UE21CS352B - Object Oriented Analysis & Design using Java

Mini Project Report

"Sports team matching website"

Submitted by:

Nachiketha Manoor	**PES1UG21CS354**
Miha Parveez	**PES1UG21CS336**
Meghana Goru	**PES1UG21CS333**
Manas Rahul Shah	**PES1UG21CS324**

6th Semester F Section

Mrs Bhargavi Mokashi
Assistant Professor

January - May 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

Table of contents

1.1. Introduction

The popularity of sports and recreational activities continues to grow, with millions of people seeking to engage in games and competitions, either as individuals or as part of a team. With this increasing interest, there arises a need for a platform that can effectively match players and teams based on skill levels, interests, and geographic proximity. The proposed project involves the development of a sports matching site, designed using Java and incorporating key software design principles and the Model-View-Controller (MVC) architectural pattern. This platform will facilitate user logins, profile management , elo scores, and the ability to match with other players or teams who share similar sports preferences and objectives.

1.2. Problem statement (synopsis)

Many sports enthusiasts face challenges in finding suitable teammates or opponents who match their skill level and interests within their locality. Existing solutions often lack personalized matching algorithms and do not cater to the specific needs and preferences of individual players and teams, leading to mismatches and a suboptimal playing experience. There is also a significant gap in platforms that offer seamless integration of scheduling and communication tools that are essential for coordinating matches and related activities.

