# Web Platform for Soccer Fanatics

Uche D. Anota,Olajide Oladapo, Okeke Adaeze Lesley, Eugene Domson-Appiah
Leonard Alajemba, Ariful Islam.

**GITHUB REPOSITORY**: https://github.com/AAP-FH-KIEL/AAP_Project.git

## ABSTRACT

The increasing love for football by supporters and fans, have been seen in their unselfish loyalties to their adopted team. Hence, this project is focused on designing a soccer web platform consisting of a selection of twenty(20) clubs with the highest fan base from different leagues such as (Champions, La Liga, Bundesliga, premier league e.t.c) in order to provide complete information of the teams and clubs to their respective loyal fans.

The website has been designed using frameworks such as HTML5, Javascript, Bootstrap, MongoDB expressJS, NodeJS and PassportJS, with a notion of upgrading features in the future. This project will serve soccer fanatics with better control of sporting information.The main motive behind this project is to develop a user friendly web platform that provides information to our users concerning their favourite soccer teams,players,news,games statistics and lineups, fixtures and points table,teams and player profiles, articles and social networking.

**Keywords:** soccer fans,soccer,football news, football matches,soccer website

## 1. INTRODUCTION

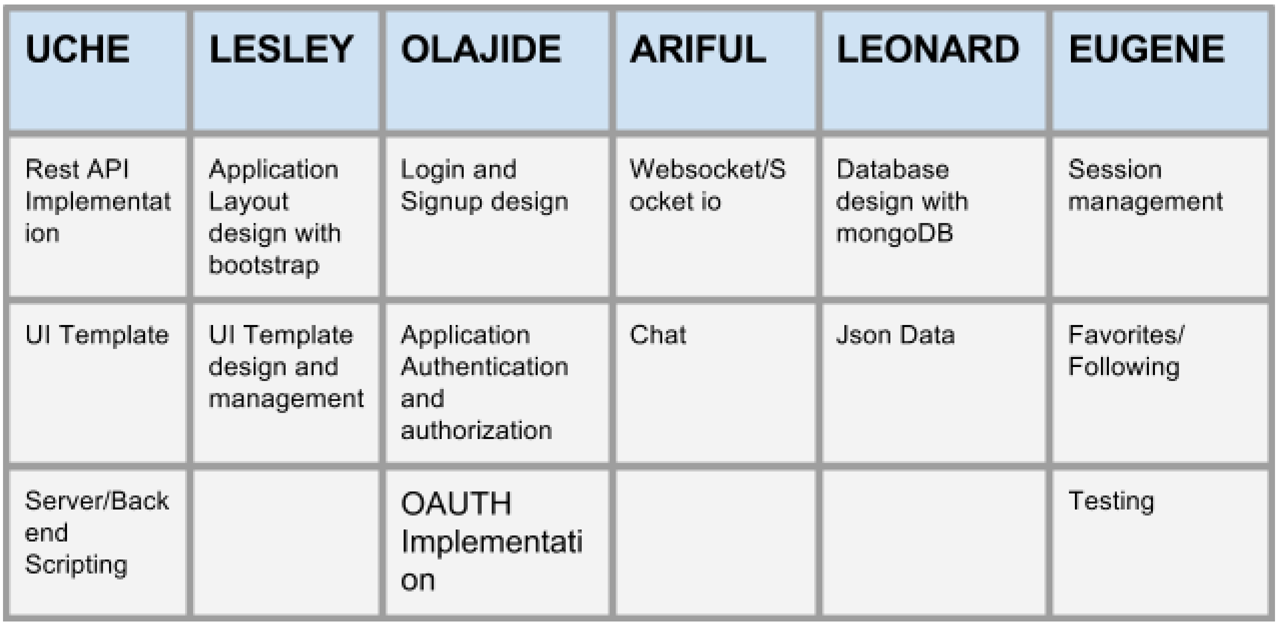
SoccerFanz is an online information website for everything sports. This website is developed by a team of javascript programmers for the Masters module course Advanced Application Programing. For the purpose of this documentation, football and soccer will be referred to as one and the same.The website is developed with javascript as its programming language.While many websites exist with similar goals, SoccerFanz is unique with functionalities.

