Soccer Results Portal - Project Report

Istanbul Aydin University
Department of Software Engineering

Course: Software Architecture
Instructor: Prof. Dr. METİN ZONTUL
Semester/Year: Spring 2025

Project Title: Soccer Results Portal

Group: 46

Group Members

Emir Beşir, B2105.090065, emirbesir@stu.aydin.edu.tr

Kerem Taşpınar, B2105.090084, keremtaspinar@stu.aydin.edu.tr

Google Drive Link

Table of Contents

1. Introduction

1.1 Problem Definition

In the modern digital era, soccer enthusiasts and team managers require an efficient platform to track match results, upcoming fixtures, and team information. Traditional methods of sharing soccer results through scattered sources lack centralization and real-time updates. There was a need for a comprehensive web-based solution that could provide:

- Centralized access to match results and fixtures
- Team and player management capabilities
- Administrative control for data management
- Responsive design for various devices
- Secure user authentication and authorization

1.2 Objective of the Project

The Soccer Results Portal aims to provide a comprehensive web-based platform for managing and displaying soccer-related information. The primary objectives include:

1. **User-Friendly Interface**: Develop an intuitive and responsive web portal that allows users to easily navigate through match results, fixtures, and team information.

2. **Administrative Functionality**: Implement secure administrative features enabling authorized users to perform CRUD operations on teams, players, matches, fixtures, and venues.

3. **Scalable Architecture**: Utilize ASP.NET Core MVC pattern to ensure maintainability, scalability, and separation of concerns.

4. **Data Integrity**: Design a normalized relational database following 1NF, 2NF, and 3NF principles to eliminate redundancy and maintain consistency.

5. **Security Implementation**: Integrate ASP.NET Core Identity for user authentication and role-based authorization to protect sensitive data.

6. **Responsive Design**: Employ Bootstrap framework to ensure the application works seamlessly across different devices and screen sizes.

2. Technologies Used in the Project

2.1 Frontend Technologies
- **HTML5**: Markup language for structuring web content
- **CSS3**: Styling language for designing the visual presentation
- **JavaScript**: Client-side scripting for interactive functionality
- **Bootstrap 5**: Frontend framework for responsive design and UI components
- **Razor Pages**: Server-side rendering technology for dynamic web pages

2.2 Backend Technologies
- **ASP.NET Core 8.0**: Cross-platform, high-performance framework for building web applications
- **C#**: Primary programming language for server-side logic
- **Entity Framework Core**: Object-Relational Mapping (ORM) framework for database operations
- **ASP.NET Core Identity**: Authentication and authorization framework

2.3 Database Technology
- **Microsoft SQL Server**: Relational database management system for data storage
- **Entity Framework Code-First**: Database schema generation from C# model classes
- **SQL Server Management Studio (SSMS)**: Database administration and management tool

2.4 Development Tools
- **JetBrains Rider**: Integrated Development Environment (IDE)
- **Git & GitHub**: Version control system for source code management
- **NuGet Package Manager**: Package management for .NET libraries

3. Software Architecture

3.1 Frontend Architecture

The frontend follows a component-based architecture using the MVC pattern's View layer:

View Layer Components:

- **Layout Templates**: Shared layout files (_Layout.cshtml) providing consistent navigation and styling

- **Partial Views**: Reusable UI components for common functionality - **Razor Views**: Server-side rendered pages with embedded C# code

- **Bootstrap Components**: Responsive UI elements including navigation bars, cards, tables, and forms
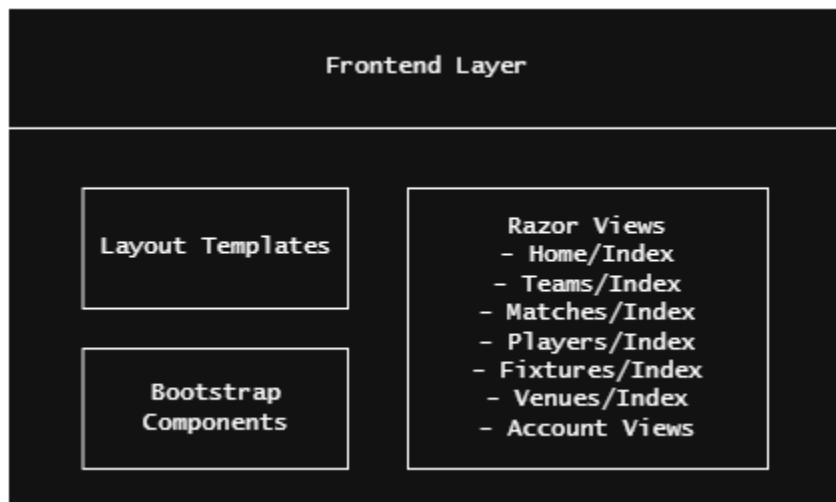
Fig. 1: Frontend Architecture Diagram

