KLE Society's
KLE TECHNOLOGICAL UNIVERSITY

Open Ended Experiment Report

On

DreamTeam

Team Details:

Name	USN	Roll No
Prasad V Patil	01FE18BCS148	329
Prasann Kshirasagar	01FE18BCS149	330
Praveen Naik	01FE18BCS153	334
Praveen Savadkar	01FE18BCS154	335

CONTENTS

I. INTRODUCTION

Sports have a sky-high fan following around the world. Every sport has its separate fan base and fans are going crazy behind it. With the ages, the fans have found different ways to show their love and engagement in their chosen sport. By wearing customized sports outfits, by clicking selfies with the cutouts of their favorite sports star, by inking tattoos, by creating unique hashtags, and what not have showcased the craziness to the sports industry. Every fan loves his favorite sport and has discrete goals for it. In recent times, we heard a new way of engaging in sports through apps. It has a simple agenda, “Get paid for the right guess!”

Yes, it is the most electrifying app among youngsters nowadays is Fantasy Sports App.

People from all over the world take part in the online leagues conducted by the app and earn the right prediction. Between the years 2014-2017, 20% population from the USA and 18% population from Canada have involved themselves in playing fantasy sports. The users are showing high interest and soon the user graphs will rise to 50+ for sure. On average, the fantasy sports app player earns $162 from Daily fantasy and $257 from the traditional league.

Dream11 is one of the most popular and trending fantasy apps in the Indian subcontinent. It has emerged to become one of the biggest names in the fantasy sports genre of mobile applications. With more than 20 million active users, the Dream11 fantasy clone app, as a result, continues to lead the fantasy app crusade. This application is popular because it allows the participants to win some quick money.

Since it is allowing users to choose from a wide range of sports, thus, it is significantly more admired. From football to ice hockey and cricket to basketball, the Dream11 app features games of all these sports tournaments played all over the world. So based on the area of expertise, users will be able to play efficiently. Over time several Dream11 clone apps have emerged as well which testifies how popular it is.

If the participants have the right knowledge and skills, they can win a significant amount of money from each game they play.

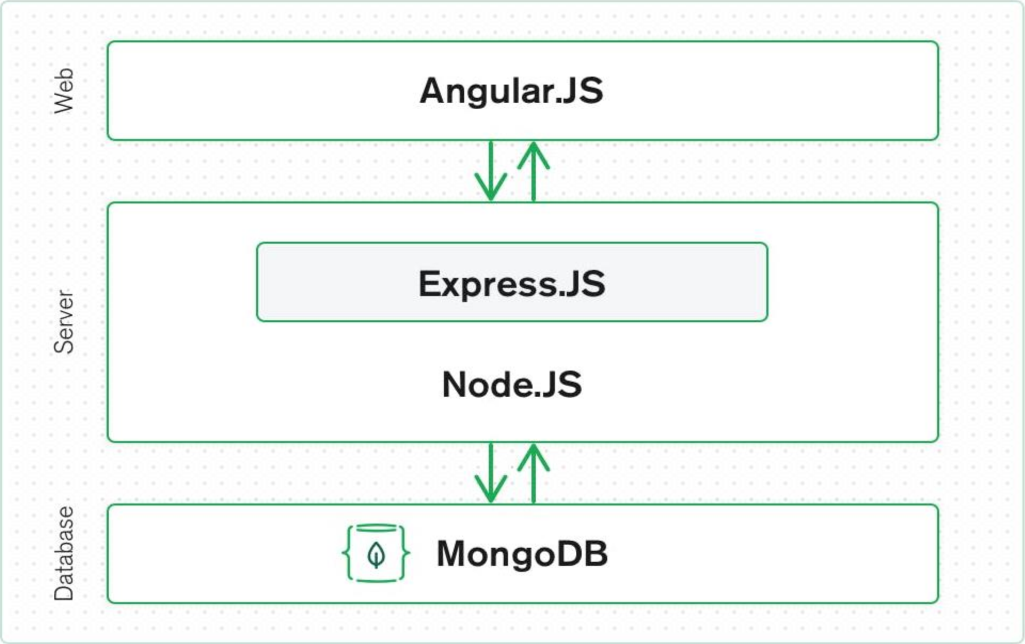
II. PROBLEM STATEMENT:

Design a Fantasy game playing application, in which the user can play a game of his choice. Every game must have a set of rules on building a team. Develop a website that gets users to build his/her Dream Team.

III. APPROACH

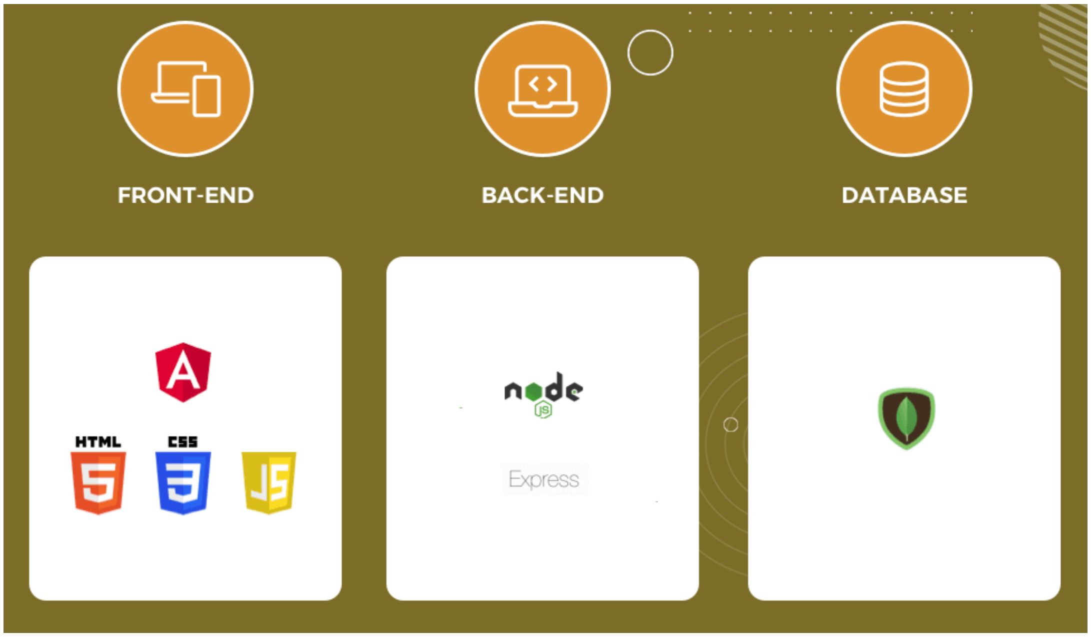
As our project was, to develop a **Fantasy game playing application.** We preferred to go with the MEAN stack framework(JavaScript-based framework). **MEAN** is named after MongoDB, Express, Angular, and Node, the four key technologies that make up the layers of the stack.

