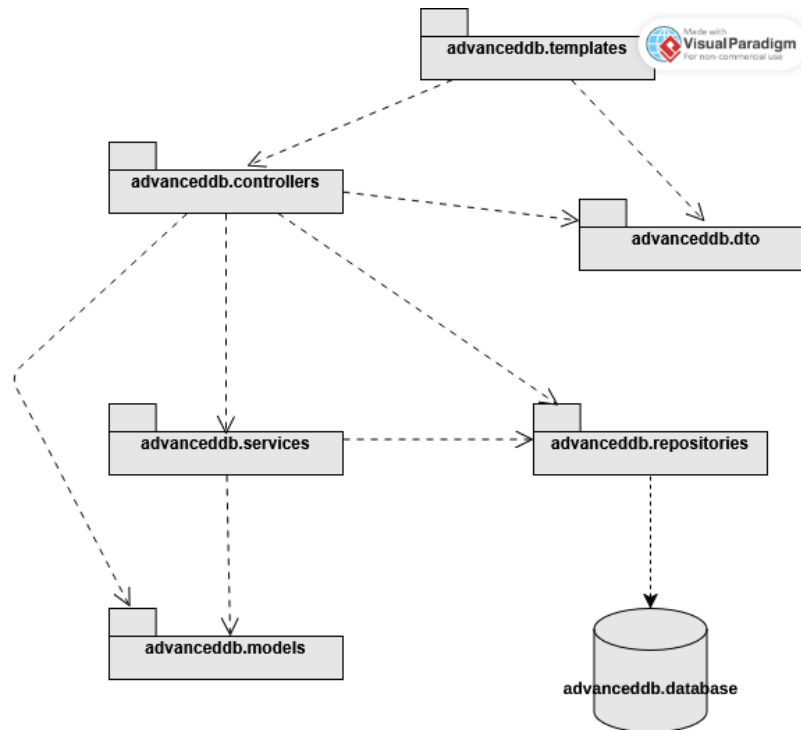
Το δεύτερο και τελευταίο αρχείο που χρησιμοποιούμε είναι το **`mod_script.py`**. Σε αυτό, το πρώτο που τροποποιούμε είναι το `goalscorers`. Στις στήλες `own_goal`, `penalty`, αλλάζουμε τις τιμές προκειμένου να έχει τιμές 1 αν είναι `true` και 0 αν είναι `false`. Επίσης στην στήλη `minute` αυτό που κάνουμε είναι ότι θέτουμε ως -1 όσες εγγραφές δεν έχουν τιμή. Ο λόγος που το κάνουμε αυτό είναι ώστε να εξασφαλίσουμε πως η βάση μας θα έχει το ελάχιστο δυνατό, ιδανικά μηδενικό, πληθος `null` τιμών.

Ομοίως, στο `shootouts` θέτουμε την τιμή -1 όπου στην στήλη `first_shooter` έχουμε κενή τιμή. Τέλος, για τον ακριβώς ίδιο λόγο, στο `countries`, θέτουμε το string "N/A" όπου δεν υπάρχει τιμή.

Αφού έχουμε ολοκληρώσει με όλα αυτά, το μόνο που έχει μείνει είναι να προσθέσουμε όλες τις χώρες και όσον το δυνατόν περισσότερες πληροφορίες μπορούμε να βρούμε για αυτές. Εφόσον οι περισσότερες δεν έχουν `iso code`, αφού πολλές δεν είναι καν πραγματικά κράτη, αυτό που κάναμε είναι ότι βάλαμε δικά μας `iso code` σε αυτές. Παρατηρήσαμε ότι δεν υπάρχουν χώρες με `iso code` πάνω από 900, οπότε αυτό που κάναμε είναι ότι ξεκινήσαμε από το 900 και αυξάνοντας από εκεί δώσαμε τα `iso code`.



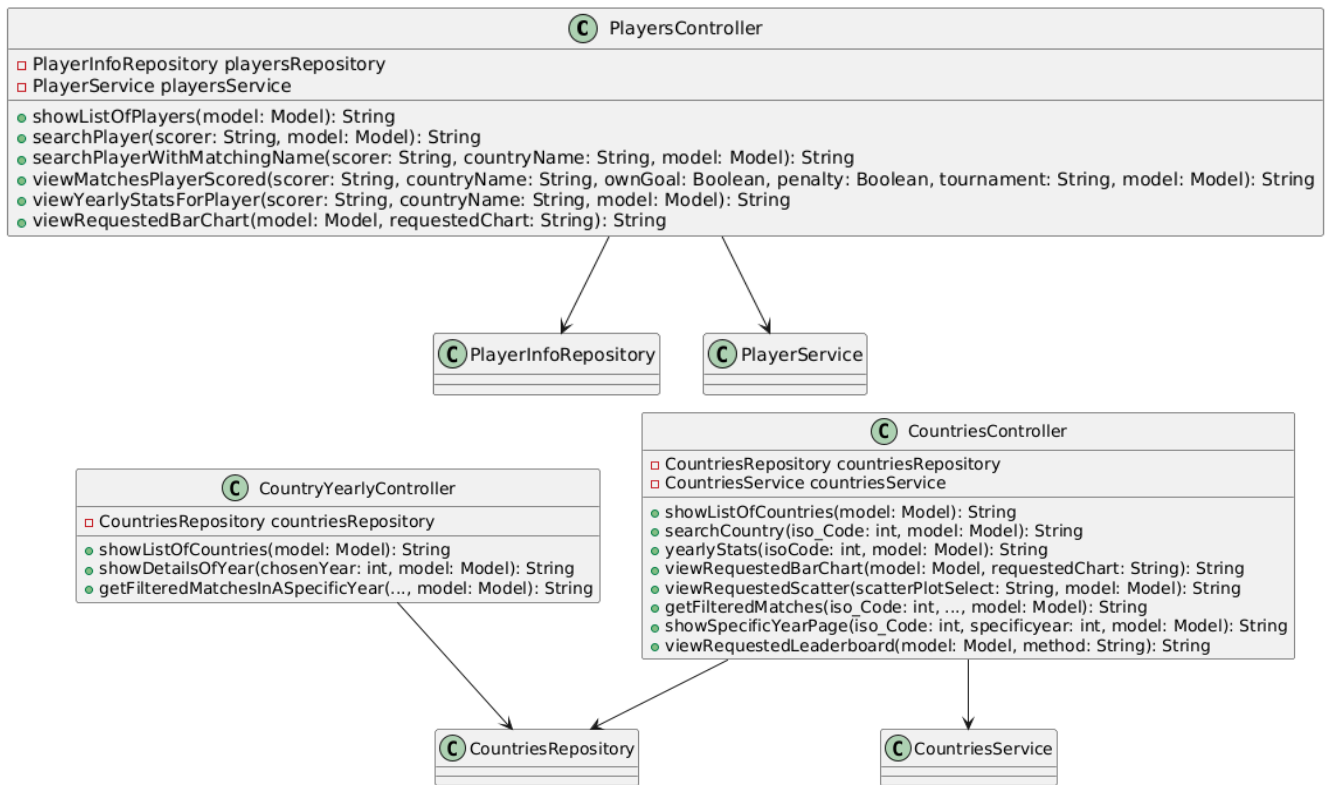
## 2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΠΑΚΕΤΩΝ / ΥΠΟΣΥΣΤΗΜΑΤΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



Σχήμα 2.2 UML Package Diagram

Η αρχιτεκτονική του Project μας χρησιμοποιεί το pattern Model View Controller. Στο πακέτο templates περιέχονται html σελίδες με τις οποίες αλληλοεπιδρά ο χρήστης. Σε αυτές φέρνουμε τα δεδομένα δυναμικά μέσω των controllers, τα οποία λαμβάνουν τα αιτήματα του χρήστη και επικοινωνούν με τα υπόλοιπα πακέτα για να πάρουν τα δεδομένα που χρειάζονται. Τα repositories επικοινωνούν με τη βάση δεδομένων, κάνουν ερωτήματα και αποθηκεύουν τα αποτελέσματα στα dto ή στις οντότητες των models.

