# MINI PROJECT

## (2020-21)

**Building and Deployment of Web application**

# UNPLUG THE PLAYERS

## MID-TERM REPORT

**Institute of Engineering & Technology**

***Submitted by: -***

Sarvesh Kumar Sharma (181500625)

Satyam Kumar Jha (181500627)

Ashutosh Tripathi (181500625)

Jeevesh Gangwar (181500625)

Sachi Tripathi (181500598)

***Supervised By: -***

**Mr. Vaibhav Diwan**

Technical Trainer

**Department of Computer Engineering & Applications**

**GLA University**

**Mathura- 281406,**

**INDIA**

# ABSTRACT

In this we will build a simple web application which will be based on QUIZ GAME related to the field of sport, ‘Football’. We will then deploy the that web application using Flask with python.

This work deals with the development of web-based application game namely: Unplug the Players. This application is developed for knowledge purpose, allowing the users to play a quiz and enhance their knowledge in the field of sports particularly Football. This application is based on giving the scores to the player who have attempted to play.

For the development of this web application, data is collected from different sources about the football players from all over the world. This application will make use of JavaScript, Flask, CSS, Python.

A photograph will be displayed on the screen along with some information about the player and the user have to identify the player based on that information. After each try the photograph displayed will be changed and a new photograph will be displayed on the screen. The player will be given several attempts and after every chance some message will be displayed according to the players performance and based on that scores will be awarded.

# Table of Contents

# Table of figures

# INTRODUCTION

## 1.1. General Introduction

To build a web-based application for the entertainment and knowledge of the people, where someone can play a quiz and be entertained and as well as can enhance his knowledge in that field.

A simple enjoyable game to test anyone’s knowledge in the field of sports and particularly in one sport as it will be a little problematic if every sport will be included in a single game.

This mainly focuses on the sport Football. Football is a family of team sports that involves, to varying degrees, kicking a ball to score a goal. Unqualified, the word football normally means the form of football that is the most popular where the word is used. Sports commonly called football include association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby union or rugby league); and Gaelic football. These various forms of football share to varying extent common origins and are known as football codes.

There are a number of references to traditional, ancient, or prehistoric ball games played in many different parts of the world. Contemporary codes of football can be traced back to the codification of these games at English public schools during the 19th century.

In 1888, The Football League was founded in England, becoming the first of many professional football competitions.

### 1.1.1. What is a Web application (Web app)

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. Web services are Web apps by definition and many, although not all, websites contain Web apps. According to Web.AppStorm editor Jarel Remick, any website component that performs some function for the user qualifies as a Web app.

Web applications can be designed for a wide variety of uses and can be used by anyone; from an organization to an individual for numerous reasons. Commonly used Web applications can include webmail, online calculators, or e-commerce shops. Some Web apps can be only accessed by a specific browser; however, most are available no matter the browser.

### 1.1.2. How a web application works

Web applications are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the program executable. Some of the applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server.

The web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information. Application server technology ranges from ASP.NET, ASP and ColdFusion, to PHP and JSP.

Here's what a typical web application flow looks like:

1. **User** triggers a request to the **web server** over the **Internet**, either through a web browser or the application's user interface
2. **Web server** forwards this request to the appropriate **web application server.**
3. **Web application server** performs the requested task – such as querying the **database** or processing the data – then generates the results of the requested data
4. **Web application server** sends results to the **web server** with the requested information or processed data
5. **Web server** responds back to the client with the requested information that then appears on the user's display