3.2 Backend Architecture

The backend implements the MVC (Model-View-Controller) architectural pattern:

Controller Layer:

- **HomeController**: Handles homepage and dashboard functionality - **TeamsController**: Manages CRUD operations for soccer teams

- **PlayersController**: Handles player management and team associations - **MatchesController**: Manages match records and results

- **FixturesController**: Handles upcoming match fixtures

- **VenuesController**: Manages venue information

- **AccountController**: Handles user authentication and authorization

Model Layer:

- **Entity Models**: Represent database tables (Team, Player, Match, Fixture, Venue)

- **View Models**: Data transfer objects for view rendering (HomeViewModel, AuthViewModels)

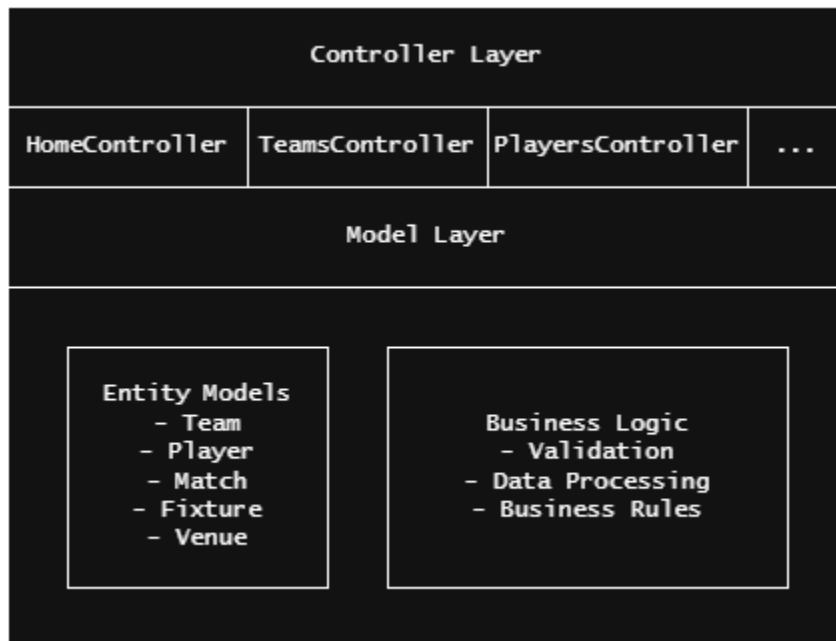- **DbContext**: Entity Framework context for database operations

Fig. 2: Backend MVC Architecture

3.3 Database Architecture

The database follows a normalized relational structure with the following tables and relationships:

Table 1: Database Tables Structure

Table Name	Primary Key	Purpose
Teams	TeamID	Store soccer team information
Players	PlayerID	Store player details and team associations
Venues	VenueID	Store match venue information
Matches	MatchID	Store match records and results
Fixtures	FixtureID	Store upcoming match schedules
AspNetUsers	Id	Store user authentication data
AspNetRoles	Id	Store user role definitions

Entity Relationships:

- **One-to-Many**: Team → Players (One team has many players)

- **Many-to-One**: Match → Team1, Team2 (Many matches reference teams)

- **Many-to-One**: Match → Venue (Many matches at one venue)