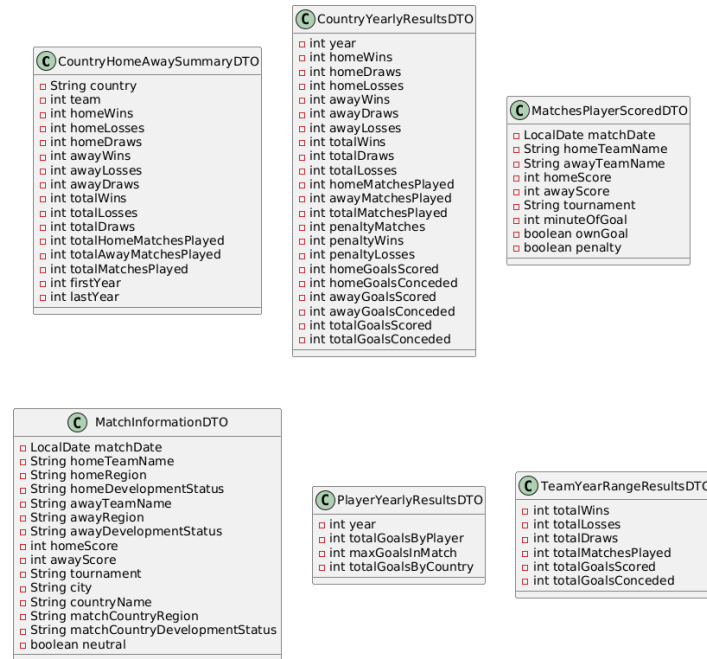
## 2.3 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ ΚΕΝΤΡΙΚΗΣ ΕΦΑΡΜΟΓΗΣ



Σχήμα 2.3 Controllers UML Diagram



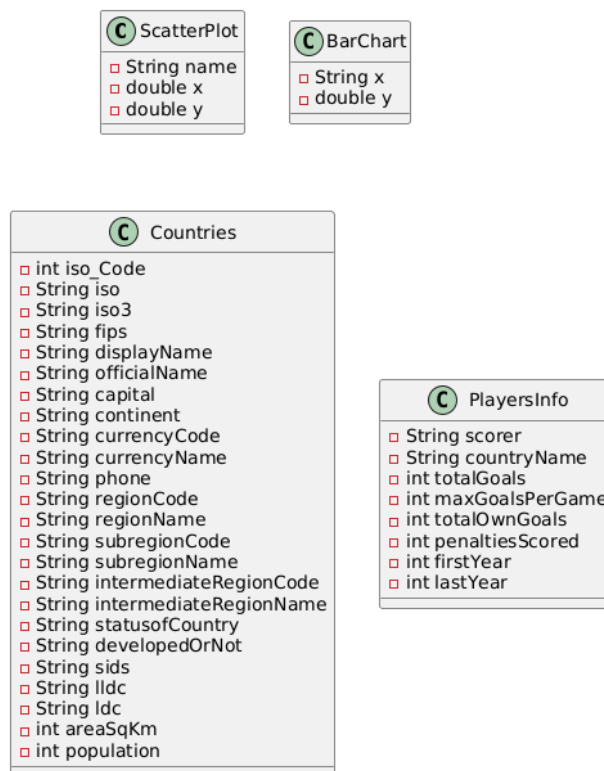
Σχήμα 2.4 Services UML Diagram



Σχήμα 2.5 Dtos UML Diagram



Σχήμα 2.6 Repositories UML Diagram



Σχήμα 2.7 Domain Model UML Diagram

Στο domain model έχουμε τις κύριες οντότητες της εφαρμογής. Στην κλάση Countries έχουμε κάνει map τον πίνακα countries της βάσης και περιέχει τις βασικές πληροφορίες για μία χώρα. Η PlayersInfo έχει τις βασικές πληροφορίες ενός παίκτη και έχει γίνει mapped στο view main\_info\_of\_player που φτιάξαμε. Οι ScatterPlot και BarChart είναι υπεύθυνες για τα δεδομένα των γραφημάτων που δημιουργούνται, με δεδομένα ανάλογα με αυτά που ζήτησε ο χρήστης.

Τα repositories είναι κάποια interface που μέσω του CrudRepository που υλοποιούν, επικοινωνούν με τη βάση δεδομένων, θέτουν queries, λαμβάνουν τα αποτελέσματα και τα αποθηκεύουν στο κατάλληλο DTO ή κλάση του domain model.

Τα δύο services CountriesService και PlayersService είναι κυρίως βοηθητικά για την εμφάνιση των γραφημάτων. Λαμβάνουν τα δεδομένα των controllers σχετικά με το αίτημα του γραφήματος από τον χρήστη και καλούν τα repositories για να γίνει η σωστή ερώτηση στη βάση. Επειδή υπάρχει μια ποικιλία από διαφορετικές εκδοχές των γραφημάτων, η οργάνωση είναι πιο καθαρή έτσι και ευκολότερο επεκτάσιμη.

Οι controllers όπως αναφέραμε και πριν λαμβάνουν αιτήματα από τον χρήστη για τα html templates, δίνουν input στα services, repositories, models ή DTOs, λαμβάνουν τα αποτελέσματα και τα εμφανίζουν δυναμικά στις σελίδες. Υπάρχουν τρεις controllers, για τις χώρες, τις χρονιές και τους παίκτες, δηλαδή ένα για κάθε βασική λειτουργία του Project.

Στο πακέτο dto οι κλάσεις χρησιμοποιούνται ως δομές για να αποθηκεύουν τα δεδομένα των ερωτήσεων από τα repositories. Τα δεδομένα αυτά τα παίρνουμε κυρίως από τα διάφορα views που έχουμε φτιάξει στην βάση. Ειδικότερα, τα CountryYearlyResultsDTO και PlayerYearlyResultsDTO μεταξύ άλλων, χρησιμοποιούνται και για να φέρουν τα δεδομένα στα templates σχετικά με τα line charts.