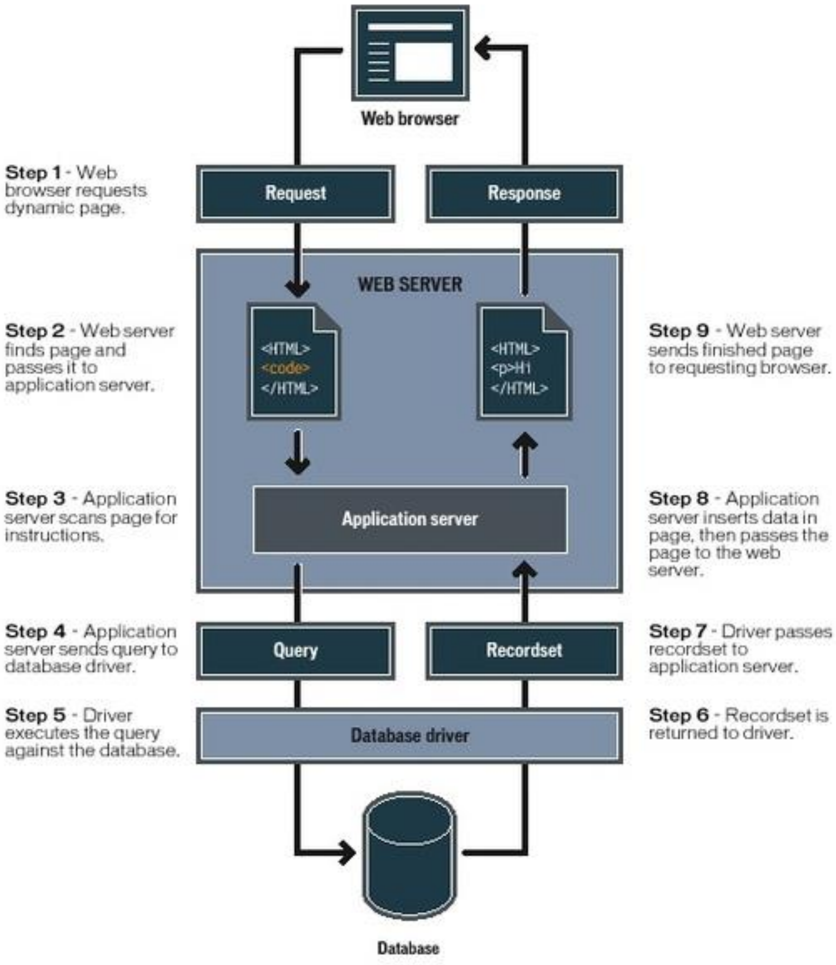

Web browser
Step 1 - Web browser requests dynamic page.
Request
Response
WEB SERVER
Step 2 - Web server finds page and passes it to application server.
<HTML>
<code>
</HTML>
<HTML>
<p>Hi
</HTML>
Step 9 - Web server sends finished page to requesting browser.
Step 3 - Application server scans page for instructions.
Application server
Step 8 - Application server inserts data in page, then passes the page to the web server.
Step 4 - Application server sends query to database driver.
Query
Recordset
Step 7 - Driver passes recordset to application server.
Step 5 - Driver executes the query against the database.
Database driver
Step 6 - Recordset is returned to driver.
Database

Figure 1: Web App Working

## 1.2. Area of Computer Science

Here in this project we have touched the area of Computer Science, Web application in which we have created a game using the JavaScript, Python, Flask, CSS etc. These are some fields used in this project up till now.

Many of the largest technology companies maintain large scale web applications, providing services such as social media, search, advertising and video and audio streaming.

## 1.3. Hardware Requirement

The requests of the hardware for the web application are as followed:

1. 64 bits laptop or desktop.

## 1.4. Software Requirement

To access a web portal of this application, its only need a PC/Laptop/Mobile with an integrated and updated web browser.

**Desktop browser**: Safari, Chrome, Firefox, Opera, IE9+.

**Mobile browsers**: Android, Chrome Mobile, iOS Safari.

**On the server side**: a PC/Web Server which meets these specifications:

1. Window Operating System
2. At least 2 GB RAM and 150 GB Free Space
3. Redis Server Installed
4. Python Compiler Installed

## 1.4. Dependencies

- Flask
- Python
  - ➢ SciPy
  - ➢ Pandas
  - ➢ NumPy
- JavaScript
- HTML
- CSS

# OBJECTIVE

The main objective of the project is to create a web app game which allows to explore our knowledge in football. It will be like quiz-based game where game player will be awarded different points on his correct answer at different stage. The pre-objective is to gather the complete data and pre-process the data on which our web app will run.

## Implementation Details

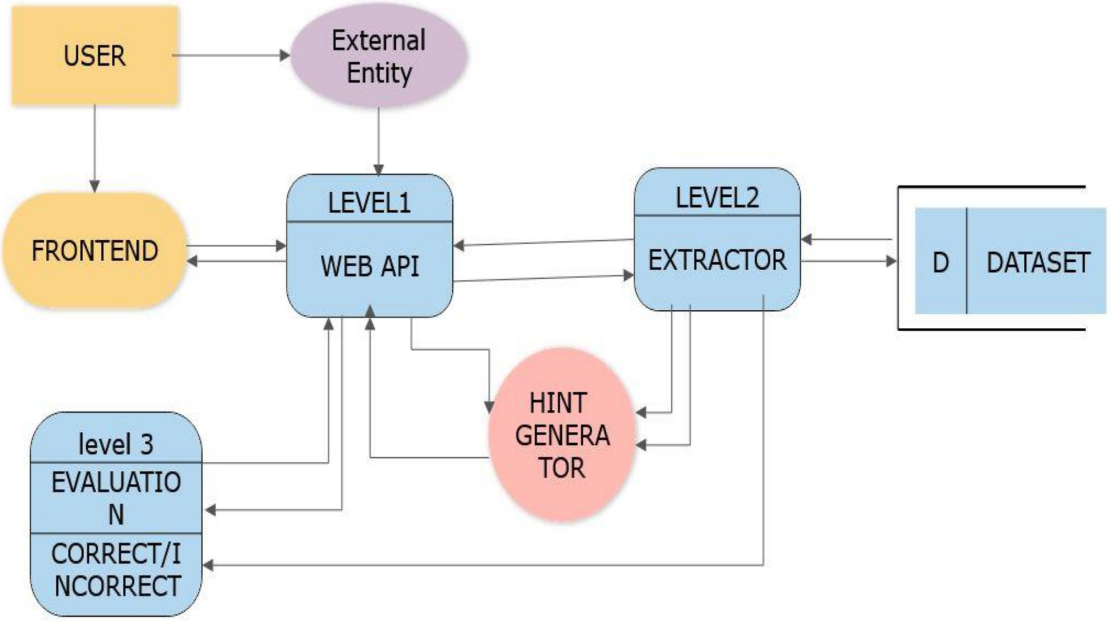


Figure 2: working of app

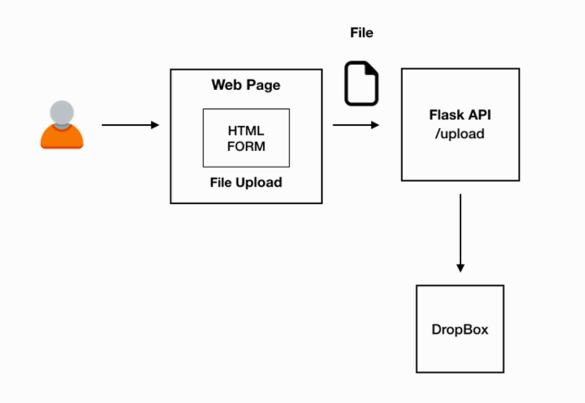


Figure 3: user view

**Part 1:**

To Collect the data of 1000 football players and compile them to finally Store them as a CSV file.