- **One-to-Many**: Match → Fixtures (One match can have multiple fixture entries)
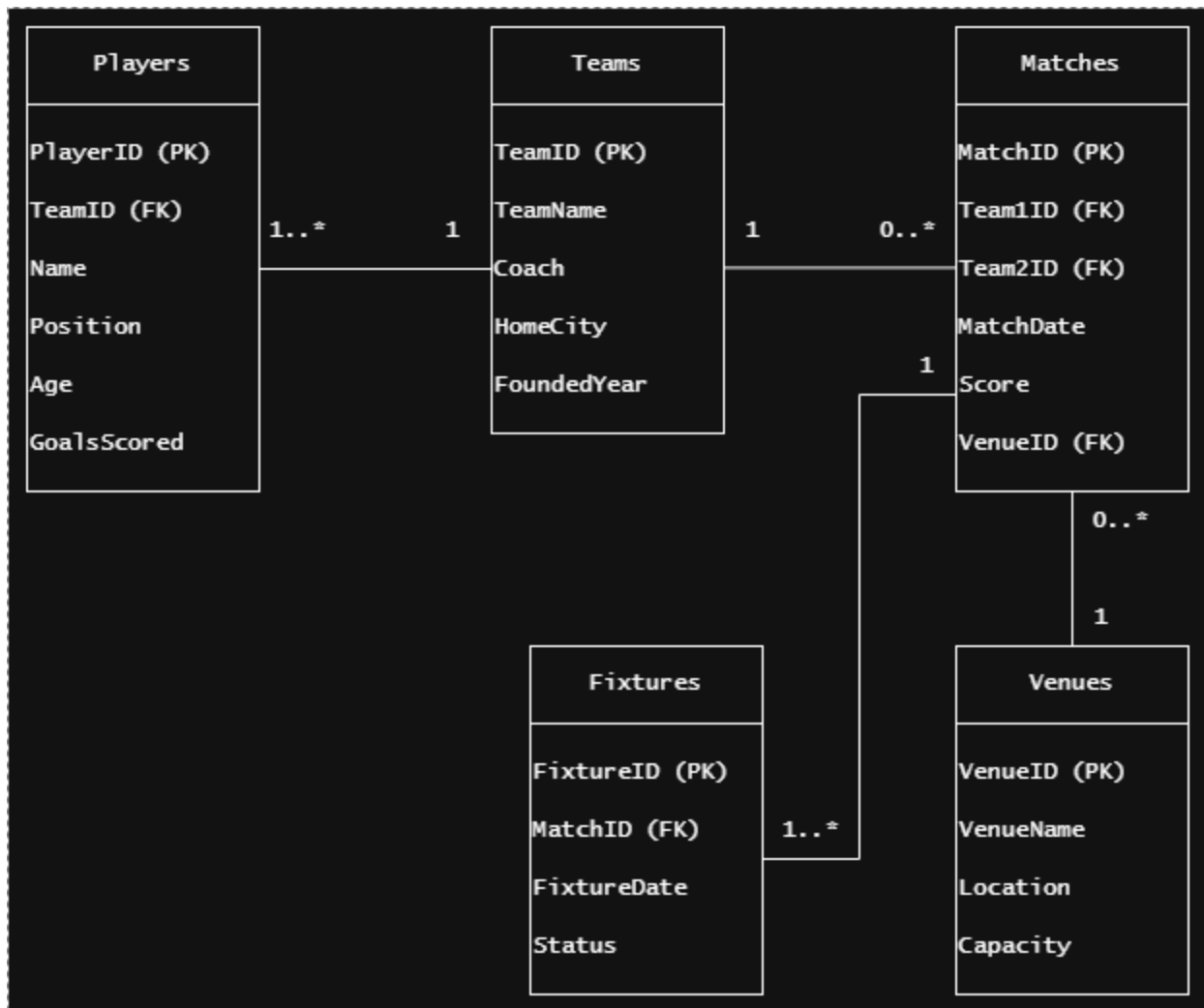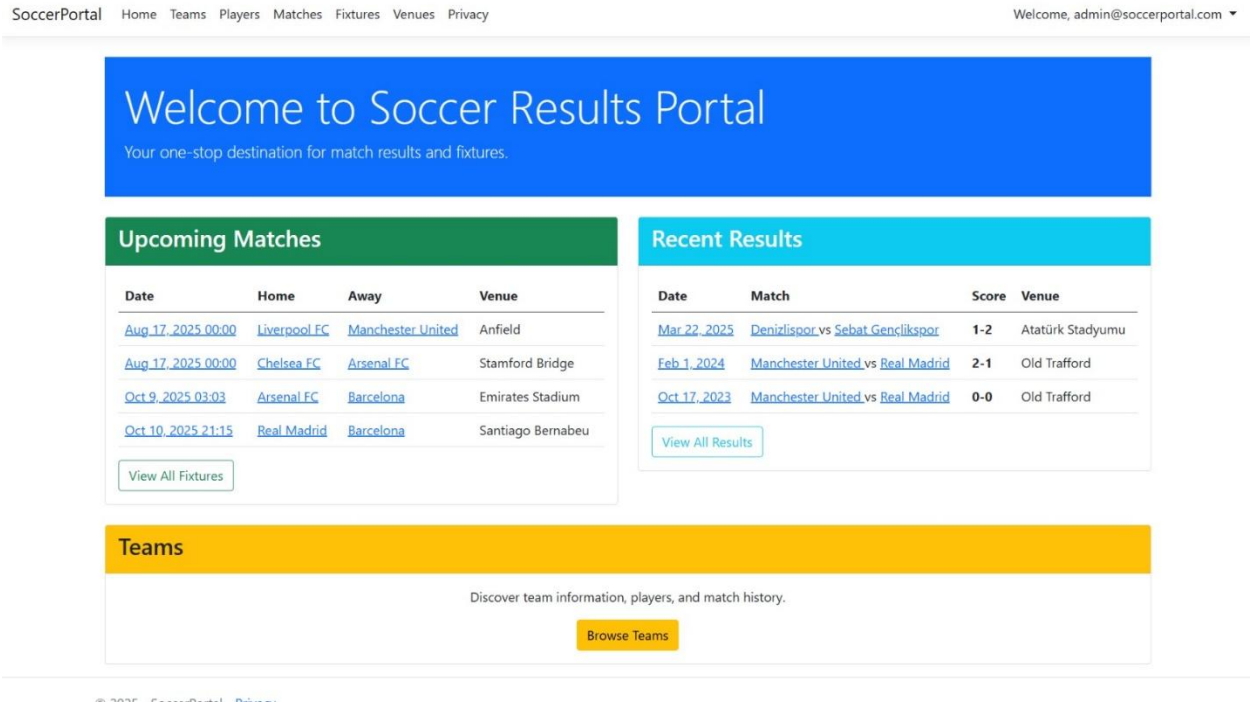
Fig. 3: Entity Relationship Diagram