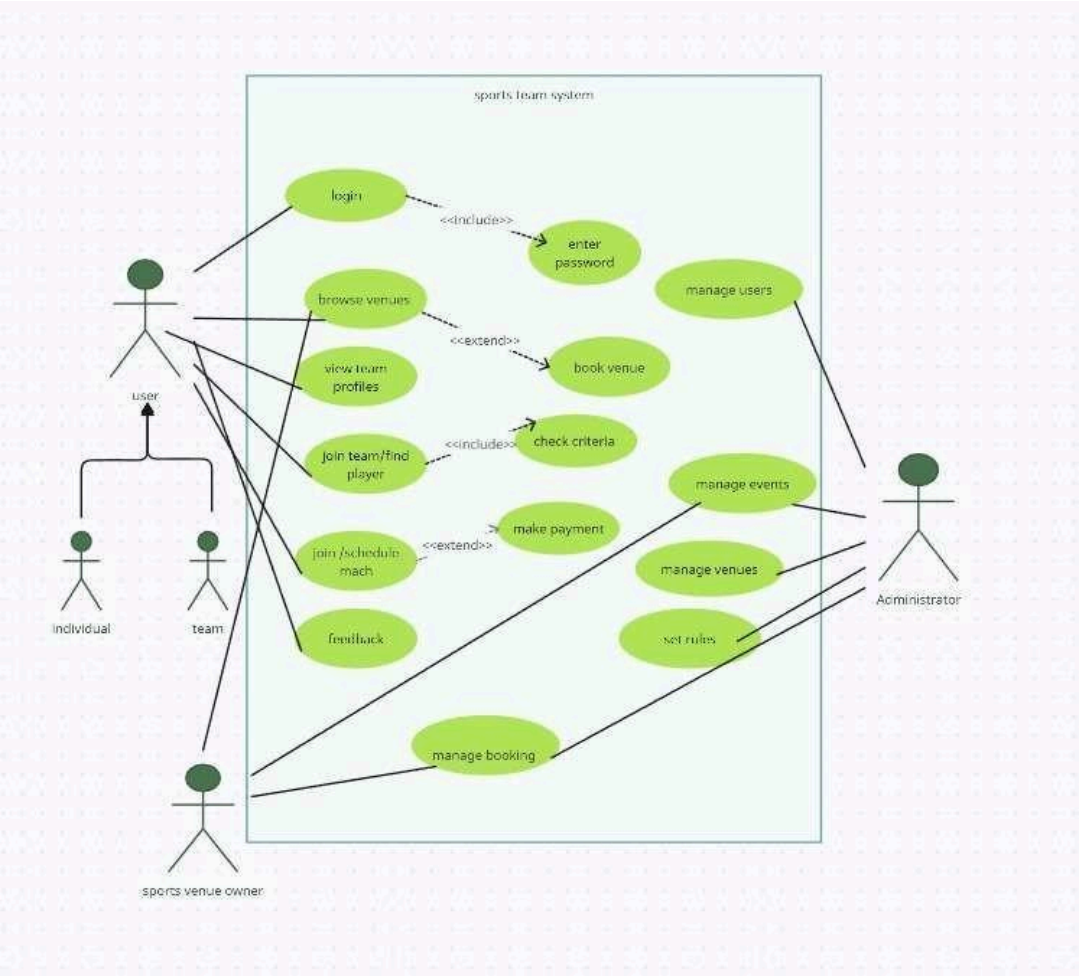
1.3. Scope

Development Environment: Utilize Java and modern development tools and libraries suited for building scalable and maintainable web applications.
MVC Architecture: Apply the Model-View-Controller pattern to separate the application's concerns, enhancing the maintainability and scalability of the codebase.
User Management: Functionality for user registration and login.
Matching System: Development of a core algorithm for matching users based on their inputs and preferences and updating user’s elo rating.

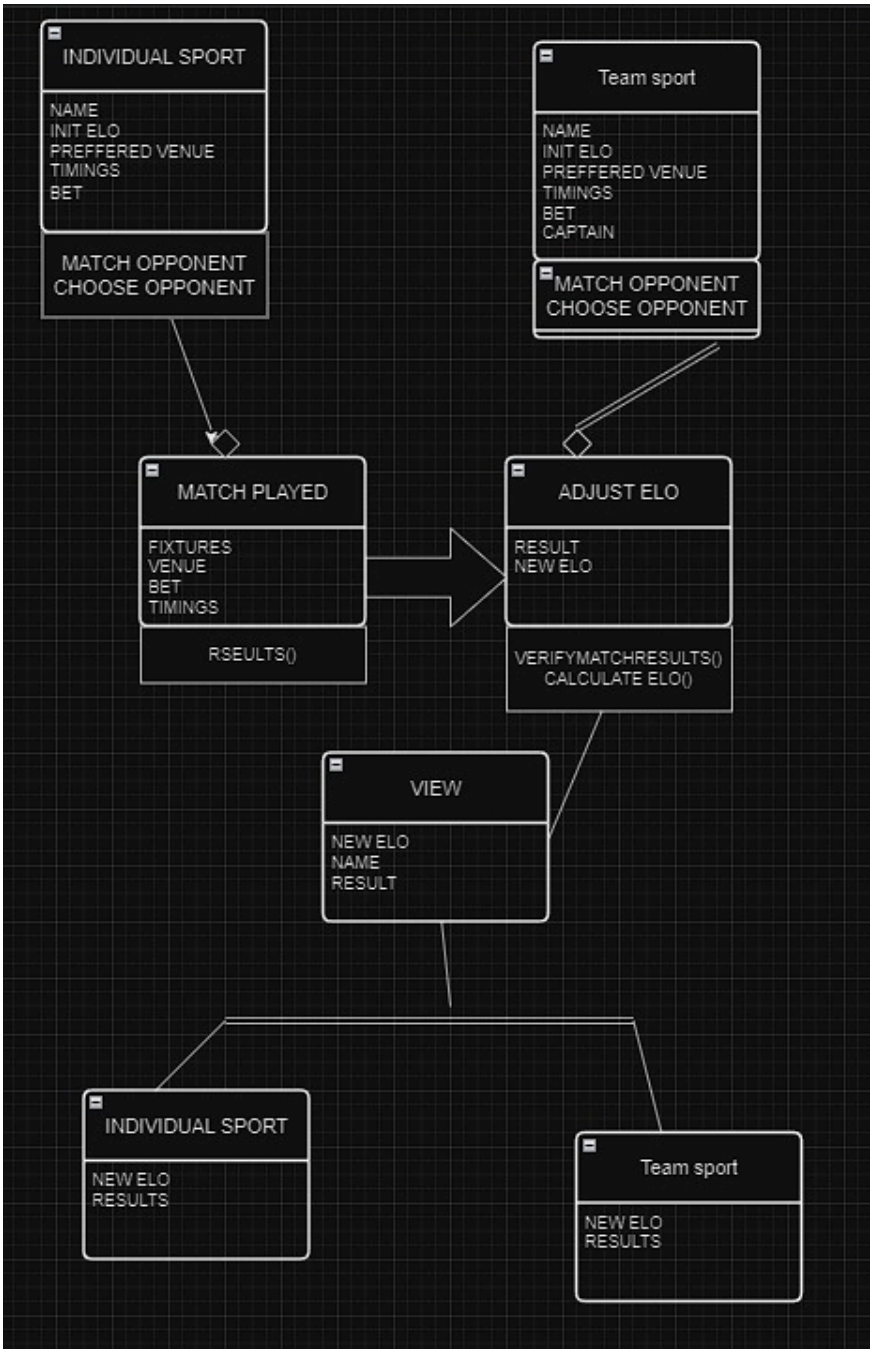
Data Management: Utilization of a suitable backend database to store user data, preferences, match histories, and other relevant information.