- To Collect players images.
- To collect players Country images.
- To collect player's Club Images
- All other information like nationality, age, potential and value.

**part 2:**

- Install the dependencies.
- Develop function to find similar type of players using SciPy spatial Distance.
- Develop function to clean the raw data.
- Develop UI for actual output of project.

**Part 3:**

- Develop game page.
- Develop game script.
- Make the application attractive and user friendly.

**Part 4:**

- Deploy the fully developed web app on Heroku platform.

# PROGRESS

Part 1 is completed

**Collecting all the data and store it in a .csv file.**

- Collected the images of the players.
- Collected unique hints and information about each player.
- Gathered information about the club or the country for which a particular player plays.
- Collected the images of the flags of their country or their club.
- Stored all the gathered data in data.csv file.

Part 2 is completed

**Developing the index page and the application using flask.**

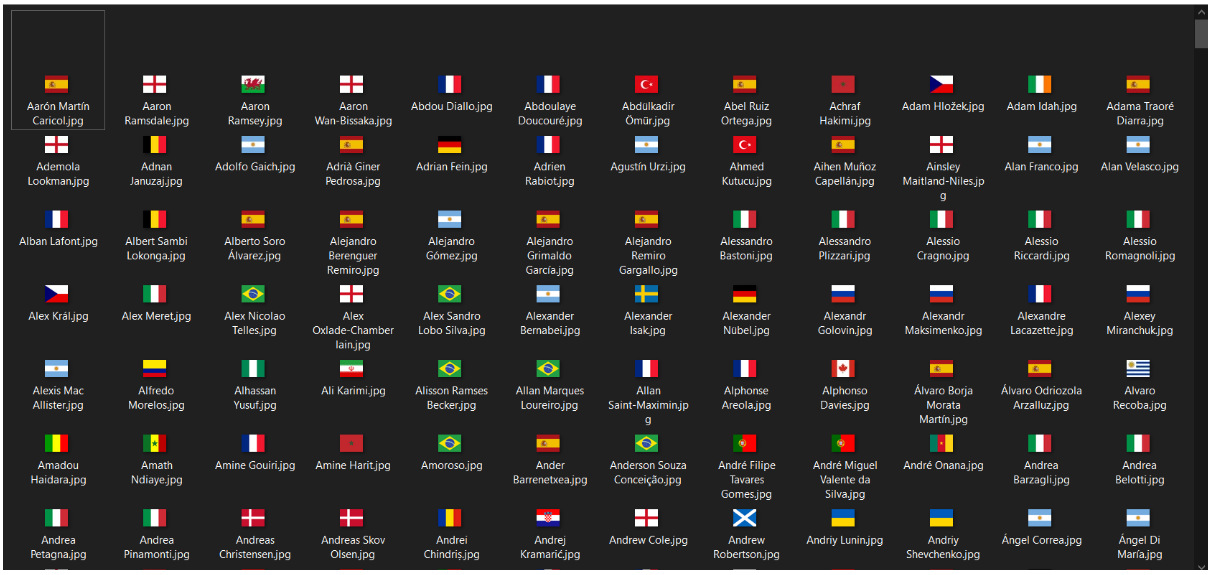
Before starting the coding part, we needed to install and configure some softwares and libraries to be used in this project. Those libraries and softwares are as follows:

- Pycharm
- gunicorn==20.0.4
- Flask==1.1.1
- itsdangerous==1.1.0
- Jinja2==2.10.1
- MarkupSafe==1.1.1
- Werkzeug==0.15.5
- numpy>=1.9.2
- scipy>=0.15.1
- scikit-learn>=0.18
- matplotlib>=1.4.3
- pandas>=0.19

After all the installation and configuration,

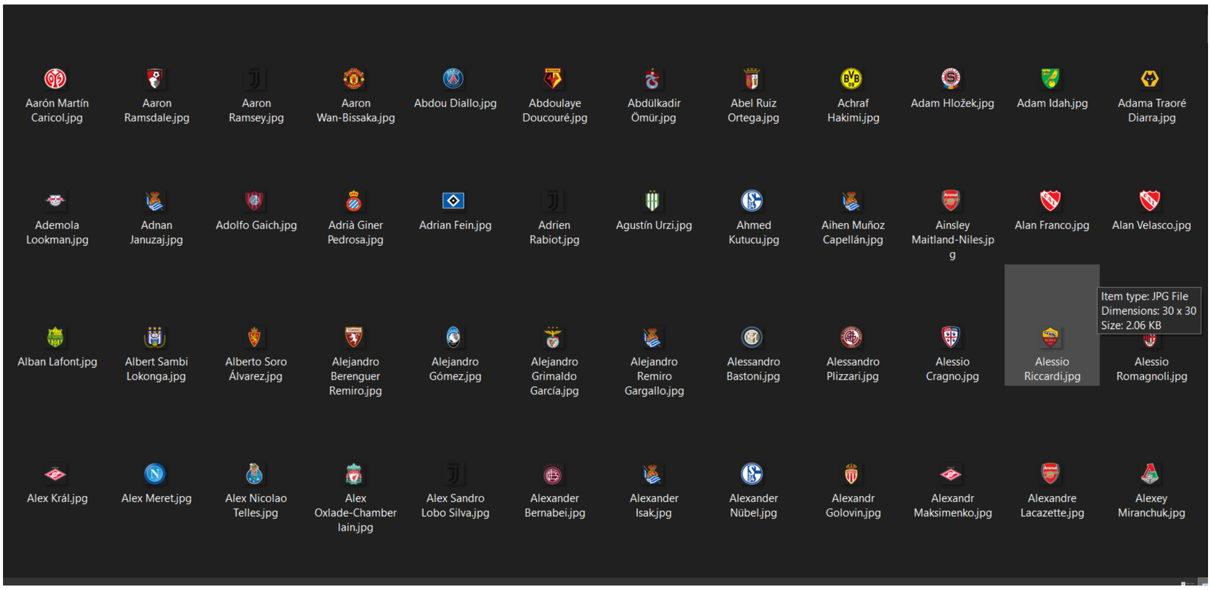
- Develop function to find similar type of players using SciPy spatial Distance.
- Develop function to clean the raw data.
- Develop UI for actual output of project.