4. Screenshots and Explanations

4.1 Home Page Dashboard

Fig. 4: Soccer Portal Homepage

[Screenshot Description: The homepage displays a clean, modern interface with a welcome banner, navigation menu, and three main sections showing upcoming matches, recent results, and teams. The Bootstrap-styled layout is responsive and user-friendly.]

Key Features:

- Responsive navigation bar with links to all major sections

- Dynamic display of upcoming matches with team names, dates, and venues

- Recent match results showing scores and outcomes

- Clean, professional design using Bootstrap components

4.2 Teams Management

Fig. 5: Teams Index Page

[Screenshot Description: A comprehensive grid view displaying all registered teams with columns for team name, coach, home city, and founded year. Action buttons for Create, Edit, Details, and Delete operations are visible.]

Key Features:

- Display of all teams in a grid layout.

- CRUD operation buttons for administrative users

- Responsive design for mobile compatibility

4.3 Players Management

Fig. 6: Players Management Interface

[Screenshot Description: Player listing page showing player details including name, position, age, goals scored, and associated team. Administrative controls are present for managing player records.]

Key Features:

- Detailed player information display

- Team association through foreign key relationships

- Statistical information (goals scored, position)

- Full CRUD operations for player management

4.4 Match Results

Soccer Matches

Add New Match

Filter Matches

Team:

All Teams

Venue:

All Venues

Date:

gg.aa.yyyy

Filter   Clear Filters

Match Schedule & Results

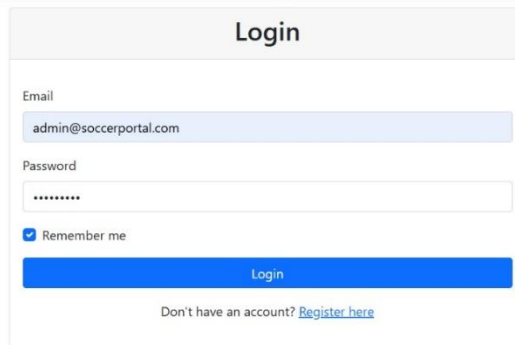| Date	Home Team	Away Team	Score	Venue	Actions
Aug 17, 2025 00:00	Liverpool FC	Manchester United	Upcoming	Anfield	Details Edit Delete
Aug 17, 2025 00:00	Chelsea FC	Arsenal FC	Upcoming	Stamford Bridge	Details Edit Delete
Oct 9, 2025 03:03	Arsenal FC	Barcelona	Upcoming	Emirates Stadium	Details Edit Delete
Oct 10, 2025 21:15	Real Madrid	Barcelona	0-0	Santiago Bernabeu	Details Edit Delete

Fig. 7: Matches Management System

[Screenshot Description: Match management interface displaying past and upcoming matches with team names, dates, scores, and venue information.]