Alongside giving football news and articles about teams, players and matches, SoccerFanz has a community chatroom for fans all over the world to discuss and connect with other fans their views and opinions. There are fans that would like to closely follow teams that they like and for this we provide a follow/favorite option. Under this, fans will get their own personalised news about their teams chosen. they will get news of transfers, goals, fixtures and points table. These functionalities will also be available as the general settings of the main website but gives a tailored dashboard for users favorites.

SoccerFanz as a website will as well use API's to give breaking news so that users will always know what's happening in the football world;Transfer rumours etc. As with all other football websites, we will have these following features to offer our users. Game statistics, Upcoming matches,Line-ups and live commentary. However, in the nearest future SoccerFanz will be able to run on mobile devices and offline as desktop applications both for windows and OSX.

The rest of this documentation will focus on the functionalities of our application, the architecture and implementation and technology used.

## 2. MATRIX OF TEAM MEMBER/TASK ASSIGNMENT



| UCHE | LESLEY | OLAJIDE | ARIFUL | LEONARD | EUGENE |
|---|---|---|---|---|---|
| Rest API Implementation | Application Layout design with bootstrap | Login and Signup design | Websocket/Socket io | Database design with mongoDB | Session management |
| UI Template | UI Template design and management | Application Authentication and authorization | Chat | Json Data | Favorites/ Following |
| Server/Back end Scripting | | OAUTH Implementation | | | Testing |

*TABLE 1.0*

During the development of SoccerFanz, a matrix was designed for everyone to work on. As a team of six members the distribution of the tasks has been categorized in a tabular form below with member names corresponding to their tasks given.

## 3. METHODOLOGY

### 3.1 CLIENT SIDE

#### 3.1.1 JAVASCRIPT

Javascript is the primary language that was used for the development of our application. It is a scripting language that is used to program the client side browser and cuts across all the other frameworks used in our application.

#### 3.1.2 BOOTSTRAP

For developing our front end, we used Twitter bootstrap. Bootstrap is an open source tool set that is used to develop HTML,CSS and javascript [8]. Most of the application layout of our application is done with Bootstrap. The font, buttons and all the styling is an example.

### 3.2 SERVER SIDE

#### 3.2.1 NODE JS

Node Js the platform used for our backend side of our application. It is built on google chrome's javascript runtime environment for building easy, fast and scalable network application. It is an event driven and non blocking model that is used for data intensive and real time application that is run across distributed environment [7].

## 3.3 MIDDLEWARES/FRAMEWORK

### 3.3.1 EXPRESS JS

According to [2], “Express Js is a fast un opinionated, minimalistic web framework for Node.js”. It works together with Node.js to ensure that web application development is done easily and fast. The use of Express.js in the design of the SoccerFanz website allowed for ease of configuration and customization, for defining routes of the application with regards to HTTP methods and URLs. It also aided with the creation of REST API Server and ease in connecting with the implemented database which is the (MongoDB).

There are two ways of installing Express.js using Npm [2]:

1. Installing Express globally on so that every node.js application on that machine can access it using this command:

```
npm install -g express
```

2. Installing Express.js to a local project folder with this command:

```
C:\MyNodeJSAPP> npm install express --save
```

### 3.3.2 AJAX

For the handling of our forms and reloading some static pages, we used Ajax to submit these forms and also to reload the pages at the background.

### 3.3.3 PASSPORT JS

The integration of a good authentication strategy for any application can be a frustrating task and can also be experienced in a node.JS application[1]. However, to resolve all authentication concerns, Passport.JS an authentication middleware was introduced as a security control measure and an authentication system to differentiate between users. This would help each user that is logged in get a customizable feed on their individual dashboards depending on which players, league or teams they are following and their favourites. The authentication page which offers a login feature for users also has a signup page for new users. Some of the reasons that led to the choosing of Passport.JS includes its ability to work flawlessly with express and its flexibility [4]. Passport allows OAUTH sign on capabilities which we make use of. An examples would be signing in or signing up as a User using the said user’s already existing facebook, twitter, github, google account etc. As a standard feature with Passport.js, the application could also verify using its own databases for authentication by looking up the entered data with already stored data in our website application’s database.Passport allows us to route and output our authentication .