# SCREENSHOTS



Aarón Martín Caricol.jpg
Aaron Ramsdale.jpg
Aaron Ramsey.jpg
Aaron Wan-Bissaka.jpg
Abdou Diallo.jpg
Abdoulaye Doucouré.jpg
Abdülkadir Ömür.jpg
Abel Ruiz Ortega.jpg
Achraf Hakimi.jpg
Adam Hložek.jpg
Adam Idah.jpg
Adama Traoré Diarra.jpg
Ademola Lookman.jpg
Adnan Januzaj.jpg
Adolfo Gaich.jpg
Adrià Giner Pedrosa.jpg
Adrian Fein.jpg
Adrien Rabiot.jpg
Agustín Urzi.jpg
Ahmed Kutucu.jpg
Aihen Muñoz Capellán.jpg
Ainsley Maitland-Niles.jpg
Alan Franco.jpg
Alan Velasco.jpg
Alban Lafont.jpg
Albert Sambi Lokonga.jpg
Alberto Soro Álvarez.jpg
Alejandro Berenguer Remiro.jpg
Alejandro Gómez.jpg
Alejandro Grimaldo García.jpg
Alejandro Remiro Gargallo.jpg
Alessandro Bastoni.jpg
Alessandro Plizzari.jpg
Alessio Cragno.jpg
Alessio Riccardi.jpg
Alessio Romagnoli.jpg
Alex Král.jpg
Alex Meret.jpg
Alex Nicolao Telles.jpg
Alex Oxlade-Chamberlain.jpg
Alex Sandro Lobo Silva.jpg
Alexander Bernabei.jpg
Alexander Isak.jpg
Alexander Nübel.jpg
Alexandr Golovin.jpg
Alexandr Maksimenko.jpg
Alexandre Lacazette.jpg
Alexey Miranchuk.jpg
Alexis Mac Allister.jpg
Alfredo Morelos.jpg
Alhassan Yusuf.jpg
Ali Karimi.jpg
Alisson Ramses Becker.jpg
Allan Marques Loureiro.jpg
Allan Saint-Maximin.jpg
Alphonse Areola.jpg
Alphonso Davies.jpg
Álvaro Borja Morata Martín.jpg
Álvaro Odriozola Arzalluz.jpg
Alvaro Recoba.jpg
Amadou Haidara.jpg
Amath Ndiaye.jpg
Amine Gouiri.jpg
Amine Harit.jpg
Amoroso.jpg
Ander Barrenetxea.jpg
Anderson Souza Conceição.jpg
André Filipe Tavares Gomes.jpg
André Miguel Valente da Silva.jpg
André Onana.jpg
Andrea Barzagli.jpg
Andrea Belotti.jpg
Andrea Petagna.jpg
Andrea Pinamonti.jpg
Andreas Christensen.jpg
Andreas Skov Olsen.jpg
Andrei Chindriș.jpg
Andrej Kramarić.jpg
Andrew Cole.jpg
Andrew Robertson.jpg
Andriy Lunin.jpg
Andriy Shevchenko.jpg
Ángel Correa.jpg
Ángel Di María.jpg

Figure 4: players Country flag data



Aarón Martín Caricol.jpg
Aaron Ramsdale.jpg
Aaron Ramsey.jpg
Aaron Wan-Bissaka.jpg
Abdou Diallo.jpg
Abdoulaye Doucouré.jpg
Abdülkadir Ömür.jpg
Abel Ruiz Ortega.jpg
Achraf Hakimi.jpg
Adam Hložek.jpg
Adam Idah.jpg
Adama Traoré Diarra.jpg
Ademola Lookman.jpg
Adnan Januzaj.jpg
Adolfo Gaich.jpg
Adrià Giner Pedrosa.jpg
Adrian Fein.jpg
Adrien Rabiot.jpg
Agustín Urzi.jpg
Ahmed Kutucu.jpg
Aihen Muñoz Capellán.jpg
Ainsley Maitland-Niles.jpg
Alan Franco.jpg
Alan Velasco.jpg
Alban Lafont.jpg
Albert Sambi Lokonga.jpg
Alberto Soro Álvarez.jpg
Alejandro Berenguer Remiro.jpg
Alejandro Gómez.jpg
Alejandro Grimaldo García.jpg
Alejandro Remiro Gargallo.jpg
Alessandro Bastoni.jpg
Alessandro Plizzari.jpg
Alessio Cragno.jpg
Alessio Riccardi.jpg
Alessio Romagnoli.jpg
Alex Král.jpg
Alex Meret.jpg
Alex Nicolao Telles.jpg
Alex Oxlade-Chamberlain.jpg
Alex Sandro Lobo Silva.jpg
Alexander Bernabei.jpg
Alexander Isak.jpg
Alexander Nübel.jpg
Alexandr Golovin.jpg
Alexandr Maksimenko.jpg
Alexandre Lacazette.jpg
Alexey Miranchuk.jpg