- **M**ongoDB - document database
- **E**xpress(.js) - Node.js web framework
- **A**ngular(.js) - a client-side JavaScript framework
- **N**ode(.js) - the premier JavaScript web server



Working with MEAN means going along with three major components on which MEAN stack is divided:

- Front-end or Client-Side Technology Stack
- Back-end or Server-Side Technology Stack
- Databases



Based on the components (front-end, back-end and the database) we went with a component based approach to build our application and later we integrated all components to form a complete web app.

Component based approach:

- **Front-end or Client-Side Technology Stack (Angular) :**
  - HTML template :HTML pages for structuring were created in-order to declare what renders on the page.
  - A Typescript class was created for all the pages to define behavior of the pages.

  - Dynamic URL linking of the pages and back-end information retrieval: All the pages created for the front-end were made as a single page application using Angular's routing feature. Using Angular decisive display of the contents gathered from the backend was possible.

  - CSS with bootstrap styles were applied to the template.

- **Back-end or Server-Side Technology Stack (Node.js and Express) :**

  - On the back-end part all the functions related to database logic were created in Node.js which is a runtime environment for running javascript outside the browser.

  - Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses. By making XML HTTP requests ,or GETs, or POSTs from our Angular.js front end, in-order to connect to MongoDB database Express was used and our Angular.js front-end was connected to Express.js functions.

  - These Express.js functions in turn used MongoDB's Node.js drivers, either via callbacks or using Promises, to access and update data in our created MongoDB database.

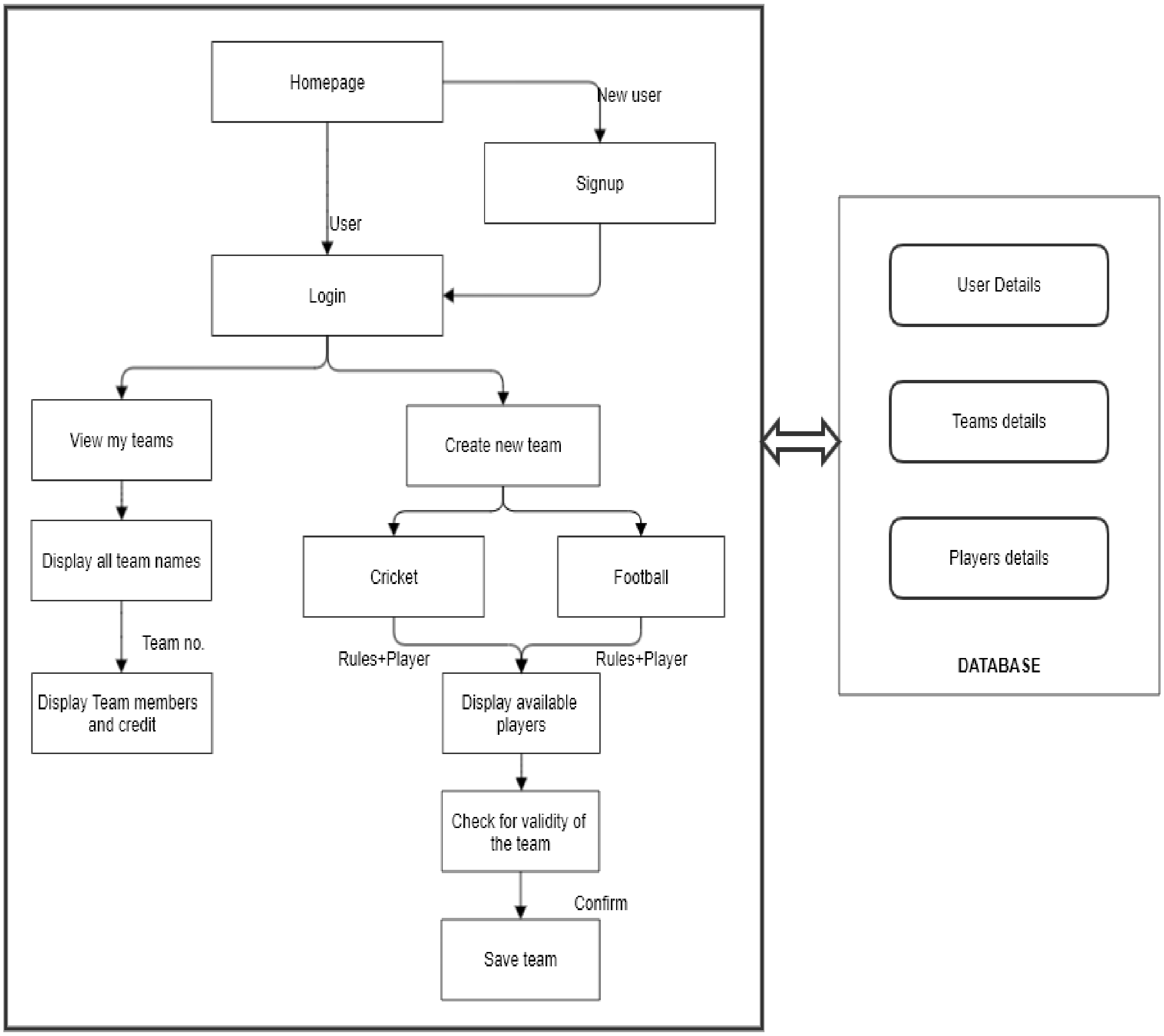
- **Database(MongoDB) :**

  - For storing our application data (user profiles, team details, player details etc.), MongoDB was used which just makes it easy to work with Angular, Express, and Node.

  - JSON documents created in our Angular.js front end were now sent to Express.js server, where they can be processed and (assuming they're valid) and were stored directly in MongoDB for later retrieval.

Couple of team members worked on the front-end part (Creating dynamic HTML pages with CSS bootstrap styling) and other two team members worked on back-end and database modules (coding database logic in Node.js and writing Express.js functions in-order to connect Fantasy game application with MongoDB database) and then all the modules were integrated to complete the project.