## 3.4 DATABASE

The database part of our application would be used to store all the data from the users in a form of JSON data. Though there are lots of NOSQL Databases that accept data in JSON formats including MongoDB, CouchDB, Cassandra etc, We chose MongoDB as the technology to use.

### 3.4.1 MONGODB

MongoDB assumes the database that stores data for one’s application[4]. It’s ease of deployment, high availability,high performance and ability to support multiple storage made it more preferable over other databases like CouchDB etc. MongoDB is a distributed database so has a high availability which made it more feasible to use with our web application.

### 3.5 API

Our application depends a lot on APIS because it needs to update on live and current events in the football world. SoccerFanz as a web application connects to these API's to keep current.

### 3.6 WEBSOCKET

Websocket provides full duplex connection over a single TCP connection. The websocket allows the browser and the web server to communicate which facilitates real time data transfer and communication to and fro.

Socket io which was used for the implementation of the websocket is a javascript library that is used to enable real time bidirectional communication between the client and the server [6].

Websocket was implemented in SoccerFanz application to provide news broadcast capabilities, fans chatroom, and live commentary of matches. The mode of setup and workings of websocket in SoccerFanz are detailed in the diagram below

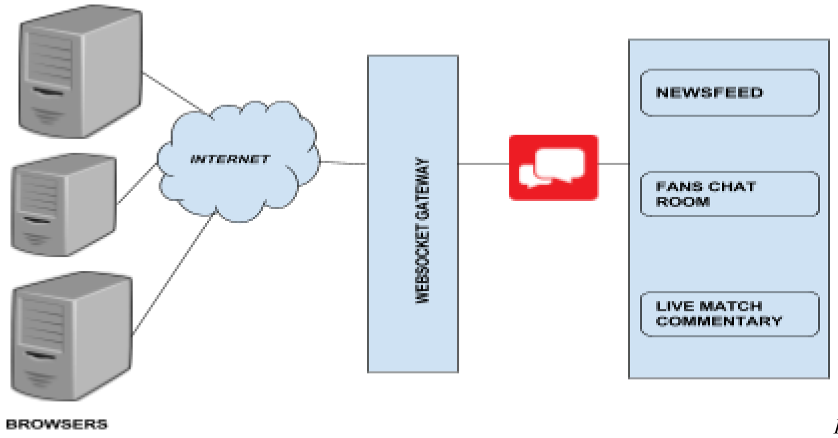


*FIG 1.0*

### 3.7 UI TEMPLATES

The client side of our website consists of HTML and styled with CSS. Bootstrap is used to design the UI templates for all the forms and pages to be used. We chose to use bootstrap for our UI because of its customizability and responsiveness. Another distinct reason for its use was ease of integration with other platforms. Our CSS we used could easily be combined with our bootstrap elements we implemented.

## 4.0 ARCHITECTURE

### 4.1 STRUCTURE OF OUR APPLICATION

1) *APP* - This is a server that manages the system logic, controls the route, models, the chat logic and also handles the database. It also contains all the dependencies for the application.
   a) *Routes* - This manages the users request and forwards them to the appropriate controller.
   b) *Views* - This handles the UI displayed to the user through EJS and its displayed as HTML to the browser..
   c) *Models* - the Model contains the structure and schema of the users data.
   d) *Public Folder* - This contains all our static files which includes images and videos.

SoccerFanz website is based on Model View Controller architectural approach. A simple flow diagram shows the links and libraries used for the development of SoccerFanz.The figure below shows the look of the architecture of the website.