Item type: JPG File
Dimensions: 30 x 30
Size: 2.06 KB

*Figure 5: players Club images data*

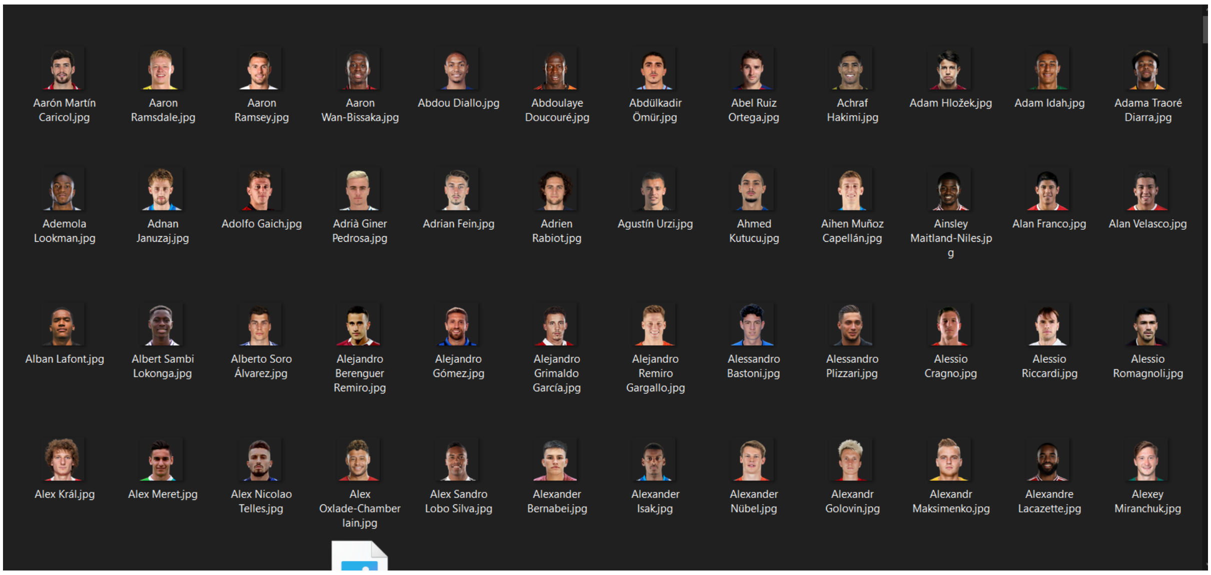
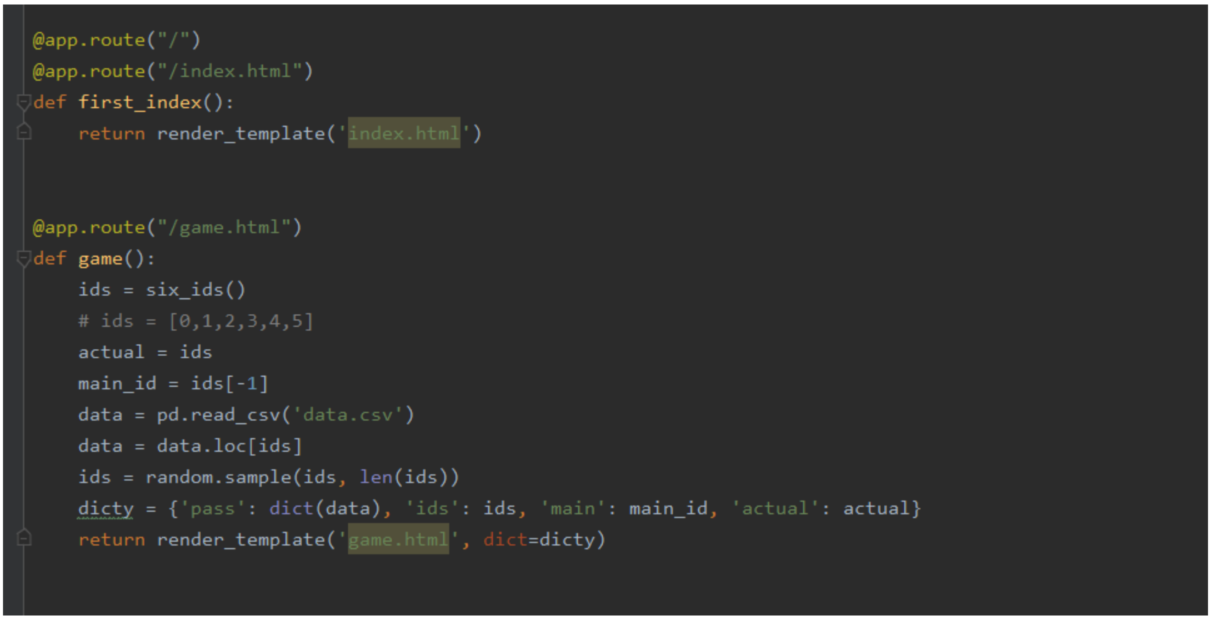
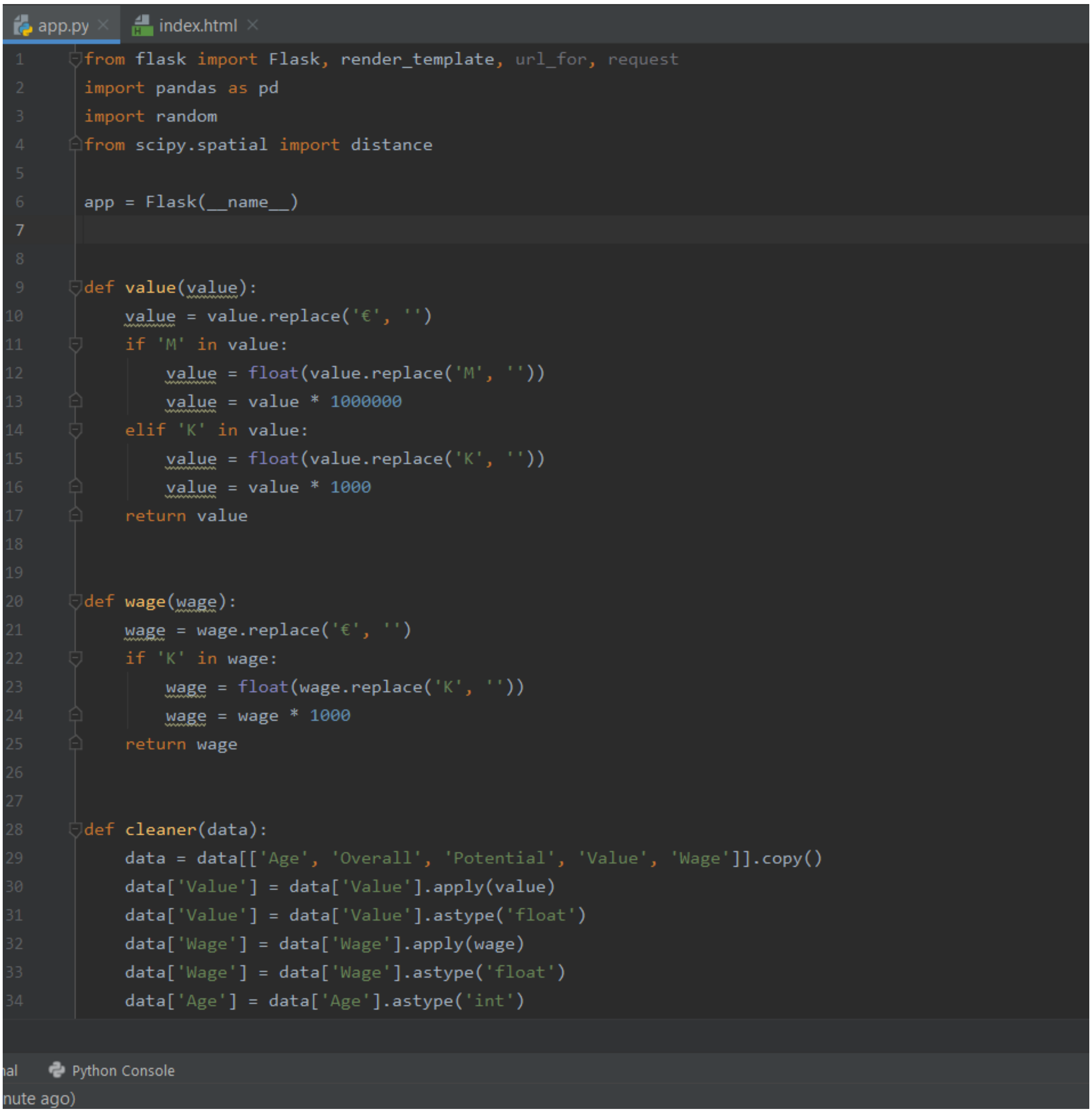