### 3 ΥΠΟΔΕΙΓΜΑΤΑ ΕΡΩΤΗΣΕΩΝ ΚΑΙ ΑΠΑΝΤΗΣΕΩΝ

Τα ερωτήματα στη βάση δεδομένων γίνονται στα repositories CountriesRepository και PlayerInfoRepository. Με το annotation **@Query** σε κάθε μέθοδο ορίζεται το ερώτημα που θα γίνει στη βάση και τα αποτελέσματα αποθηκεύονται στο dto που έχει οριστεί για την επιστροφή της μεθόδου.

Στα παρακάτω παραδείγματα βλέπουμε το ερώτημα για εύρεση πληροφοριών μιας χώρας με βάση το ISO Code της με τα αποτελέσματα να αποθηκεύονται στο CountryHomeAwaySummaryDTO. Επίσης το ερώτημα της μεθόδου getYearlyResults βρίσκει τα στατιστικά ανά χρονιά μιας χώρας.

```
@Query("SELECT " +
    "r.country, r.team, r.home_wins, r.home_losses, r.home_draws, " +
    "r.away_wins, r.away_losses, r.away_draws, r.total_wins, r.total_losses, " +
    "r.total_draws, r.total_home_matches_played, r.total_away_matches_played, r.total_matches_played, r.first_year ,r.last_year " +
    "FROM country_home_away_summary r " +
    "WHERE r.team = :isoCode")
Optional<CountryHomeAwaySummaryDTO> findByIsoCode(@Param("isoCode")int isoCode);

@Query("SELECT t.year,t.home_wins,t.home_draws, t.home_losses,"
    + "t.away_wins, t.away_draws, t.away_losses,"
    + " t.total_wins, t.total_draws, t.total_losses, "
    + "t.home_matches_played, t.away_matches_played, t.total_matches_played, t.penalty_matches,t.penalty_wins,t.penalty_losses,"
    + "t.home_goals_scored,t.home_goals_conceded,t.away_goals_scored,t.away_goals_conceded,t.total_goals_scored,t.total_goals_conceded "
    + " FROM team_yearly_results t WHERE t.team = :isoCode ORDER BY t.year")
List<CountryYearlyResultsDTO> getYearlyResults(@Param("isoCode") int isoCode);
```

Εικόνα 3.1: Παράδειγμα εκτέλεσης query στο CountriesRepository

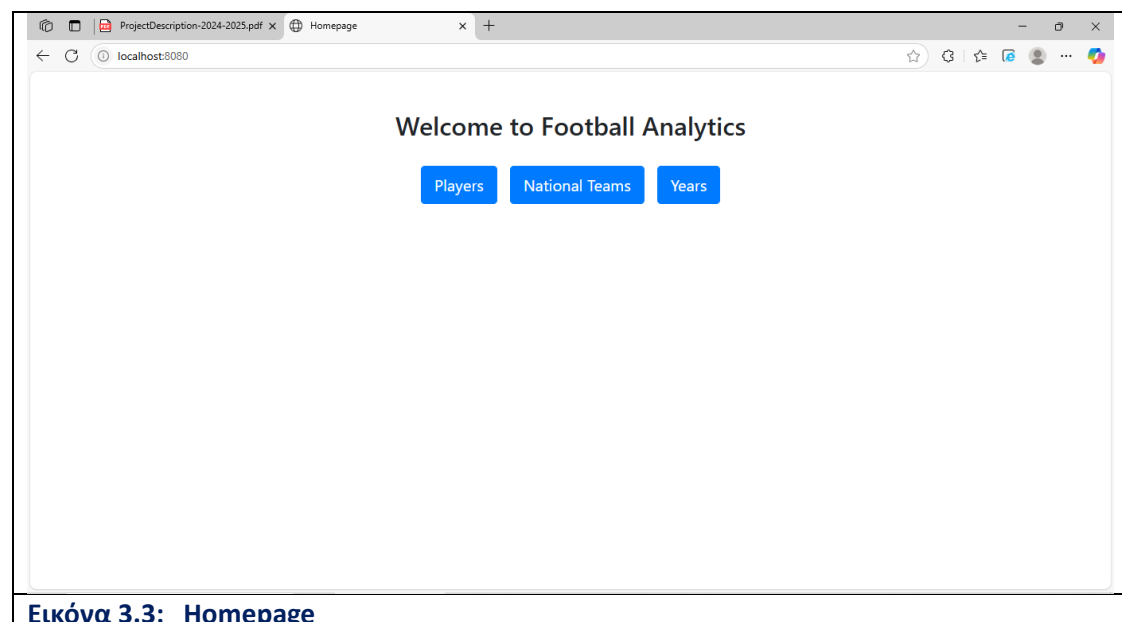
Στο επόμενο παράδειγμα έχουμε το ερώτημα που βρίσκει όλους τους αγώνες που σκόραρε ένας παίκτης. Χρησιμοποιούνται περισσότεροι παράμετροι για να μπορεί ο χρήστης να επιλέγει φίλτρα για αυτούς τους αγώνες.

```
@Query("SELECT m.match_date, m.home_team_name, m.away_team_name, m.home_score, m.away_score, m.tournament, "
+ "g.minute_of_goal, g.own_goal, g.penalty "
+ "FROM match_information m JOIN goalscorers g ON g.match_date = m.match_date "
+ "AND (g.team = m.home_code OR g.team = m.away_code) "
+ "WHERE g.scorer = :scorer "
+ "AND ((g.team = m.home_code AND m.home_team_name = :countryName) "
+ "OR (g.team = m.away_code AND m.away_team_name = :countryName)) "
+ "AND (:ownGoal IS NULL OR :ownGoal = ' ' OR (g.own_goal = :ownGoal)) "
+ "AND (:penalty IS NULL OR :penalty = ' ' OR (g.penalty = :penalty)) "
+ "AND (:tournament IS NULL OR :tournament = ' ' OR m.tournament = :tournament)")
List<MatchesPlayerScoredDTO> findMatches(@Param("scorer") String scorer, @Param("countryName") String countryName, @Param("ownGoal")
Boolean ownGoal, @Param("penalty") Boolean penalty, @Param("tournament") String tournament);
```

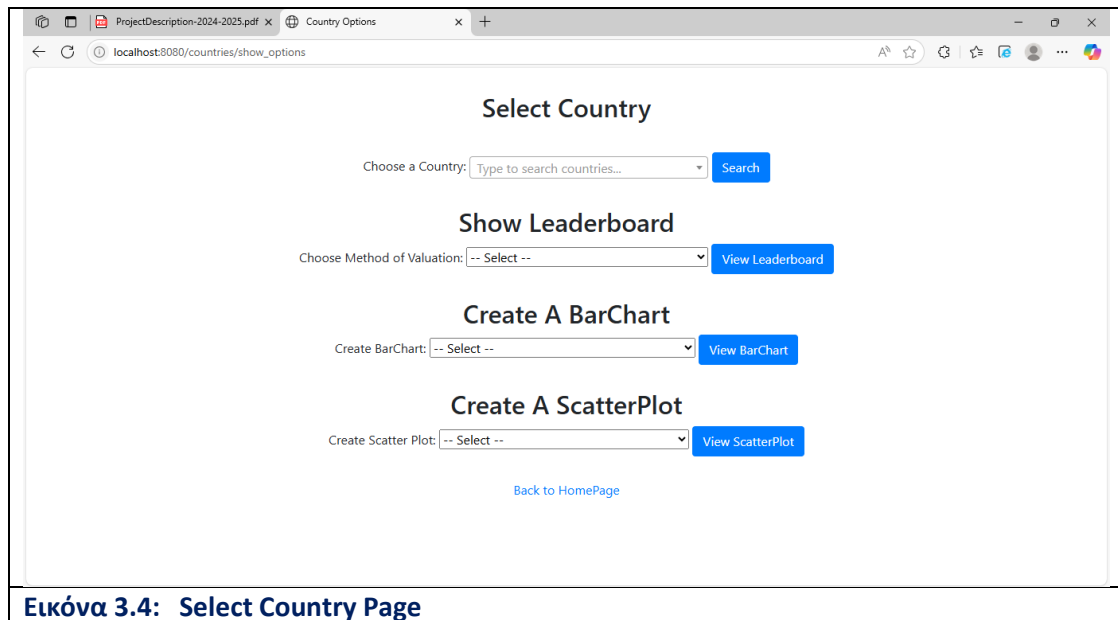
**Εικόνα 3.2: Παράδειγμα εκτέλεσης query στο PlayerInfoRepository**