# ARCHITECTURAL OVERVIEW

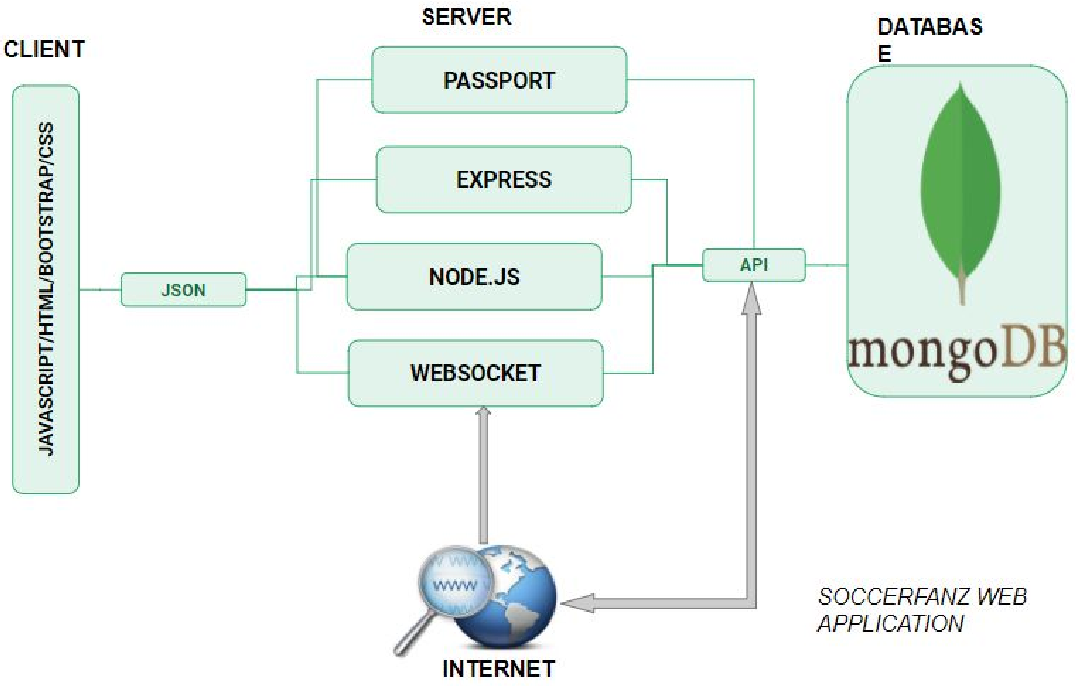


FIG 2.0

## 5. RESULTS

At the end of development of our website application, we successfully implemented the functionalities that required the application to fully operate. Soccerfanz implemented websocket with live commentary, a session page that gave tailored feeds of the user. Data from API being used fetched and populated as required by the users' session. Users are able to be authenticated by user of the website database or any of the sign in methods chosen. Below are the screenshots showing the user interfaces of SoccerFanz website application. and its outputs.