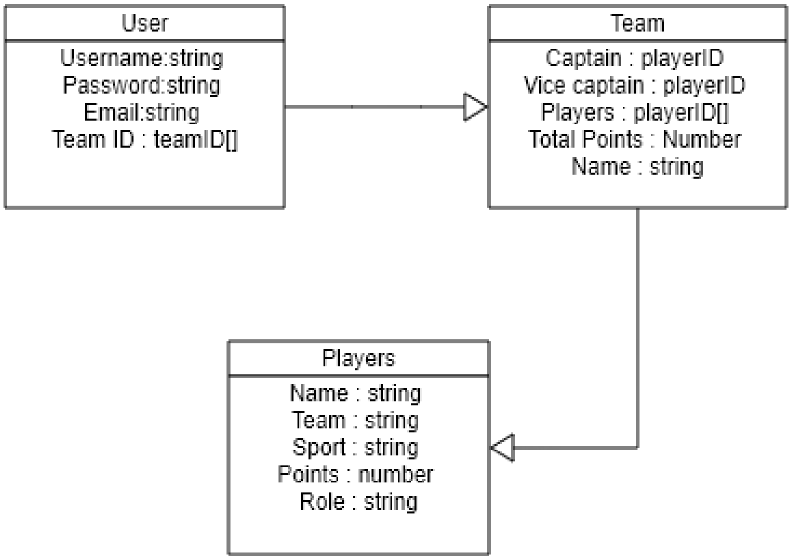
IV. DESIGN



The above figure depicts the flow of the web application. When the user visits the website, the homepage of DreamTeam is displayed. Users will have the login, signup options in the navigation bar. After logging into the system. Users get two options, that is view the already created teams or create a new team.

Under ‘view my teams’, users can see all the teams they created; they can choose any team to see the players of the team and other details. Under ‘create new team’, users can create a team of either football or cricket. Based on users selection, the respective players will be displayed out of which they can select any 11 players based on the constraints of the game. Team will be saved once after validating all the rules.

Database schema:



We have used mongodb as our database. The schema of the database is as shown above.

We store 3 type of entities:

1. **Users**
   By saving user data, the user will be able to login without losing previous data like password id or his teams. The passwords are saved using the **Salt and Hash** which is most secure because of the encryption used.
   Users will have list teams where only team Id is stored so as to minimise data redundancy.

2. **Team**
   As our application says, every user can have his customised team. To store every team we use ‘Team’ entity. Every team will have captain ID, vice captain ID and other details. Team only stores the ID of the players from which we can easily write queries and retrieve data.

3. **Players**
   Each Team must have a set of players, storing these players in a database is the most optimal way. Each Player will have a set of attributes like : Name, Team, Sport, Points, Role. Players will also have their profile pictures which are stored locally inside the assets folder.

V. FEATURES

1. Users can login to the system.
2. New users can create an account.
3. The user doesn't have to login again within 24 hours unless he logged out by himself.
4. Sessions will be continued and session time is restricted for 24 hours.
5. Users can create teams of cricket or football.
6. Users can select players they want until it is valid according to the rules of that particular sport.
7. Users can select captain and vice captain for their teams.
8. Users can view the teams that they have created.
9. Users will also get an option to delete any team.
10. Users can logout from the system after their use.

VI. IMPLEMENTATION DETAILS

All the modules are implemented on the MEAN stack.

Backend implements all the function related database logic and is implemented in Node.js which is a runtime environment for running javascript outside the browser. Express makes it easier to connect with the Mongodb database. These functions are hosted using API (Application Programming Interface) in localhost port 8081.

Frontend does the job of displaying contents to the user in a user friendly way. HTML for structuring, CSS with bootstrap for styling, Angular for the decisive display of contents gathered from the backend. Angular makes the whole application as a single page application with it's routing feature. It makes the life of users easy by making it easier to navigate between the options.

1. User Validation

<u>JWT (JSON Web Token)</u>

Comparing with Session-based Authentication that need to store Session on Cookie, the big advantage of Token-based Authentication is that we store the JSON Web Token (JWT) on Client side: Local Storage for Browser, Keychain for IOS and SharedPreferences for Android… So we don't need to build another backend project that supports Native Apps or an additional Authentication module for Native App users.

Backend overview:

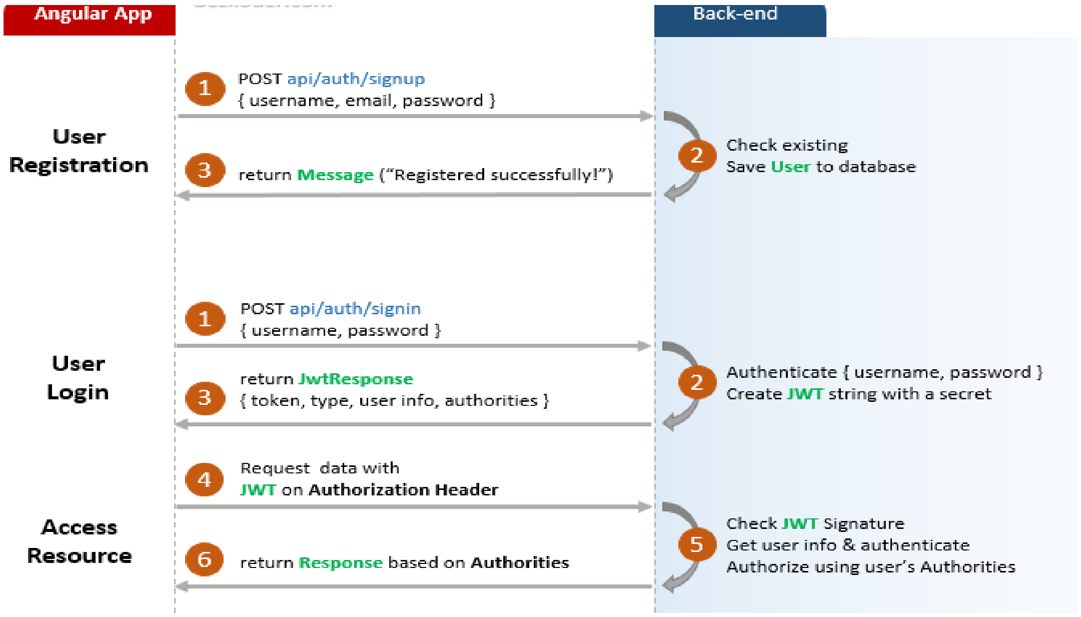
Via Express routes, HTTP requests that match a route will be checked by CORS Middleware before coming to the Security layer.

Security layer includes:
JWT Authentication Middleware: verify SignUp, verify token
Authorization Middleware: check User's roles with record in database
An error message will be sent as HTTP response to Client when the middlewares throw any error .

Controllers interact with MongoDB Database via Mongoose library and send HTTP responses (token, user information, data based on roles…) to Client.