*Figure 6: players images data*



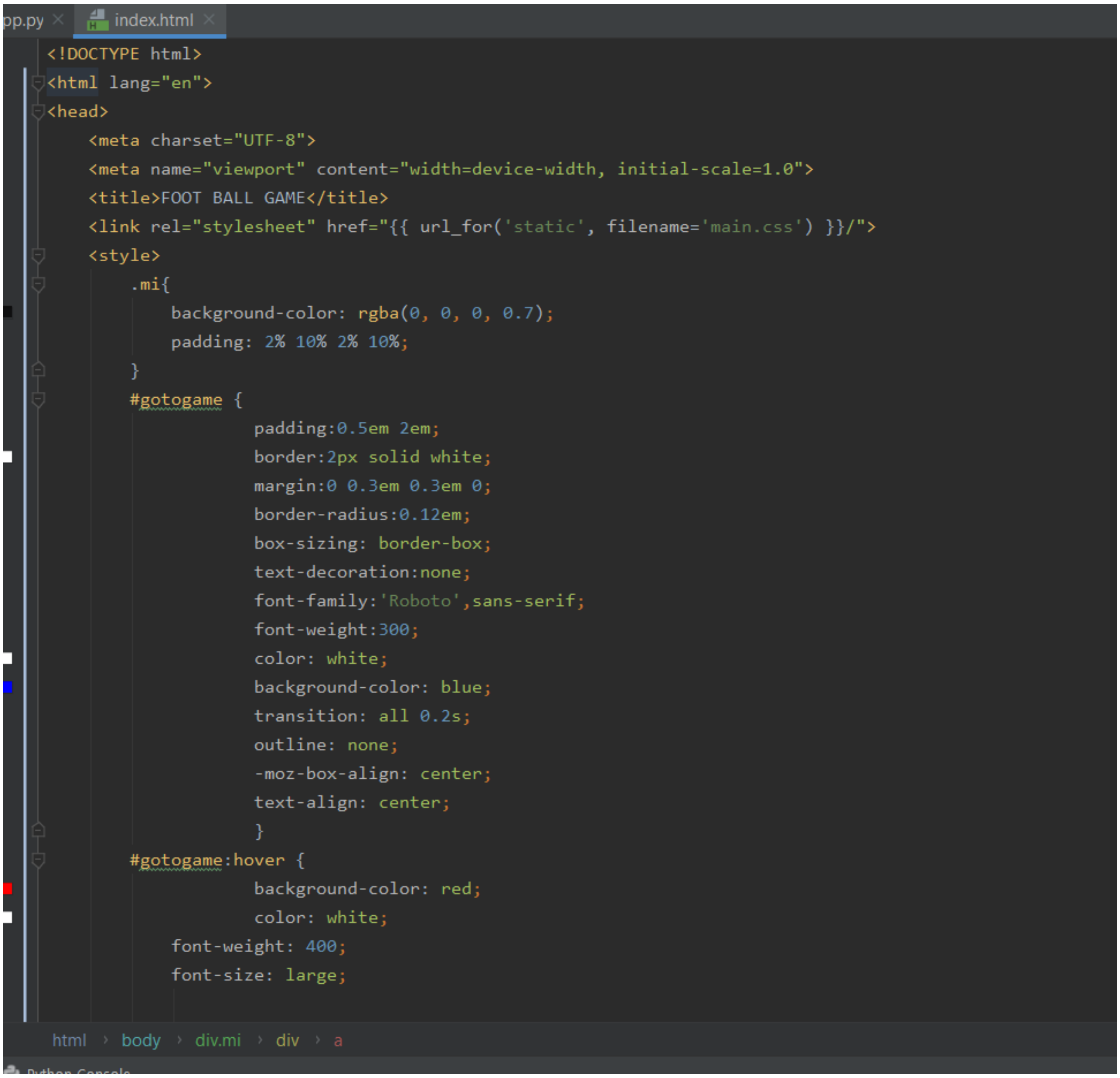
```
@app.route("/")
@app.route("/index.html")
def first_index():
    return render_template('index.html')


@app.route("/game.html")
def game():
    ids = six_ids()
    # ids = [0,1,2,3,4,5]
    actual = ids
    main_id = ids[-1]
    data = pd.read_csv('data.csv')
    data = data.loc[ids]
    ids = random.sample(ids, len(ids))
    dicty = {'pass': dict(data), 'ids': ids, 'main': main_id, 'actual': actual}
    return render_template('game.html', dict=dicty)
```

Figure 7: *app Routing*



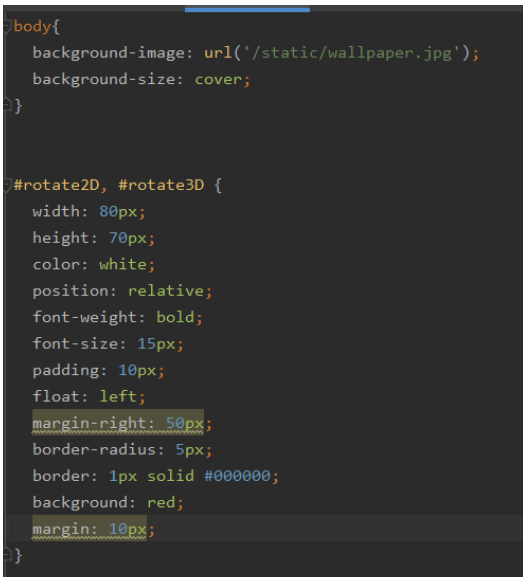
```python
from flask import Flask, render_template, url_for, request
import pandas as pd
import random
from scipy.spatial import distance

app = Flask(__name__)


def value(value):
    value = value.replace('€', '')
    if 'M' in value:
        value = float(value.replace('M', ''))
        value = value * 1000000
    elif 'K' in value:
        value = float(value.replace('K', ''))
        value = value * 1000
    return value


def wage(wage):
    wage = wage.replace('€', '')
    if 'K' in wage:
        wage = float(wage.replace('K', ''))
        wage = wage * 1000
    return wage


def cleaner(data):
    data = data[['Age', 'Overall', 'Potential', 'Value', 'Wage']].copy()
    data['Value'] = data['Value'].apply(value)
    data['Value'] = data['Value'].astype('float')
    data['Wage'] = data['Wage'].apply(wage)
    data['Wage'] = data['Wage'].astype('float')
    data['Age'] = data['Age'].astype('int')
```