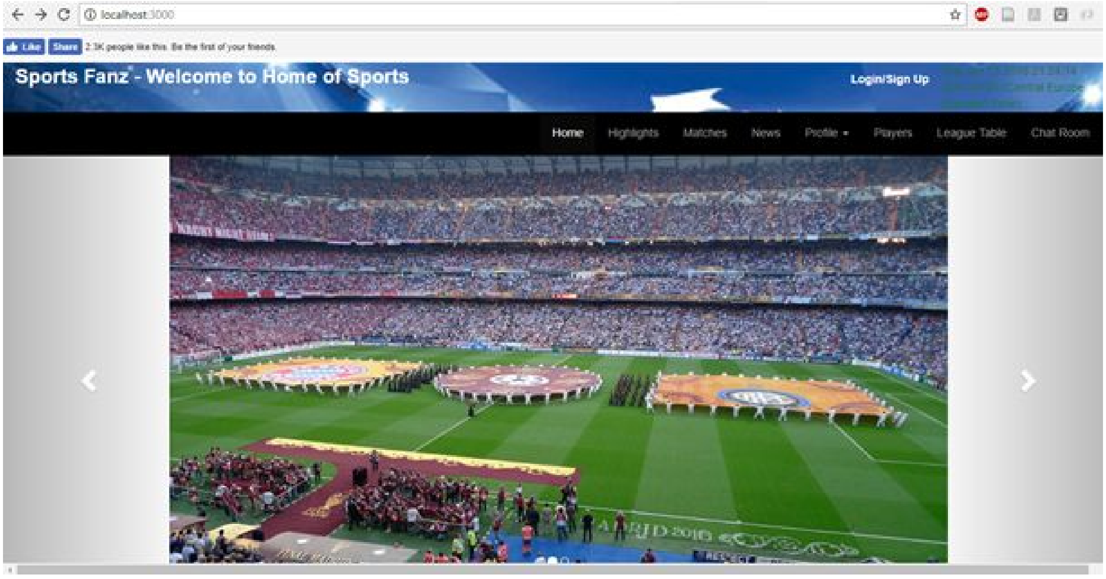


*FIG 3.0* Homepage

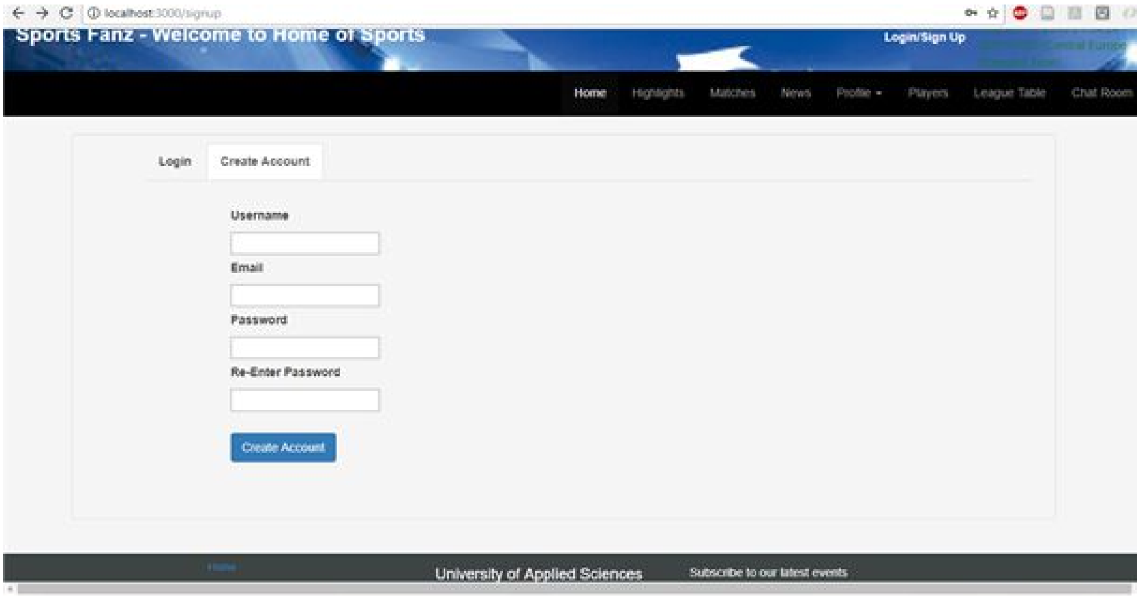
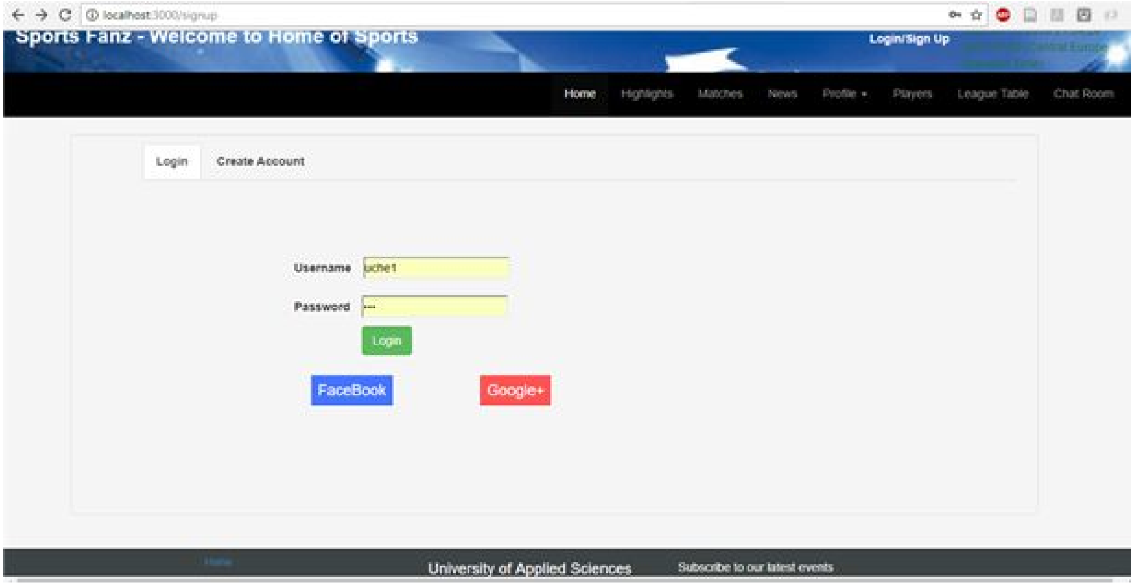