Ακολουθούν κάποια screenshot από τις λειτουργίες της εφαρμογής.

Μπαίνοντας κανείς στην εφαρμογή το πρώτο πράγμα που αντικρίζει είναι η κύρια σελίδα.

**Εικόνα 3.3: Homepage**

Αν επιλέξει το κουμπί National Teams, θα βρεθεί στο παρακάτω μενού. Από εκεί, έχει τις συγκεκριμένες επιλογές.

**Εικόνα 3.4: Select Country Page**

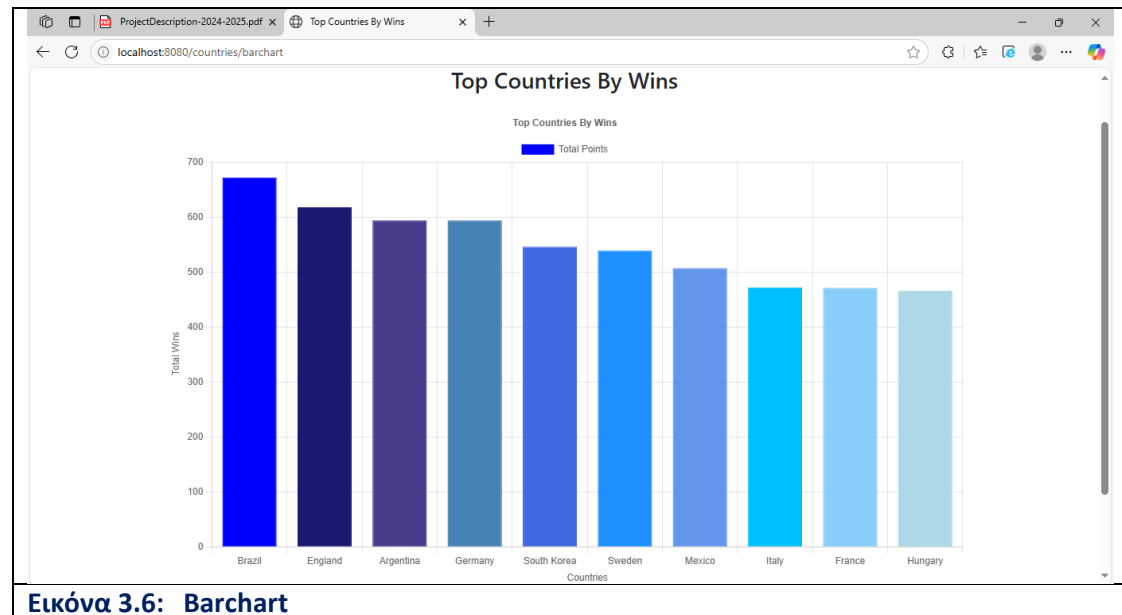
Μπορεί να επιλέξει το Leaderboard με κάποιο από τα διαθέσιμα κριτήρια.

The screenshot shows a web browser window with the URL `localhost:8080/countries/leaderboard`. The page is titled "Leaderboard" and contains a table with the following data:

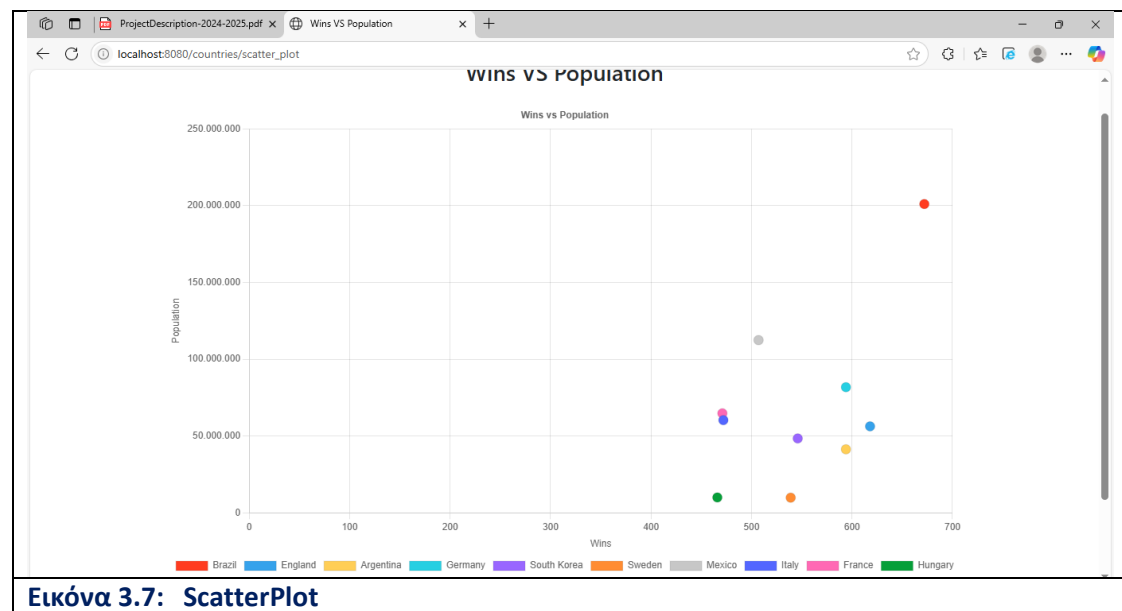
Ranking	Country Name	Score
1	Brazil	2215.0
2	England	2099.0
3	Argentina	2017.0
4	Germany	1986.0
5	South Korea	1863.0
6	Sweden	1841.0
7	Mexico	1732.0
8	Italy	1643.0
9	Hungary	1616.0
10	France	1596.0
11	Ireland	1581.0

**Εικόνα 3.5: Leaderboard**

Μπορεί να επιλέξει επίσης το Barchart με κάποιο από τα διαθέσιμα κριτήρια.