2. Models (Use Case, Class Models, State & Activity)

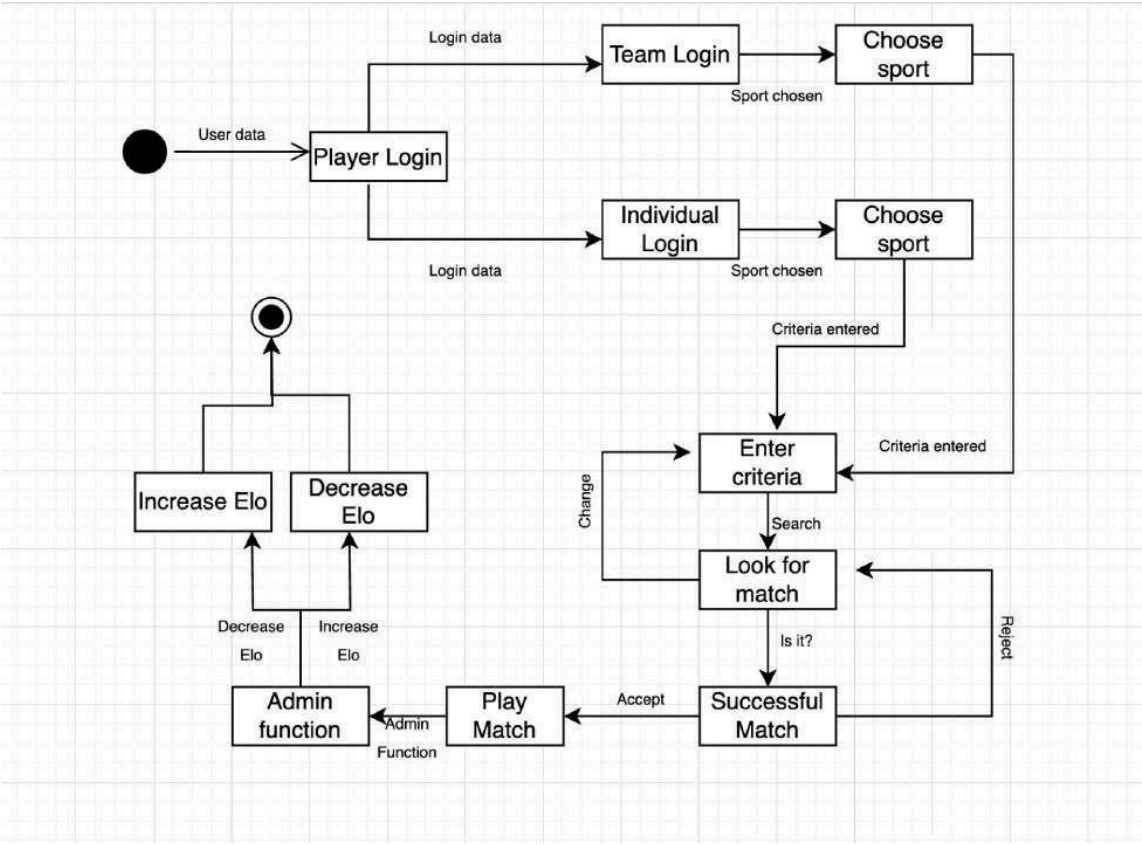
USE CASE:



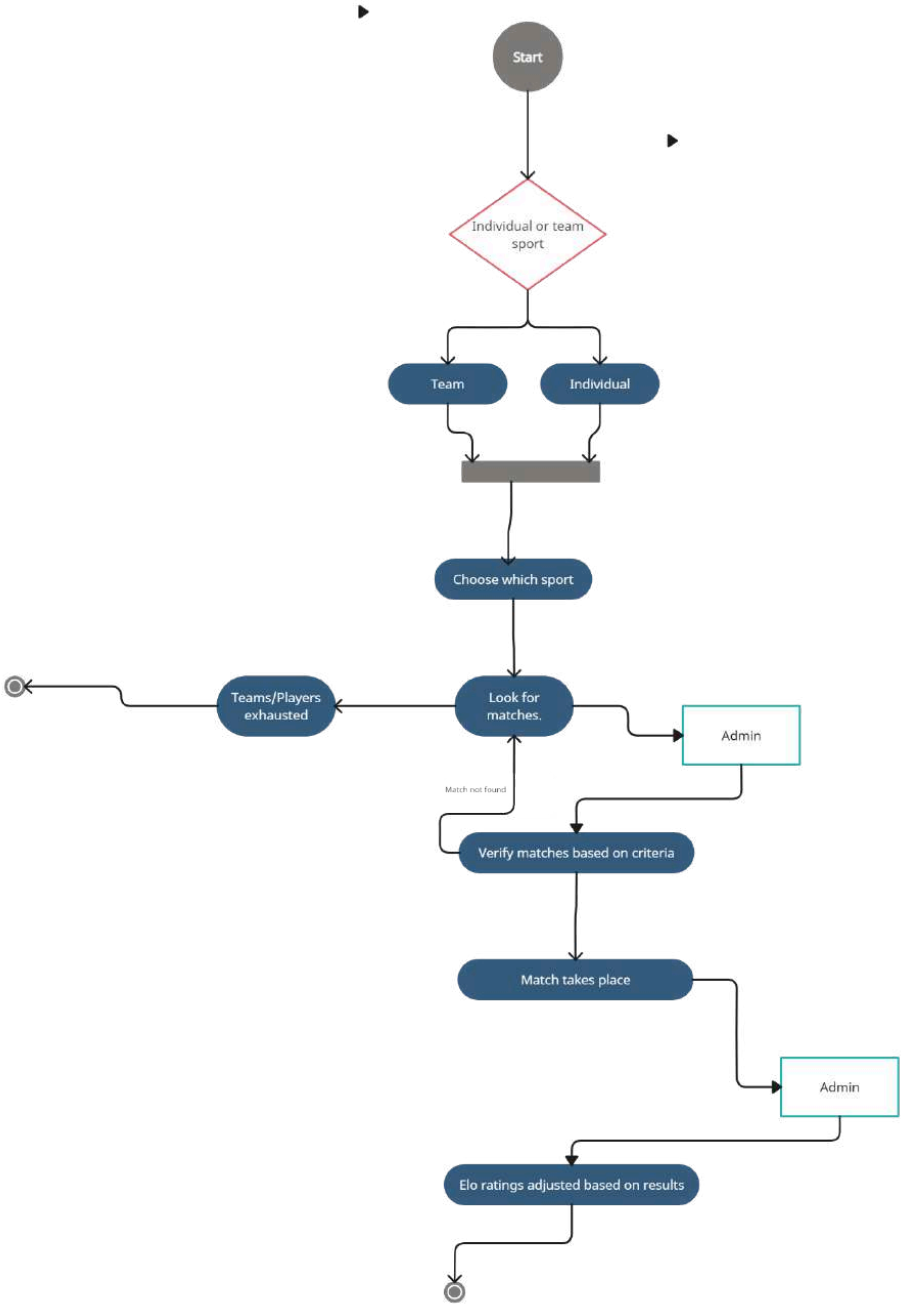
CLASS DIAGRAM:



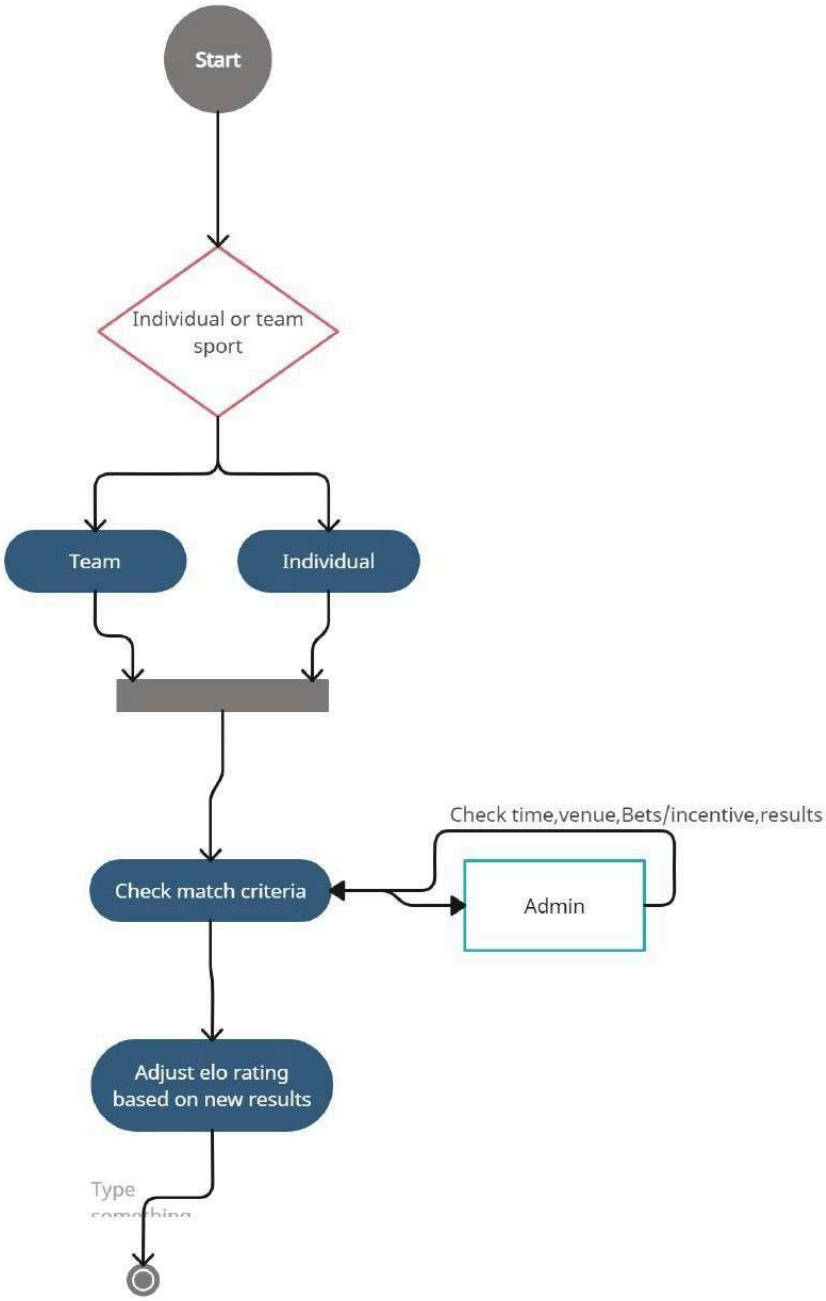
STATE DIAGRAM:



ACTIVITY DIAGRAM:



MINOR ACTIVITY DIAGRAM:



3. Architectural patterns used:

Model-View-Controller (MVC)
Although more of an architectural pattern, MVC is crucial here. The AppController acts as the Controller in this pattern, handling the business logic and then delegating the display to the View. User and UserBet likely act as the Model in this setup, encapsulating the application data.