We have 2 endpoints for authentication:

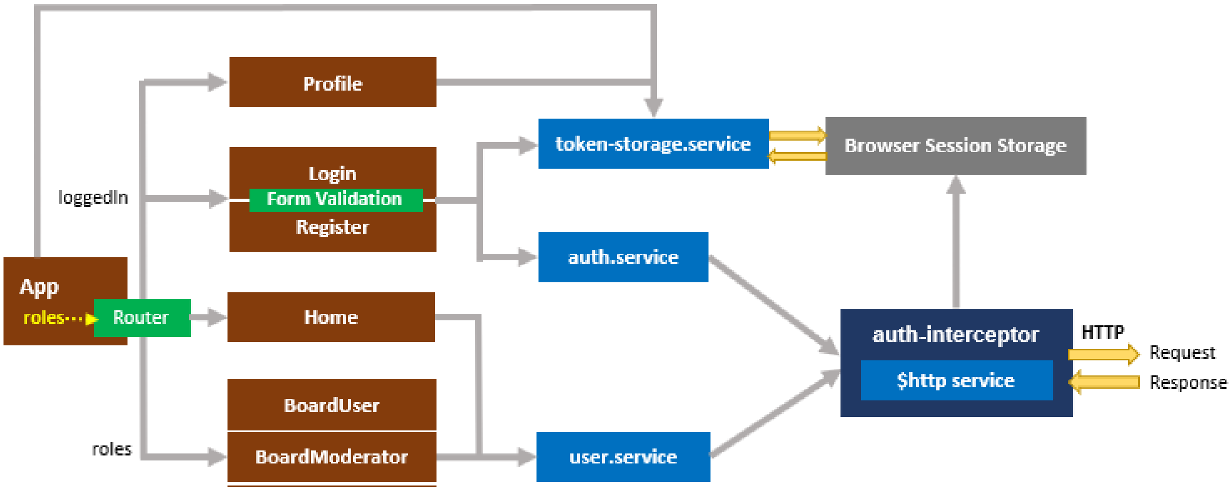
- api/auth/signup for User Registration
- api/auth/signin for User Login


Angular App
Back-end
User Registration
1 POST api/auth/signup
{ username, email, password }
2 Check existing
Save User to database
3 return Message ("Registered successfully!")
User Login
1 POST api/auth/signin
{ username, password }
2 Authenticate { username, password }
Create JWT string with a secret
3 return JwtResponse
{ token, type, user info, authorities }
Access Resource
4 Request data with
JWT on Authorization Header
5 Check JWT Signature
Get user info & authenticate
Authorize using user's Authorities
6 return Response based on Authorities

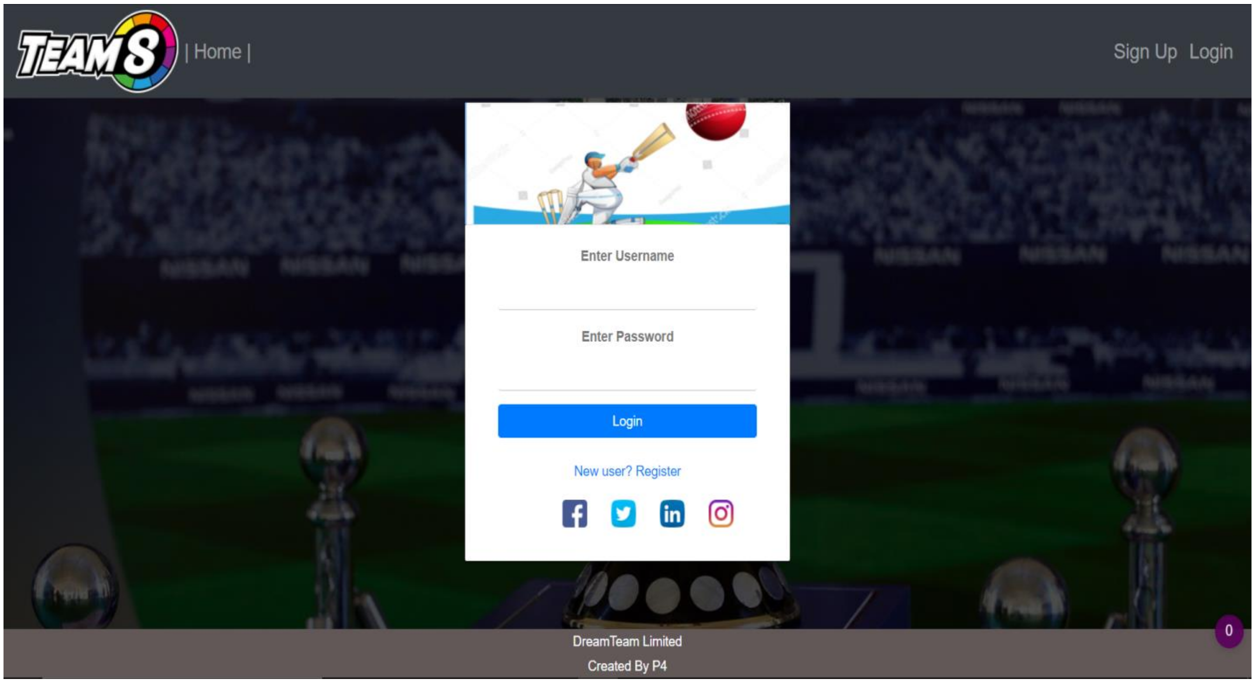
Frontend overview:

The App component is a container using Router. It gets user token & user information from Browser Session Storage via token-storage.service. Then the navbar now can display based on the user login state & roles.

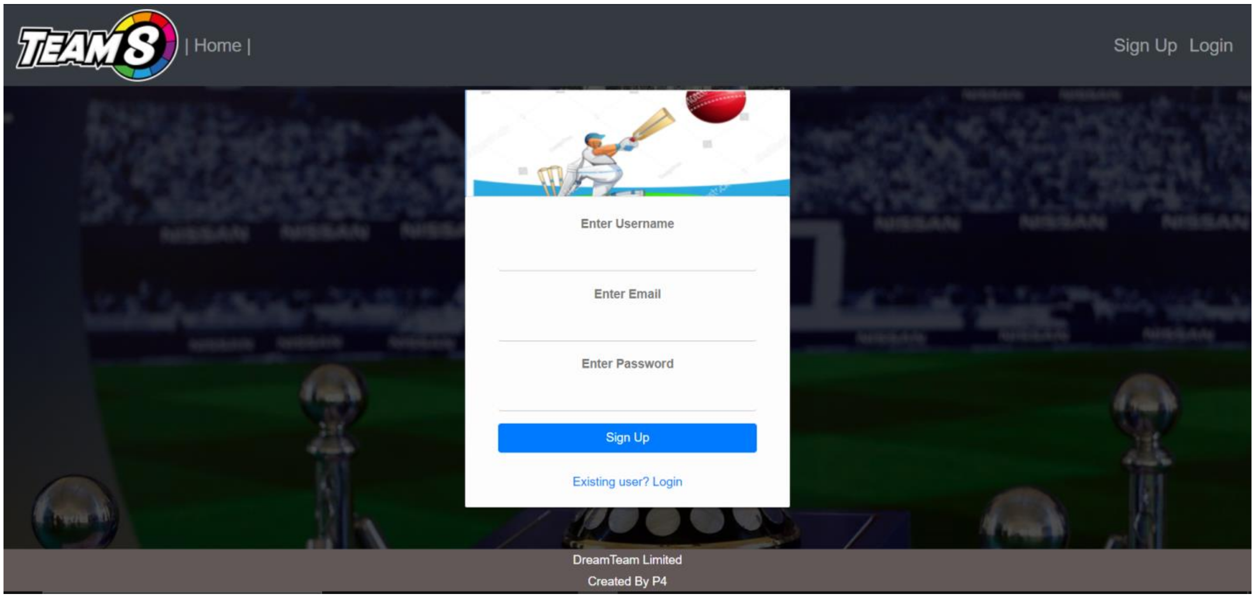
– Login & Register components have forms for submission data (with support of Form Validation). They use token-storage.service for checking state and auth.service for sending signin/signup requests.