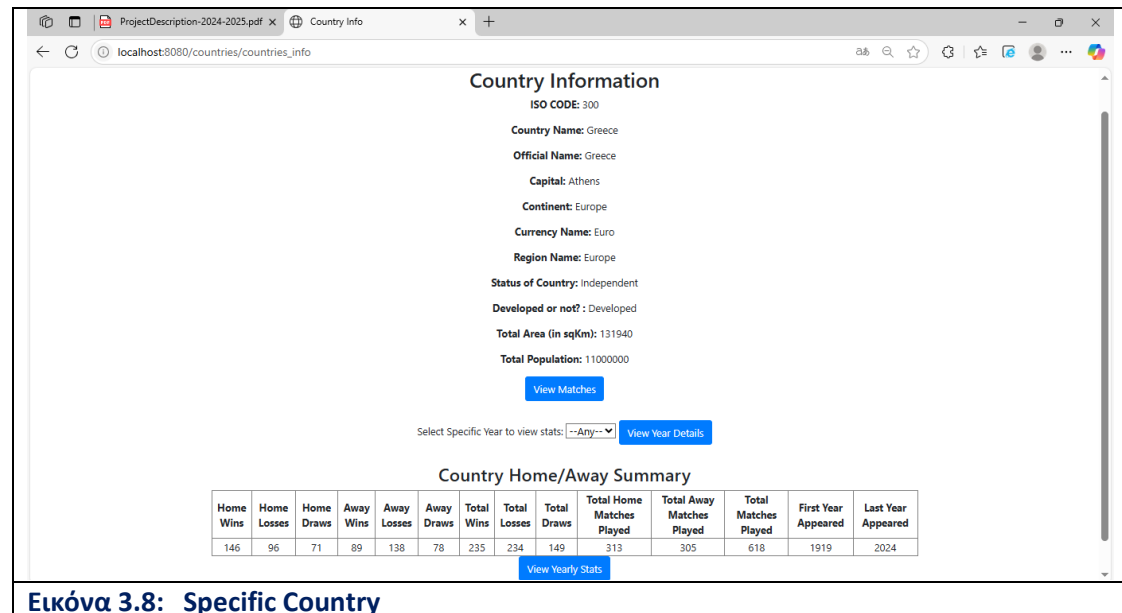


Τέλος μπορεί να επιλέξει το scatterplot με βάση τα διαθέσιμα κριτήρια.



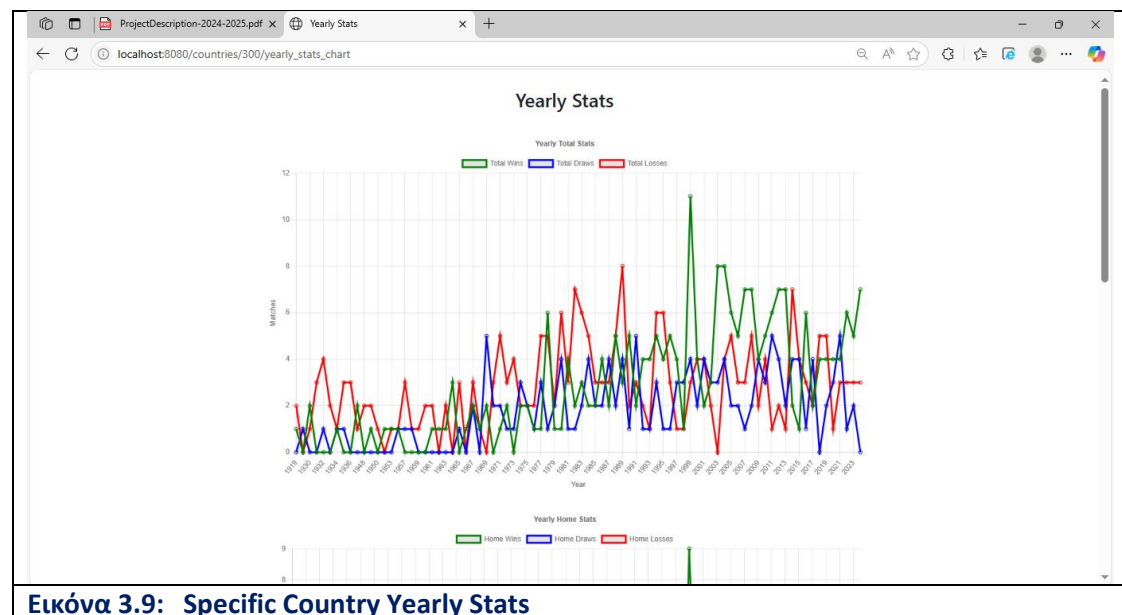
Αναζητώντας κάποια συγκεκριμένη χώρα και επιλέγοντάς την και πατώντας το κουμπί Search θα βρεθεί στην παρακάτω σελίδα όπου θα είναι διαθέσιμα τα παρακάτω στατιστικά.





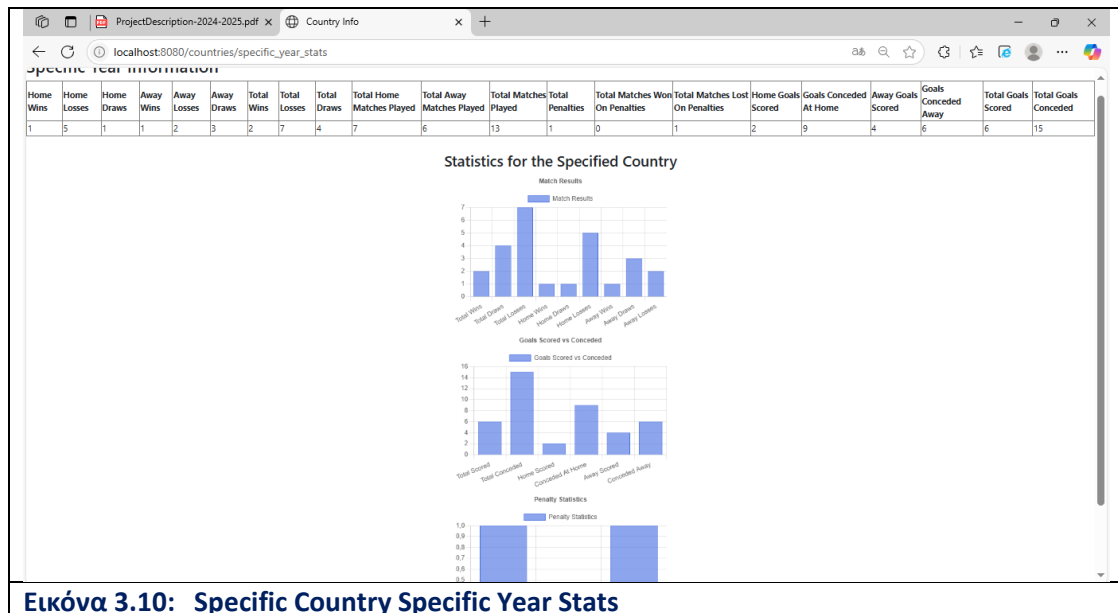
Εικόνα 3.8: Specific Country

Αν επιλέξει το View Yearly Stats θα βρεθεί στην παρακάτω σελίδα.



Εικόνα 3.9: Specific Country Yearly Stats

Αν επιλέξει μία συγκεκριμένη χρονιά, τότε θα βρεθεί στην παρακάτω σελίδα με τα αντίστοιχα στατιστικά.