Single Responsibility Principle (SRP)
The AppController handles all the HTTP requests related to user operations such as registration, login, and listing users. While it might seem that the controller does a lot, it adheres to SRP in the context of handling all actions related to a specific domain (users and their interactions).

Dependency Inversion Principle (DIP)
This principle is demonstrated through the use of Spring's Dependency Injection. The controller is configured to depend on abstractions (UserRepository and UserBetRepository) rather than concrete implementations. Spring manages these dependencies and injects them at runtime.

Builder Pattern(in match result): Allows for step-by-step creation of complex objects, facilitating customization by specifying different options and variations during construction.
This pattern is applied where we initialise the backend variables/objects in setters/getters class files.

Singleton Pattern(during login):Ensures that a class has only one instance and provides a global point of access to that instance, commonly used for managing resources or settings. This pattern is applied to have one instance while logging in and creating a new instance of a user in the controller section of the Application while we add new user details.

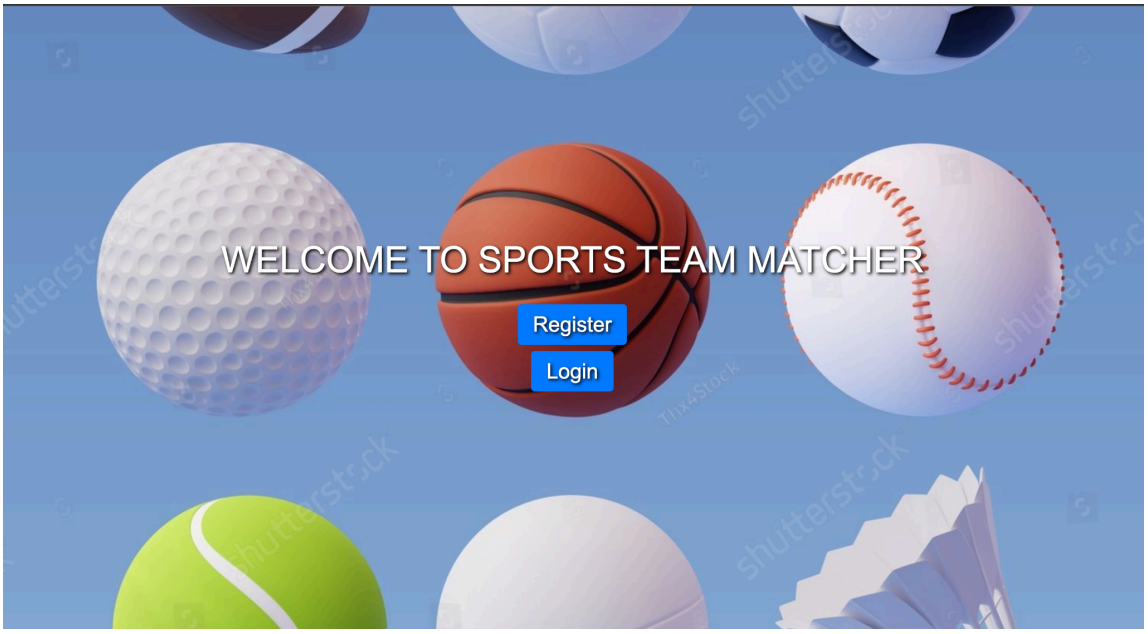
4. Github link to the Codebase (Repository should be public and accessible to all)

https://github.com/meghanagoru/SportsMatchingSystem

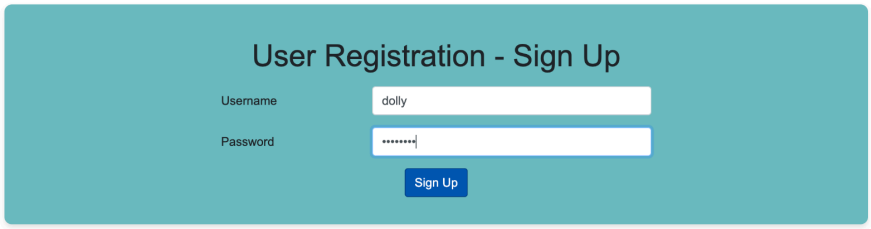
5. Individual contributions of the team members.