1.1 Login Page

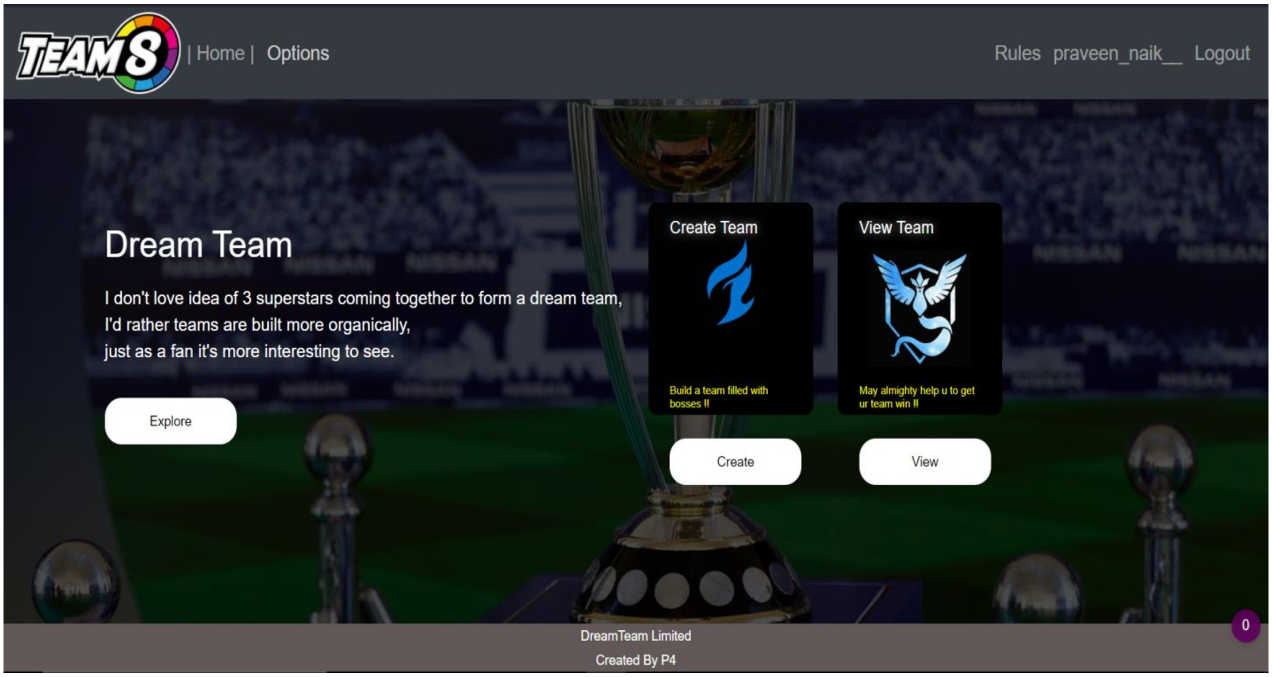


1.2 Register Page



2. User Portal

This page displays the options for the user. It is completely independent of the backend because it is a general page displayed for every user. The username and logout options will be visible at the top right corner of the page.


Dream Team

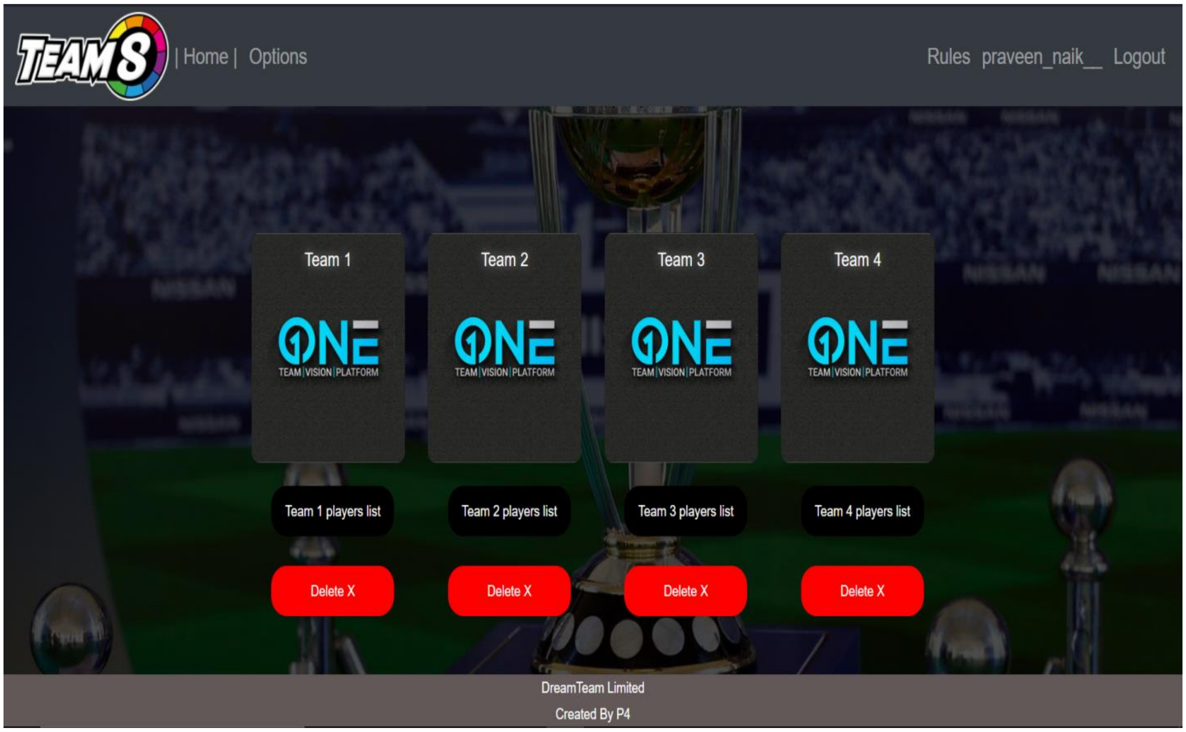
I don't love idea of 3 superstars coming together to form a dream team,
I'd rather teams are built more organically,
just as a fan it's more interesting to see.