Εικόνα 3.10: Specific Country Specific Year Stats

Τέλος, μπορεί να δει όλους τους αγώνες της ομάδας και να εφαρμόσει τα παρακάτω φίλτρα.

The screenshot shows a web browser window with the URL `localhost:8080/countries/matches`. The page title is "Specific Country Matches". It features a table of matches and various filters.

**Filters For Match Day**

Opponent Development Status: --Any-- Status: --Any-- Region Of Opponent's Country: --Any--

Tournament: --Any-- Region Of Country: --Any-- Status Of Country: --Any-- Development Status: --Any-- Location: --Any--

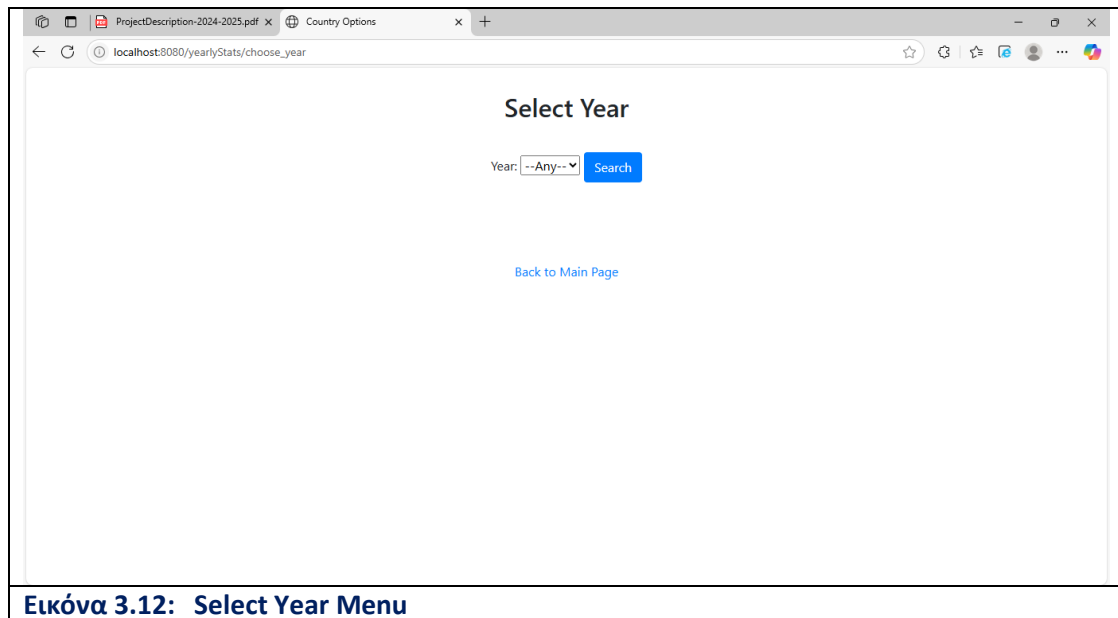
Year: --Any-- Only Home Matches or Away Matches: --Any-- Result: --Any-- Start Date: ηη/χλ/εεεε End Date: ηη/χλ/εεεε

**Apply Filter**

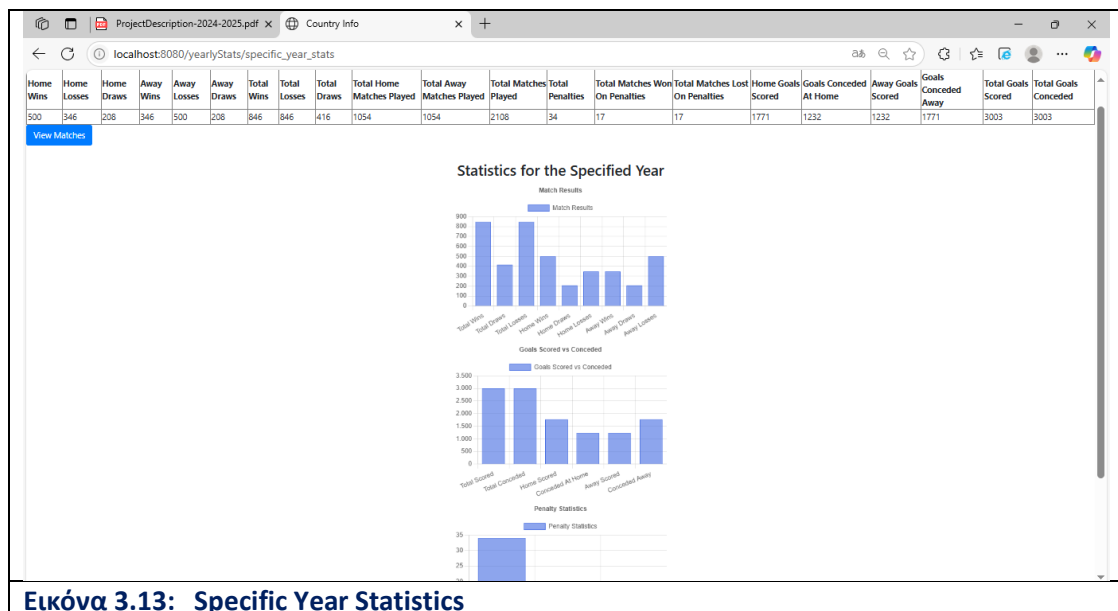
Date	Home	Away	Score	City	Location	Tournament
1919-06-25	Italy	Greece	9 - 0	Paris	France	Inter-Allied Games
1919-06-26	France	Greece	11 - 0	Paris	France	Inter-Allied Games
1919-06-28	Greece	Romania	3 - 2	Paris	France	Inter-Allied Games
1929-06-30	Bulgaria	Greece	1 - 1	Sofia	Bulgaria	Friendly
1930-01-26	Greece	Yugoslavia	2 - 1	Athens	Greece	Balkan Cup
1930-05-25	Romania	Greece	8 - 1	Bucharest	Romania	Balkan Cup
1930-12-07	Greece	Bulgaria	6 - 1	Athens	Greece	Balkan Cup
1931-03-15	Yugoslavia	Greece	4 - 1	Belgrade	Yugoslavia	Balkan Cup