MEGHANA GORU	Individual sport portion of the application, frontend and backend including APIs and database connection.
MANAS SHAH	Design of classes and flow. Elo algorithm.
MIHA PARVEEZ	“Team player” portion of the application. Front end of the whole website and design patterns.
NACHIKETHA MANOOR	Integration of APIS with the MYSql Backend; configuring patterns and system properties w.r.t each system.

6. Screenshots with input values populated and output shown



New user can be registered:



You have signed up successfully!

Your ID is: 13

Click here to Login

Existing user can log in:

Login

Username:

meg

Password:

•••••••

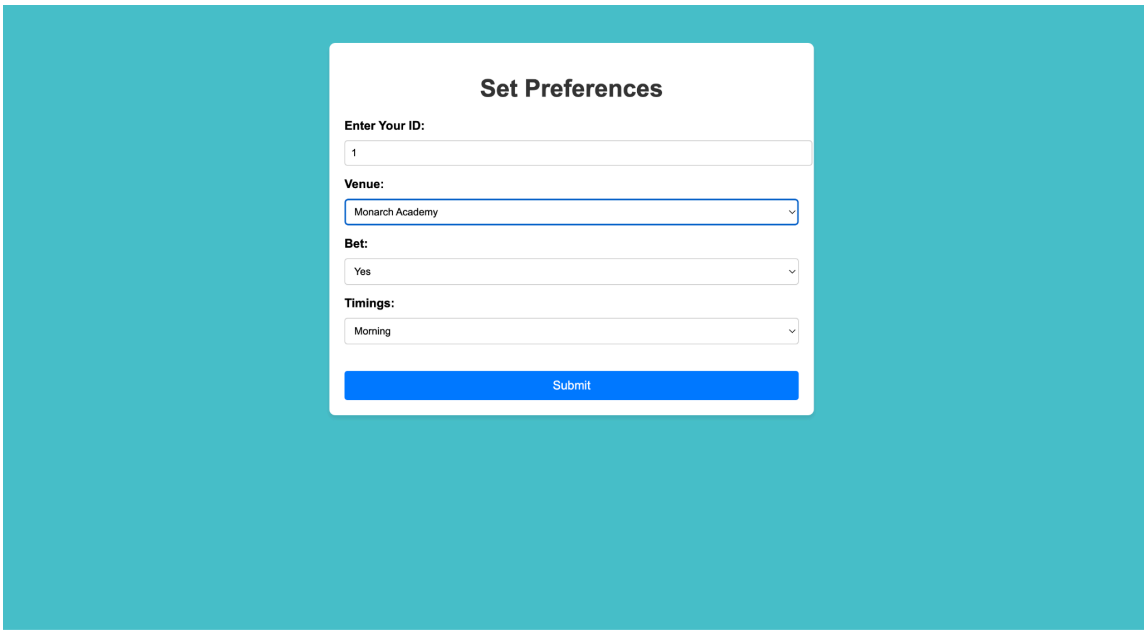
Login

Choose Sport

Individual Sports

Team Sports

"Competition brings out the best in people. It's the little voice in your head that says, 'Whatever you do, do it better than them.'



Top Players (Venue: Monarch Academy, Timings: Morning)

User ID	Timing	Venue	Action
4	Morning	Monarch Academy	Play
6	Morning	Monarch Academy	Play

Match Details

Your Details

User ID: **1**

Name: **meg**

Opponent Details

User ID: **6**

Name: **nachi**

Enter Set Scores

Your Score: 5

Opponent Score: 2

Submit Scores

Updated Elo Ratings

New Elo rating for Player 1: 1184

New Elo rating for Player 2: 1216

Welcome to Sports Team Matcher

Pick a Sport:

Football

Pick the number of people in the team:

11

Match

Match Result

Team 1: **Team A**

Team 2: **Team B**

Date: **2024-04-15**

Venue: **Stadium XYZ**

Let's Play

Enter Scores

Team 1 Score: 3
Team 2 Score: 2
Submit Scores