Build a team filled with bosses !!

May almighty help u to get ur team win !!

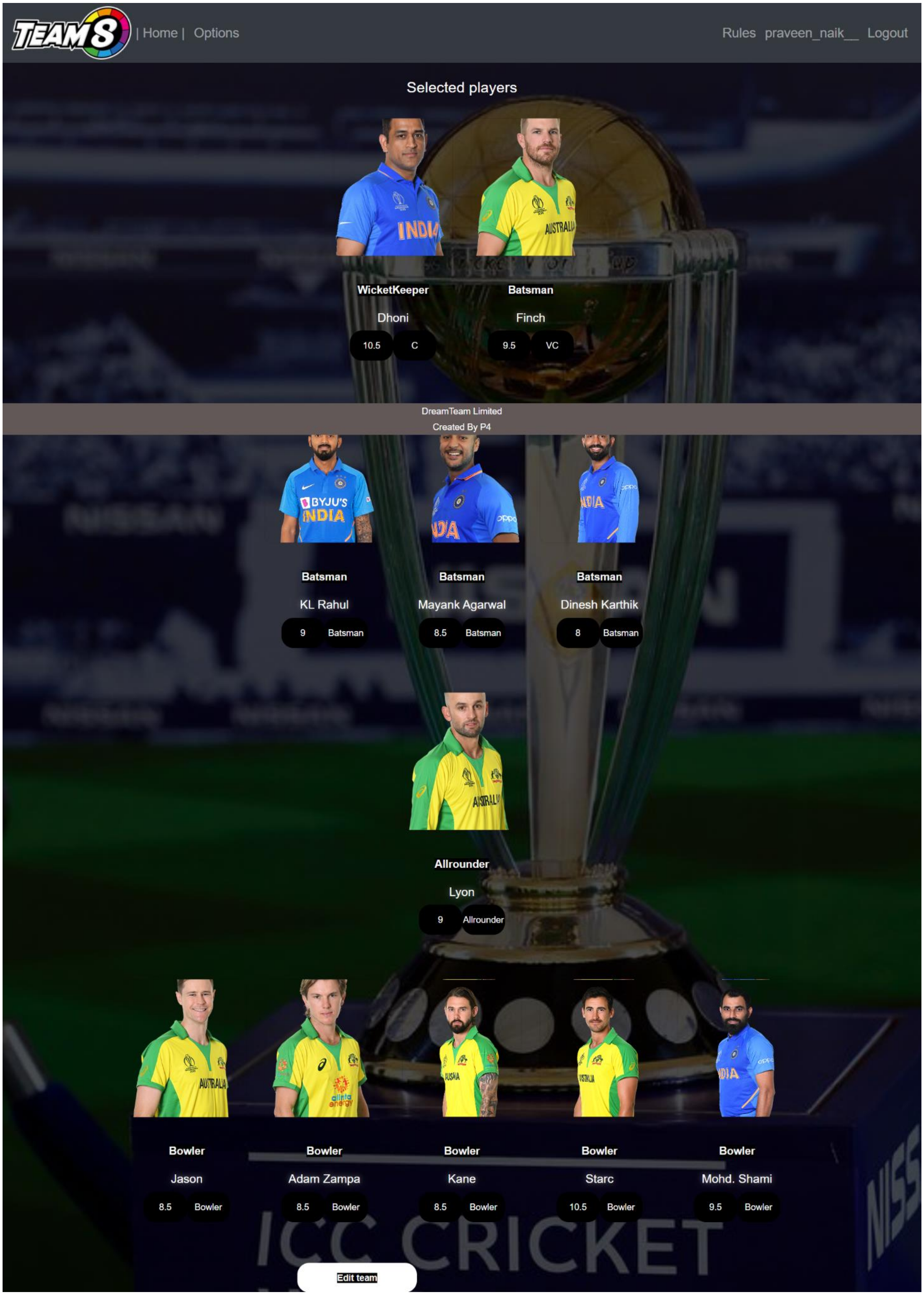
3. View my teams

In this page users can see all the teams they have created so far. Under each team name there will be an option to delete that team which will directly delete the team from the database.
Backend provides all the team information by querying with username in the database.
Backend API : /api/play/getAllTeams



4. View Team

In this page, users can see any specific team details such as players, points, captain, etc.
Backend API : `/api/play/getOneTeambyID`



Selected players

WicketKeeper
Dhoni
10.5 C

Batsman
Finch
9.5 VC

Batsman
KL Rahul
9 Batsman

Batsman
Mayank Agarwal
8.5 Batsman

Batsman
Dinesh Karthik
8 Batsman

Allrounder
Lyon
9 Allrounder

Bowler
Jason
8.5 Bowler

Bowler
Adam Zampa
8.5 Bowler

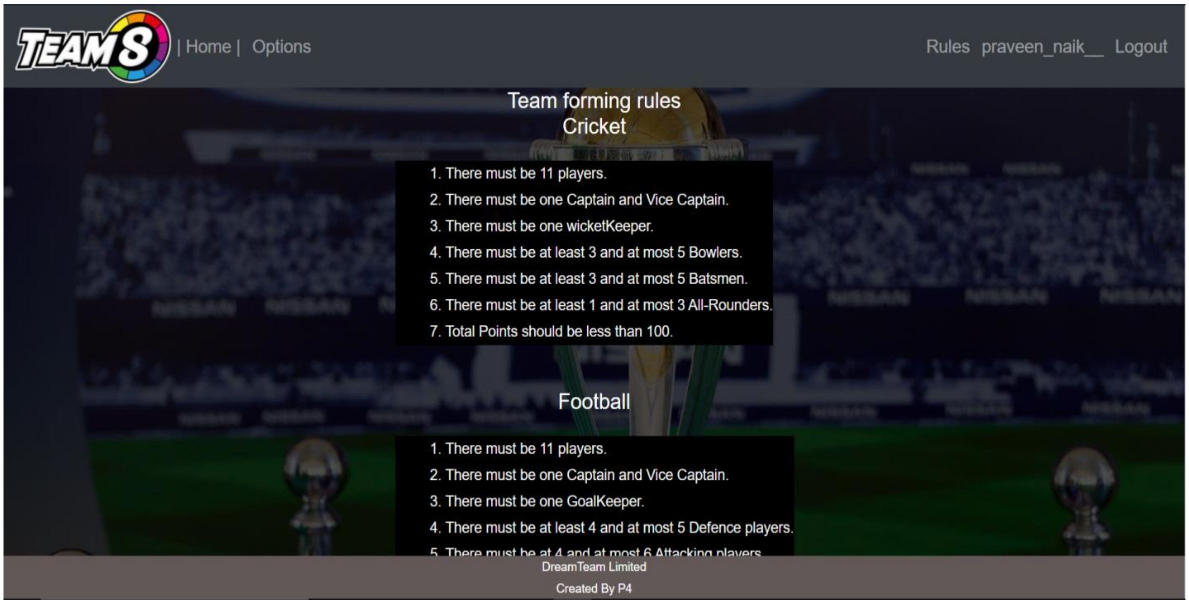
Bowler
Kane
8.5 Bowler

Bowler
Starc
10.5 Bowler

Bowler
Mohd. Shami
9.5 Bowler

5. Rules

This page is used to display the rules to the users.
It is completely independent of the backend because it is a general page displayed for every user.