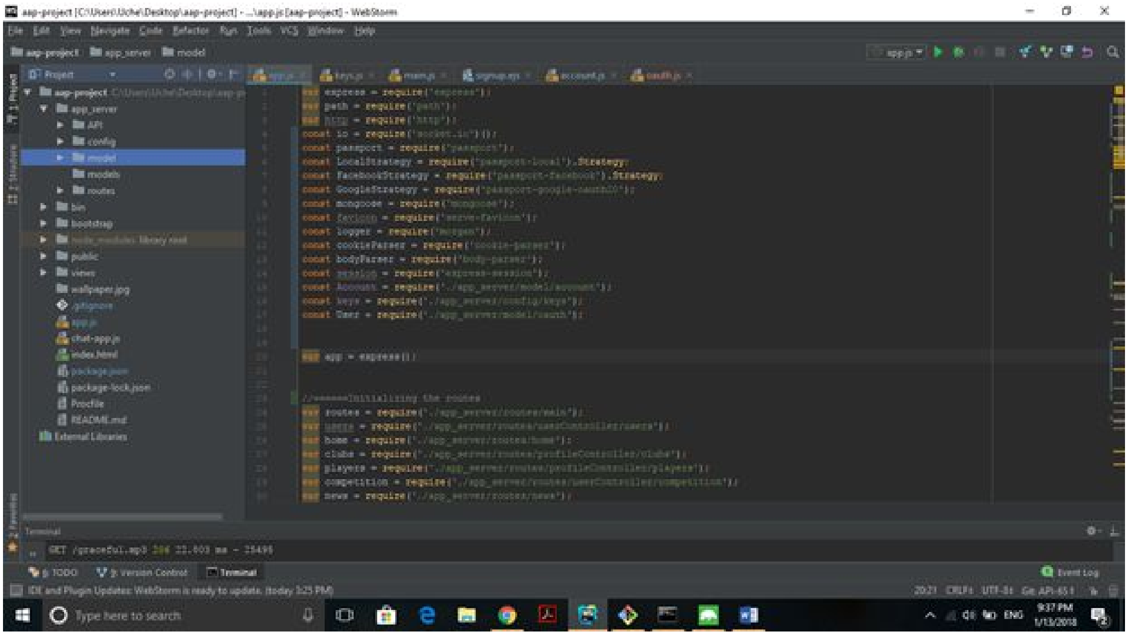
*FIG 3.1 SIGNUP PAGE*



Username uche1

Password ···

*FIG 3.2 LOGIN PAGE*



```
var express = require('express');
var path = require('path');
var http = require('http');
const io = require('socket.io')();
const passport = require('passport');
const LocalStrategy = require('passport-local').Strategy;
const FacebookStrategy = require('passport-facebook').Strategy;
const GoogleStrategy = require('passport-google-oauth20');
const mongoose = require('mongoose');
const favicon = require('serve-favicon');
const logger = require('morgan');
const cookieParser = require('cookie-parser');
const bodyParser = require('body-parser');
const session = require('express-session');
const Account = require('./app_server/model/account');
const keys = require('./app_server/config/keys');
const User = require('./app_server/model/oauth');


var app = express();


//======Initializing the routes
var routes = require('./app_server/routes/main');
var users = require('./app_server/routes/userController/users');
var home = require('./app_server/routes/home');
var clubs = require('./app_server/routes/profileController/clubs');
var players = require('./app_server/routes/profileController/players');
var competition = require('./app_server/routes/userController/competition');
var news = require('./app_server/routes/news');
```

```
GET /graceful.mp3 206 22.603 ms - 25498
```