*Figure 8: app.py codes*



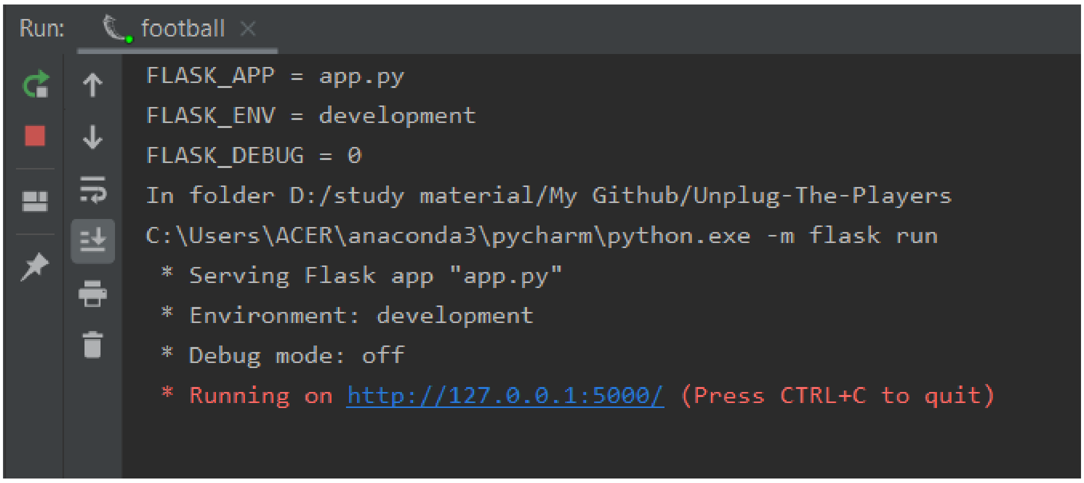
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>FOOT BALL GAME</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='main.css') }}/">
    <style>
        .mi{
            background-color: rgba(0, 0, 0, 0.7);
            padding: 2% 10% 2% 10%;
        }
        #gotogame {
                    padding:0.5em 2em;
                    border:2px solid white;
                    margin:0 0.3em 0.3em 0;
                    border-radius:0.12em;
                    box-sizing: border-box;
                    text-decoration:none;
                    font-family:'Roboto',sans-serif;
                    font-weight:300;
                    color: white;
                    background-color: blue;
                    transition: all 0.2s;
                    outline: none;
                    -moz-box-align: center;
                    text-align: center;
                    }
        #gotogame:hover {
                    background-color: red;
                    color: white;
            font-weight: 400;
            font-size: large;
```

*Figure 9: Index.html code*



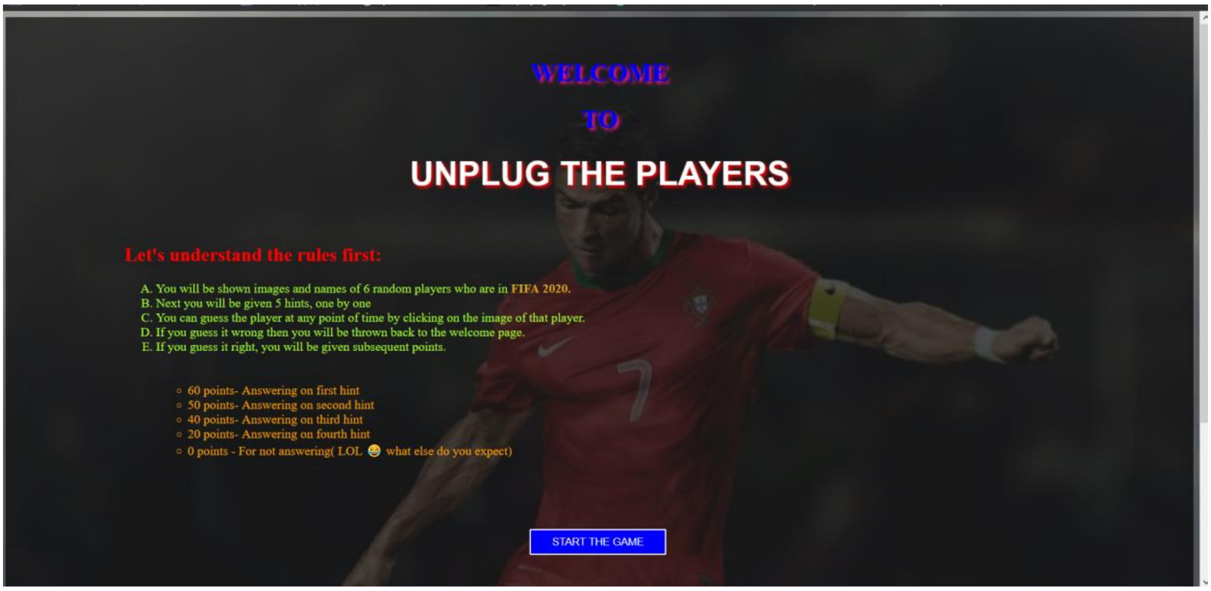
```
body{
    background-image: url('/static/wallpaper.jpg');
    background-size: cover;
}


#rotate2D, #rotate3D {
    width: 80px;
    height: 70px;
    color: white;
    position: relative;
    font-weight: bold;
    font-size: 15px;
    padding: 10px;
    float: left;
    margin-right: 50px;
    border-radius: 5px;
    border: 1px solid #000000;
    background: red;
    margin: 10px;
}
```

*Figure 10: CSS for index*



```
FLASK_APP = app.py
FLASK_ENV = development
FLASK_DEBUG = 0
In folder D:/study material/My Github/Unplug-The-Players
C:\Users\ACER\anaconda3\pycharm\python.exe -m flask run
 * Serving Flask app "app.py"
 * Environment: development
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

*Figure 11: Successful Running of web app*



WELCOME

TO

UNPLUG THE PLAYERS

Let's understand the rules first:

A. You will be shown images and names of 6 random players who are in FIFA 2020.
B. Next you will be given 5 hints, one by one
C. You can guess the player at any point of time by clicking on the image of that player.
D. If you guess it wrong then you will be thrown back to the welcome page.
E. If you guess it right, you will be given subsequent points.

- 60 points- Answering on first hint
- 50 points- Answering on second hint
- 40 points- Answering on third hint
- 20 points- Answering on fourth hint
- 0 points - For not answering( LOL 😂 what else do you expect)

*Figure 12: UI-1*

# REFERENCES

- https://www.tutorialspoint.com/flask
- https://en.wikipedia.org/wiki/Flask_(web_framework)
- https://www.scipy.org/
- https://pandas.pydata.org/