Team forming rules
Cricket

1. There must be 11 players.
2. There must be one Captain and Vice Captain.
3. There must be one wicketKeeper.
4. There must be at least 3 and at most 5 Bowlers.
5. There must be at least 3 and at most 5 Batsmen.
6. There must be at least 1 and at most 3 All-Rounders.
7. Total Points should be less than 100.

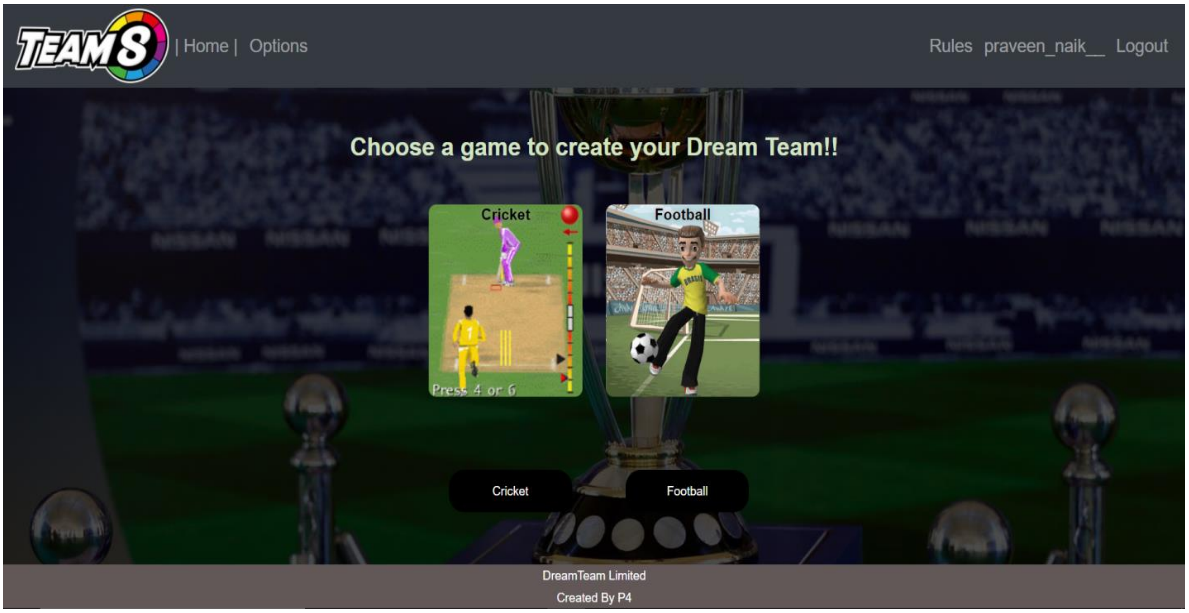
Football

1. There must be 11 players.
2. There must be one Captain and Vice Captain.
3. There must be one GoalKeeper.
4. There must be at least 4 and at most 5 Defence players.
5. There must be at 4 and at most 6 Attacking players

DreamTeam Limited
Created By P4

6. Create a team

In this page, Users can select the sport of which they want to create a team.
It is completely independent of the backend because it is a general page displayed for every user.



Choose a game to create your Dream Team!!

DreamTeam Limited
Created By P4

7. Create Cricket Team

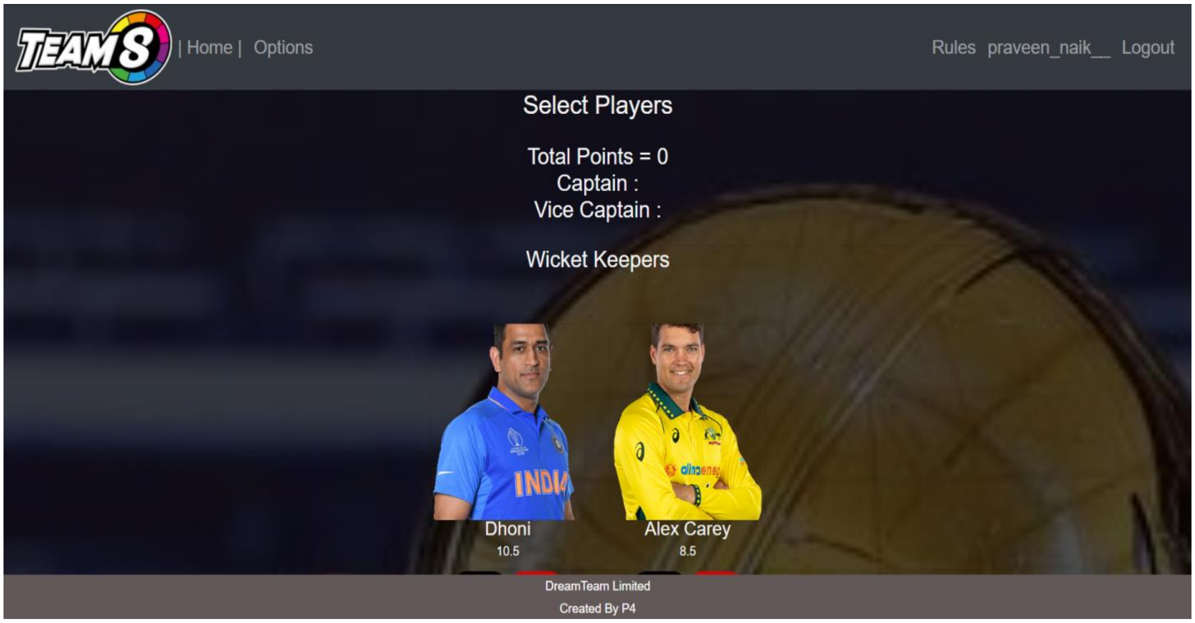
In this page, Users can select players they want to add in to create a team.

Rules which we have created for constructing a cricket team:

1. There must be 11 players.
2. There must be one Captain and Vice Captain.
3. There must be one wicketKeeper.
4. There must be at least 3 and at most 5 Bowlers.
5. There must be at least 3 and at most 5 Batsmen.
6. There must be at least 1 and at most 3 All-Rounders.
7. Total Points should be less than 100.