*FIG 3.3 SERVER AND APP DIRECTORY*

## 6. DISCUSSIONS AND LIMITATIONS

The aim of designing this web platform, is to provide football clubs loyalists a platform with unlimited information and access to live football streams about their adopted teams. However, even though similar types of this website exist, the main requirements regarding the website design is to ensure that the front and backend is designed with Javascript, to entail a git version control, user sign-up/authentication, database access and websockets. However, there are lots of challenges that was encountered in the designing this website. The first is the difficulty in choosing from a wide range of frameworks that best suites in ensuring that all the requirements were met. Second is learning how to implement this tools into our project. We also had challenges in getting some of the api's needed for this application as most of the available api's are commercially available. Finally, designing such an app is a time consuming process and to get all the functionalities to work requires a huge amount of time which is really a big challenge since we have other courses to cater for. In all, developing this application was a huge learning experience for us all as most of the work we did were through spike. In the future we intend to extend the functionalities of this app to meet the expectation of the sports fans all around the world

## 7. CONCLUSIONS AND OUTLOOK

The motive behind this project was to show the use of different javascript frameworks, database, html,css, bootstrap and version control in designing a web application. To ensure that all these requirements were met, Html , css and bootstrap was used to design the user interface, MongoDB was implemented as the database, a websocket was implemented for chatting between fans realtime, and the passport.js was also used for authentication of users for signing up/login in to view favourites and follow players and also for chatting purpose.

However, there are several things that will be done in the nearest future to enhance the user friendliness of the web application which includes; being able to run the application on mobile devices and also provide offline services as desktop applications both on windows and Osx.

## REFERENCES

[1] Mangal, Agraj. "Authenticating Node.js Applications With Passport." Code Envato Tuts , 11 Aug. 2014, https://code.tutsplus.com/tutorials/authenticating-nodejs-applications-with-passport--cms-21619

[2] "Express routing." [Online]. Available: https://expressjs.com/en/guide/routing.html. [Accessed: 01-Jan-2018].

[3] "Authenticating Node.js Applications With Passport," *Code Envato Tuts+*. [Online]. Available: https://code.tutsplus.com/tutorials/authenticating-nodejs-applications-with-passport--cms-21619. [Accessed: 07-Jan-2018].[1]

[4] "Passport." [Online]. Available: http://www.passportjs.org/. [Accessed: 07-Jan-2018].

[5] "MongoDB Documentation." [Online]. Available https://docs.mongodb.com/?_ga=2.34006689.1048438367.1515519070-1873105649.1515519070.[Accessed: 09-Jan-2018].

[6]"Socket.IO — Docs." [Online]. Available: https://socket.io/docs/. [Accessed: 09-Jan-2018].

[7] N. js Foundation, "Docs," *Node.js*. [Online]. Available: https://nodejs.org/en/docs/. [Accessed: 11-Jan-2018].

[8] M. O. contributors Jacob Thornton, and Bootstrap, "Bootstrap." [Online]. Available: https://getbootstrap.com/. [Accessed: 11-Jan-2018].

[9]"OAuth Login (Passport.js) Tutorial," YouTube. [Online]. Available: http://www.youtube.com/playlist?list=PL4cUxeGkcC9jdm7QX143aMLAqyM-jTZ2x. [Accessed: 10-Jan-2018].

[10]"Working with Socket.IO," *Lynda.com - from LinkedIn*. [Online]. Available: https://www.lynda.com/Express-js-tutorials/Working-Socket-IO/502310/519151-4.html. [Accessed: 7-Jan-2018].

[11]"ESPN API - Live news headlines and articles from ESPN," *News API*. [Online]. Available: https://newsapi.org. [Accessed: 13-Jan-2018].

[12]"football-data.org - ur src for machine readable football data." [Online]. Available: http://api.football-data.org/. [Accessed: 8-Jan-2018]

[13]S. Holmes, *getting-MEAN: The source code for the application in my book Getting MEAN*. 2018