Key Features:

- Comprehensive match history tracking

- Score recording for completed matches

- Venue and date information

- Integration with teams and venues data

4.5 User Authentication

SoccerPortal   Home Teams Players Matches Fixtures Venues Privacy   Login Register

Login

Email
admin@soccerportal.com

Password
••••••••

☑ Remember me

Login

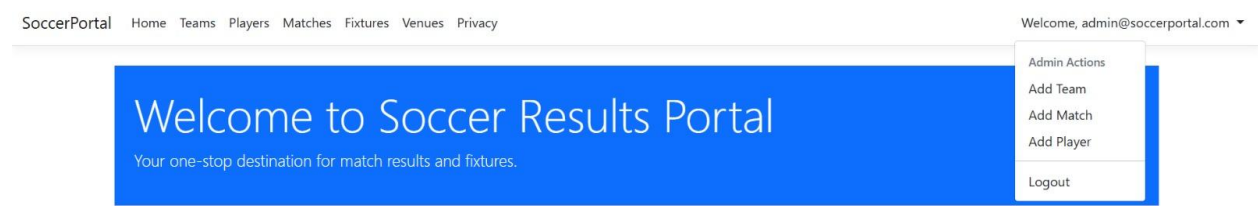Don't have an account? Register here

© 2025 - SoccerPortal - Privacy

Fig. 8: Login System

[Screenshot Description: Clean login form with username/email and password fields, along with registration link. Security features and validation messages are visible.]

Key Features:

- Secure user authentication

- Password strength validation

- Account registration functionality

- Role-based access control

4.6 Administrative Features

SoccerPortal Home Teams Players Matches Fixtures Venues Privacy   Welcome, admin@soccerportal.com ▾

	Admin Actions
	Add Team
	Add Match
	Add Player
	Logout

Welcome to Soccer Results Portal
Your one-stop destination for match results and fixtures.

Fig. 9: Admin Panel Interface

[Screenshot Description: Administrative interface showing enhanced navigation options for users with admin privileges, including access to all CRUD operations and user management features.]

Key Features:

- Role-based navigation enhancement

- Administrative control over all entities

- Secure access to sensitive operations

5. Conclusion

The Soccer Results Portal project has been successfully developed as a comprehensive web-based solution for managing soccer-related information. The implementation demonstrates proficiency in modern web development technologies and software engineering principles.

5.1 Project Achievements

1. **Successful Implementation of MVC Architecture**: The project effectively implements the Model-View-Controller pattern, ensuring clear separation of concerns and maintainable code structure.

2. **Robust Database Design**: The relational database follows normalization principles (1NF, 2NF, 3NF), ensuring data integrity and eliminating redundancy.

3. **Security Integration**: ASP.NET Core Identity provides comprehensive authentication and authorization, protecting sensitive operations and data.

4. **Responsive User Interface**: Bootstrap framework ensures the application works seamlessly across various devices and screen sizes.

5. **Complete CRUD Functionality**: All major entities (Teams, Players, Matches, Fixtures, Venues) support full Create, Read, Update, and Delete operations.

6. **Code-First Database Approach**: Entity Framework Core migrations enable dynamic database schema management and version control.

6. References

1. Microsoft Corporation. (2024). *ASP.NET Core Documentation*. Retrieved from https://docs.microsoft.com/en-us/aspnet/core/

2. Microsoft Corporation. (2024). *Entity Framework Core Documentation*. Retrieved from https://docs.microsoft.com/en-us/ef/core/

3. Bootstrap Team. (2024). *Bootstrap Documentation*. Retrieved from https://getbootstrap.com/docs/5.3/

4. Microsoft Corporation. (2024). *ASP.NET Core Identity Documentation*. Retrieved from https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity

5. Microsoft Corporation. (2024). *SQL Server Documentation*. Retrieved from https://docs.microsoft.com/en-us/sql/sql-server/