Backend API’s Used:

- `/api/play/getCricketPlayers`
- `/api/play/getPlayerbyID`
- `/api/play/createteam`



Select Players

Total Points = 0
Captain :
Vice Captain :

Wicket Keepers

Dhoni
10.5

Alex Carey
8.5

DreamTeam Limited
Created By P4

8. Create Football Team

In this page, Users can select players they want to add in to create a team.

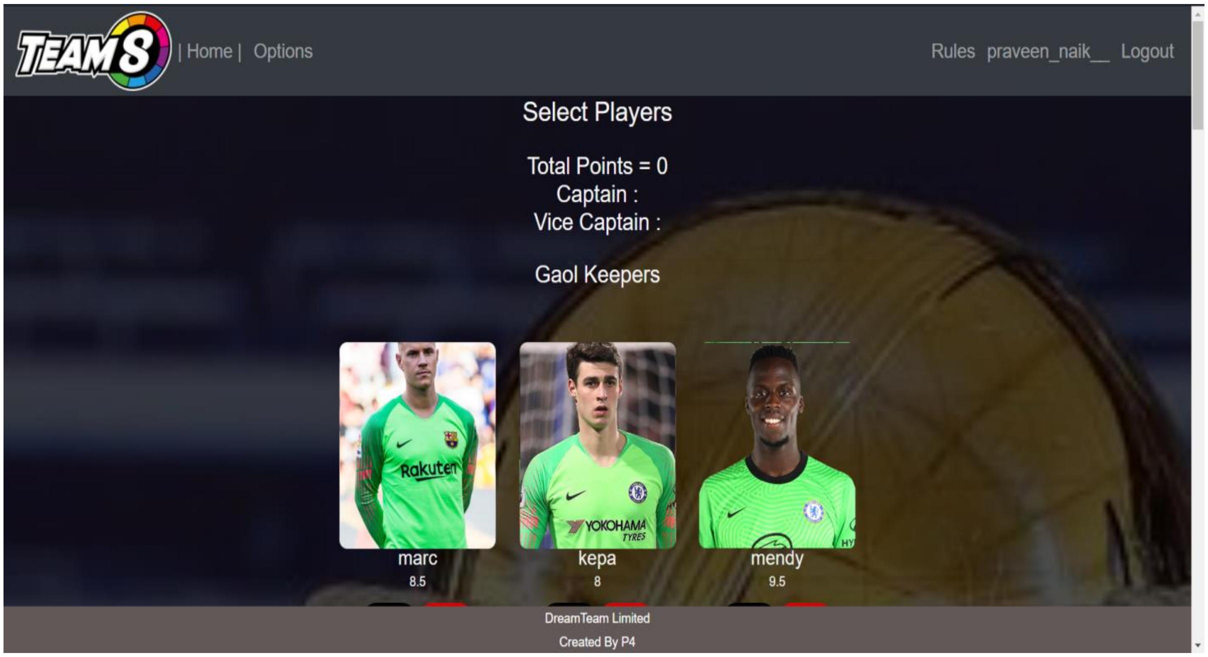
Rules which we have created for constructing a cricket team:

- There must be 11 players.
- There must be one Captain and Vice Captain.
- There must be one GoalKeeper.
- There must be at least 4 and at most 5 Defence players.

- There must be at least 4 and at most 6 Attacking players.
- Total Points should be less than 100.

Backend API’s Used:

- /api/play/getFootballPlayers
- /api/play/getPlayerbyID
- /api/play/createteam



Select Players

Total Points = 0
Captain :
Vice Captain :

Gaol Keepers

marc
8.5

kepa
8

mendy
9.5

DreamTeam Limited
Created By P4

VII. ADVANTAGES OF THE FRAMEWORK

We are using **MEAN** as our Framework for our Project.

What is meant by Mean ?
MEAN is a full stack software bundle. It uses JavaScript frameworks for front-end, back-end tools as well as for libraries. The database is a document-driven NoSQL database. Thus, MEAN stack is a full stack web development tool. It is used to develop web applications that are fast and efficient, and primarily used to build dynamic websites.

What does Mean stand for ?
Mean is an abbreviation of four web development technologies that make up a full stack tool.They are MongoDB, Express.JS, Angular.js and Node.js .It is user friendly and still used as a popular stack tool.Full-stack development tools can also used with MEAN

stack to making code easier.More full stack web development tools can be used along with MEAN stack to making coding easier.MEAN is not the order of stack.

What is MongoDB?

MongoDB is the database component of MEAN stack. It is a NoSQL database where a document is like an Object.It is written in C++,Python, Javascript. It is document oriented and supports multiple platforms. It is fast and can handle a large quantity of unstructured data.It is an open source software.

What is Express.js?

A Framework for the backend Node js web development too Web Applications and APl's.It is used for applications that have a server that will listen for connection requests from the client-side.

What is Angular.js?

The front-end framework component of MEAN stack is Angular.js. It is a licensed open source web development tool. It was developed by Google to counter challenges in Single page applications. You can build web and mobile applications using AngularJS. It allows HTML as its template language and is easy to learn and use.

What is node.js?

The back-end framework component of MEAN stack is node.js. It is a licensed open source web development tool. It is a Javascript runtime environment that runs outside the browser. It is platform independent, lightweight, and efficient. It's one of the best tools for real-time applications. It is used to develop server-side and network applications.

Why are we using Mean Framework in our Project ?

- We are using Mean Framework because it uses MongoDB, a nosql database for modern technique and it is a powerful way to store and retrieve data that allows the developers to move fast and it also supports high volume of database.MongoDB represents the data with natural clusters and variability of time or in structure.
- Usage of ExpressJS will help you to build server side applications very quickly and efficiently and it also makes applications faster and smarter.The characteristics of the Express.js are minimalism, flexibility, scalability.
- Usage of NodeJS allows event driven programming to the web servers, enabling development of fast web servers in Javascript and we can also create scalable servers without using threading, by using only a simplified model of event driven programming that basically uses callbacks to signal the completion of task

- AngularJs framework is basically a structural framework for dynamic web applications and it lets us to see HTML as our template language and it also extends HTML's syntax to express our application's components clearly. AngularJS is written in a typescript which is basically a typed superset of the Javascript which compiles to a plain Javascript. Typescript can improve the quality of angularjs applications.

VIII. CONCLUSION AND FUTURE WORK

In this project, we have designed a game playing application where a user can play a game of his choice and those games will have a certain set of rules on building a team. The proposed application has been designed with Mean Framework in which four web technologies have been used namely MongoDB,
Express.js, Angular.js, and Node.js. Usage of these web technologies have made our project an efficient one.
In this project, we have designed a game playing application where a user can play Cricket and Football fantasy games.In future, we are developing an application to the next level where a user can not only play Cricket and Football but also different sports like Basketball, Baseball, Kabbaddi,..etc. We will also develop our application such that users can join the contests of his choice.