Football Team Project

Documentation

Project #1

**Name:** Hadi Charara

**Date:** 03/07/2023

**Email:** hadich10.lb@gmail.com

**Table of Contents:**

**Introduction1**

About the Program2

**Usage Guide** **4**

Command-line Interface5

Postman6

**Configuration Options…………………………………………………………………………………………………………………7**

Configuration File…………………………………………………………………………………………………………….8

Environment Variables………………………………………………………………………………………………………………9

**Code Structure…………………………………………………………………………………………………………………………10**

Overview………………………………………………………………………………………………………………………..11

Module 1…………………………………………………………………………………………………………………………………12

Module 2…………………………………………………………………………………………………………………………………13

Module 3………………………………………………………………………………………………………………………………….14

Module 4…………………………………………………………………………………………………………………………………15

Module 5………………………………………………………………………………………………………………………………….16

Module 6…………………………………………………………………………………………………………………………………..17

Module 8…………………………………………………………………………………………………………………………………18

Module 9…………………………………………………………………………………………………………………………………19

**API Documentation……………………………………………………………………………………………………………………20**

Overview………………………………………………………………………………………………………………………..21

**References………………………………………………………………………………………………………………………………….22**

External Resources…………………………………………………………………………………………………………23

1. Introduction:

The Football Club needs an application to select 10 players according to their playing position on the field and to produce various reports. The application designed in this project, enables the user to enter the following players’ details, such as their first and last names, agility and power test (APT), strength and endurance test (SET) scores, position on the field (defender, midfielder, attacker), and the national association (England, Northern Ireland, Scotland or Wales) to which they currently belong. Upon the user’s request (search, sort and select players), the program will return the appropriate response.

1. Usage Guide:

Command Line: To run the application and test the various reports, the user must use the cmd.

Postman: To test the API, the user must download Postman from the browser, and use the appropriate requests for each module.

1. Configuration Options:

Language: Python

Environment Variables: To run the program, create a virtual environment for python.

Flask: Make sure to install flask to be able to interact with the API

1. Code Structure:

**Overview:** The program is divided into 9 modules, each responsible for generating a specific report about the team/players. A separate main file is used to run the desired module. The user inputs a number from 1 to 9, and the main file will generate the appropriate response.

arrays.py: This module contains four different arrays (players, defenders, midfielders, attackers) in order to sort the players into their positions.

create\_player.py: This module contains classes and functions for managing players in a football team.

**player(object):** Initializes a player object.

**Args:**

ID (int): Unique identifier for the player.

first\_name (str): First name of the player.

last\_name (str): Last name of the player.

APT (int): Player's APT (Ability Points Total).

SET (int): Player's SET (Skill Efficiency Total).

nationality (str): Nationality of the player.

position (str): Position of the player (Defender, Midfielder, or Attacker).

**new player(function):**

Adds a new player to the players' list.

This function prompts the user to enter the details of a new player and adds them to the appropriate position-specific

list (Defenders, Midfielders, or Attackers) based on the entered position. The player's ID is automatically generated.

- The function relies on the `players`, `Defenders`, `Midfielders`, and `Attackers` lists from the `arrays` module.

- The player's position is expected to be one of the following: "Defender", "Midfielder", or "Attacker".

Raises ValueError: If the entered input does not match the expected data types or values.

average.py: Calculates the average of a player's attributes based on their first name and last name.

-This function prompts the user to enter the first name and last name of the player for whom the average needs to be calculated. It then searches for the player in the players' list and calculates their average APT and SET if found.

- The function relies on the `players` list from the `arrays` module.

- The input values for the player's first name and last name are expected to be strings.

Raises:

Value Error: If the entered input does not match the expected data types or values.

count.py: Counts the number of players in each position and prints the results.

make\_team.py: returns a team made up of 10 players. The user must enter the number of defenders, midfielders, and attackers. If there are not enough players for each position, the program will return the appropriate message.

random\_select.py: return a random set of players based on the users input. If there are not enough players, the program will return the appropriate message.

sort\_apt.py: returns a sorted list of players by their apt from highest to lowest.

highest\_apt.py: returns the player with the highest apt.

lowest\_avg.py: returns the player with the highest apt.

**main.py:** This module displays nine previously mentioned options, and the user chooses one to be displayed.

1. API Documentation:

Overview: The API code is nearly the same as the program code. In the API code, we install and setup flask to send and receive requests. In the main module we create a route for every other module and specify the request method. All request methods are ‘GET’ requests except for the create player module, we use a ‘POST’ request. We run the flask app and then use Postman to generate requests.

1. References:

[Command Line Interface — Flask Documentation (1.1.x) (palletsprojects.com)](https://flask.palletsprojects.com/en/1.1.x/cli/#environment-variables-from-dotenv)

[(211) Build a Rest API using Flask - P4- Python Flask tutorials - YouTube](https://www.youtube.com/watch?v=8L_otSDvmR0)

[Using Python Environments in Visual Studio Code](https://code.visualstudio.com/docs/python/environments)