Εικόνα 3.11: Specific Country Matches

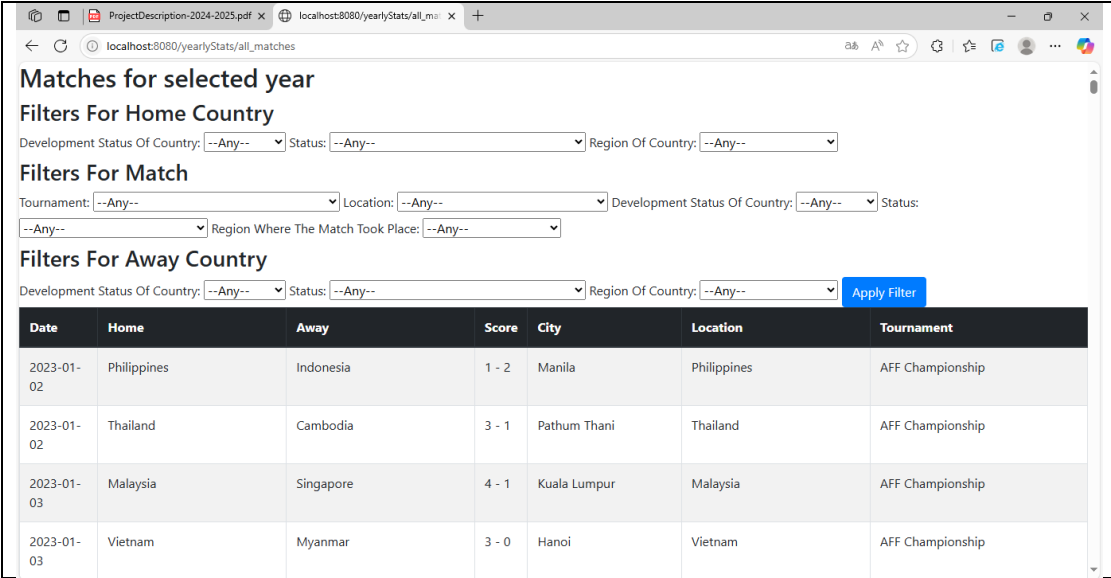
Από το αρχικό μενού πάλι, αν επιλέξει το Years θα βρεθεί στο παρακάτω μενού όπου μπορεί να επιλέξει μία συγκεκριμένη χρονιά για να δει τα στατιστικά.

**Εικόνα 3.12: Select Year Menu**

Επιλέγοντας μία χρονιά και αναζητώντας την, θα βρεθεί στην παρακάτω σελίδα όπου μπορεί να μελετήσει τα παρακάτω στατιστικά.

**Εικόνα 3.13: Specific Year Statistics**

Φυσικά, πατώντας το κουμπί View Matches, μπορεί να δει όλους του αγώνες εκείνης της χρονιάς και να εφαρμόσει διάφορα φίλτρα.



**Matches for selected year**

**Filters For Home Country**

Development Status Of Country: --Any-- Status: --Any-- Region Of Country: --Any--

**Filters For Match**

Tournament: --Any-- Location: --Any-- Development Status Of Country: --Any-- Status: --Any-- Region Where The Match Took Place: --Any--

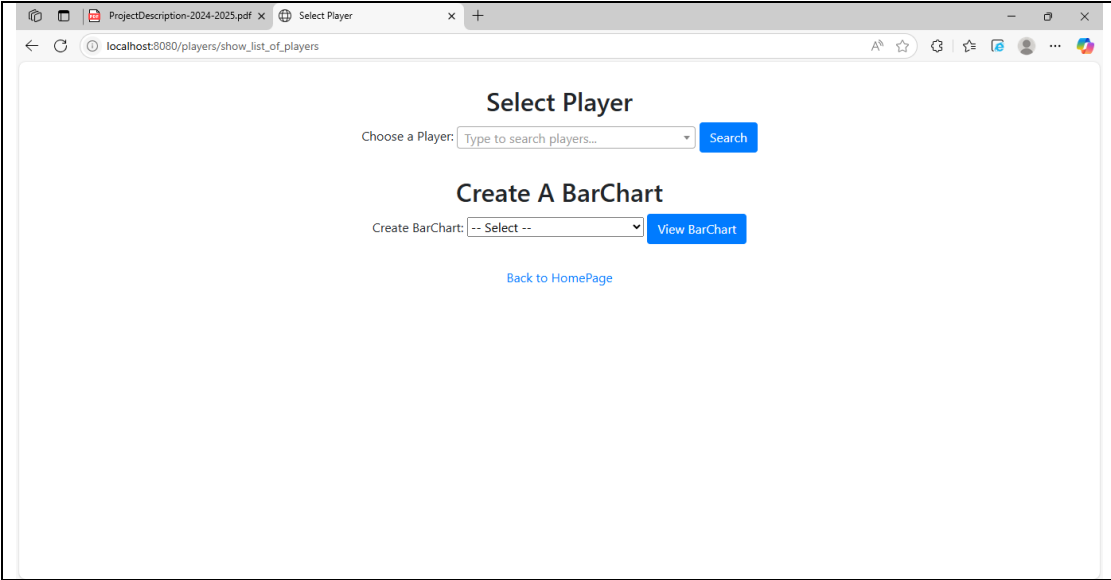
**Filters For Away Country**

Development Status Of Country: --Any-- Status: --Any-- Region Of Country: --Any-- [Apply Filter](#)

Date	Home	Away	Score	City	Location	Tournament
2023-01-02	Philippines	Indonesia	1 - 2	Manila	Philippines	AFF Championship
2023-01-02	Thailand	Cambodia	3 - 1	Pathum Thani	Thailand	AFF Championship
2023-01-03	Malaysia	Singapore	4 - 1	Kuala Lumpur	Malaysia	AFF Championship
2023-01-03	Vietnam	Myanmar	3 - 0	Hanoi	Vietnam	AFF Championship

**Εικόνα 3.14: Specific Year All Matches**

Αν από το αρχικό μενού πατήσει το κουμπί **Players**, τότε θα βρεθεί στην παρακάτω σελίδα.



**Select Player**

Choose a Player:  [Search](#)

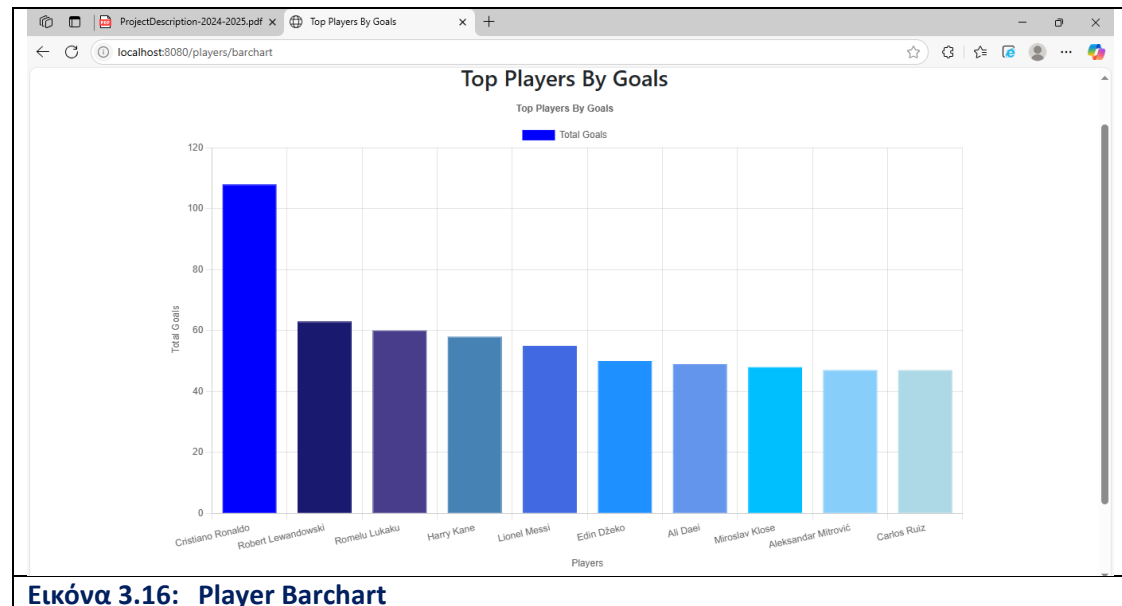
**Create A BarChart**

Create BarChart: --Select-- [View BarChart](#)

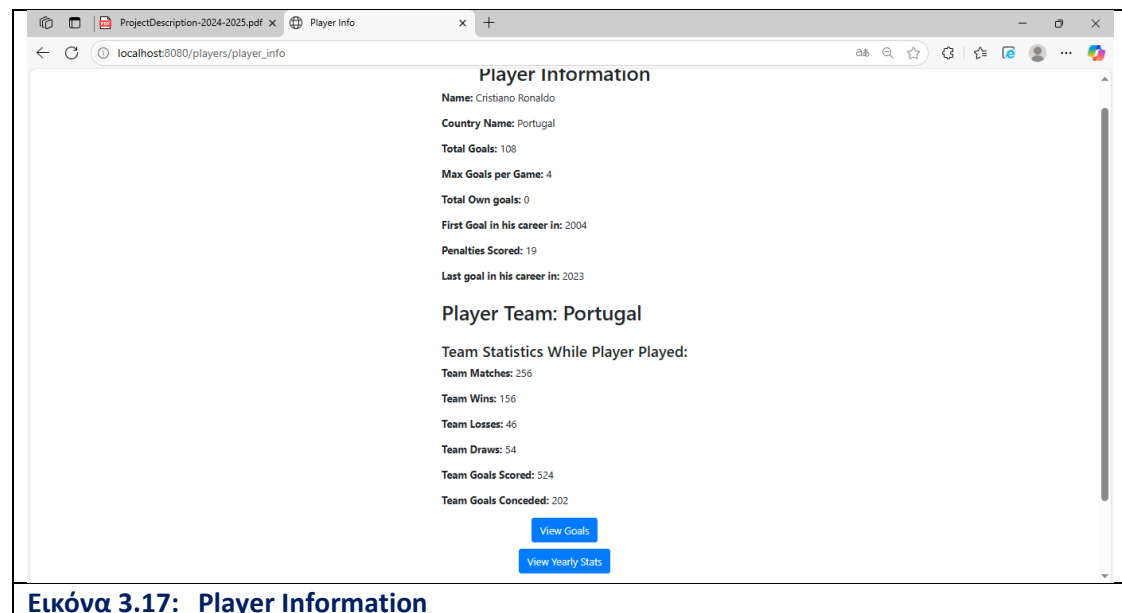
[Back to HomePage](#)

**Εικόνα 3.15: Player Menu**

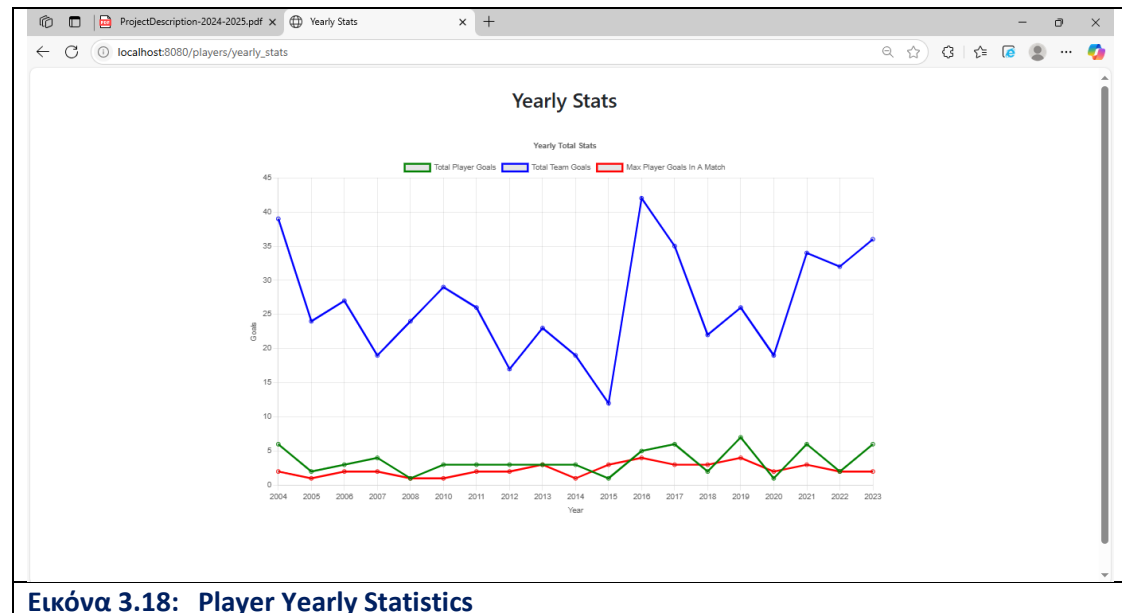
Μπορεί έπειτα να δει κάποιο από τα διαθέσιμα barchart.

**Εικόνα 3.16: Player Barchart**

Μπορεί να αναζητήσει κάποιον συγκεκριμένο παίκτη και να δει τα ακόλουθα στατιστικά του.

**Εικόνα 3.17: Player Information**

Αν από εκεί πατήσει το κουμπί View Yearly Stats, τότε θα δει τα παρακάτω στατιστικά για τον παίκτη.



Εικόνα 3.18: Player Yearly Statistics

Τέλος αν πατήσει το κουμπί View Goals, θα πάει στην παρακάτω σελίδα όπου έχει τα ματς όπου σκόραρε και μπορεί να εφαρμόσει και εδώ διάφορα φίλτρα.

Player Goals List

Filters For Matches

OwnGoal: --Any-- Penalty Goal: --Any-- Tournament: --Any-- Apply Filter

Date	Home	Away	Score	Tournament	Minute Of Goal	OwnGoal	Penalty Goal
2004-06-12	Portugal	Greece	1 - 2	UEFA Euro	90	false	false
2004-06-30	Portugal	Netherlands	2 - 1	UEFA Euro	26	false	false
2004-09-04	Latvia	Portugal	0 - 2	FIFA World Cup qualification	57	false	false
2004-09-08	Portugal	Estonia	4 - 0	FIFA World Cup qualification	75	false	false
2004-10-13	Portugal	Russia	7 - 1	FIFA World Cup qualification	39	false	false
2004-10-13	Portugal	Russia	7 - 1	FIFA World Cup qualification	69	false	false
2004-11-17	Luxembourg	Portugal	0 - 5	FIFA World Cup qualification	28	false	false
2005-06-04	Portugal	Slovakia	2 - 0	FIFA World Cup qualification	42	false	false
2005-06-08	Estonia	Portugal	0 - 1	FIFA World Cup qualification	32	false	false
2006-06-17	Portugal	Iran	2 - 0	FIFA World Cup	80	false	true
2006-10-07	Portugal	Azerbaijan	3 - 0	UEFA Euro qualification	25	false	false
2006-10-07	Portugal	Azerbaijan	3 - 0	UEFA Euro qualification	63	false	false

Εικόνα 3.19: Player Matches Scored