Michael Fleming

CourseWOrk-1

Table of Contents

[Introduction to the web app 3](#_Toc465200634)

[Design Stage 3](#_Toc465200635)

[Typography 3](#_Toc465200636)

[Color Theory 3](#_Toc465200637)

[User Interface 3](#_Toc465200638)

[Flask and Jinja 4](#_Toc465200639)

[Generator Function 4](#_Toc465200640)

[Enhancements 5](#_Toc465200641)

[Critical Evaluation 5](#_Toc465200642)

[Evaluation 6](#_Toc465200643)

[Resources 6](#_Toc465200644)

[References 6](#_Toc465200645)

# 

# **Introduction to the web app**

The webapp I have designed will be used to display information about top football leagues from around the world. These leagues include the English Barclays Premier League, German Bundesliga and the Scottish Ladbrokes Premier League. The app has a leagues page where users will be able to select which of the three leagues they wish to view. Once the user has selected their specific league each team with will have a little information along with their kit badge. The information included along with the kit badge is full club name, abbreviation, Short name and squad market value.

# Design Stage

The webapp was designed with one main file coursework.py where all routes and functions would be kept. Other folders include templates, static. The static folder contains images and static files such as CSS and JavaScript. The templates folder is where all HTML files for leagues and team’s pages are kept. During the build of the webapp one of my aims was to keep a nice and tidy folder structure with images in the correct folders etc. so all files could be easily found if they had to. Keeping on top of folder management also meant I did not encounter many errors where Jinja templates did not render correctly because they could not be located.

## Typography

The main font type I plan to use on the web app is Helvetica is because it is one of the most famous and popular fonts in the world, but also very easy for users to read on the web due to its dpi (dots per inch). Compared to that off sans-serif. Although serif fonts are more formal and professional than sans-serif Helvetica I still feel it gives of a sophisticated and professional feel to the webapp. Helvetica is also a very easy font to read so users will have an enjoyable reading experience on the webapp. Due to these reasons I intend to use Helvetica as the main font choice for the webapp.

## Color Theory

The color scheme I plan to use on the website is a fairly basic one which is overshadowed by the dominant background image which stands out over it. The color scheme I used was a monochromatic scheme. This allowed me to use different shades, tones and tints of my chosen color. The color I choose was #EEEEEE. I felt using this color helped to make the webapp look professional and convey a modern webapp to users. The use of the color #EEEEEE is clear on the landing page of the webapp with a different shade used on the leagues page. The different shade used on the leagues page helps the country’s flags stand out to the user.

## User Interface

## Flask and Jinja

Flask is the micro framework for python that I choose to help build my online catalogue of football leagues. Using this framework allowed me to use routing, render templates and response objects among other things.

As well as using flask I also used Jinja 2 which is a very well-known template engine. Using Jinja 2 allowed me to parase Json data on to my HTML pages for users to see. Rather than hard coding each individual HTML page using jinja allowed me to call data from Json files which contained all the data for each league and team.



The image above shows the use of Jinja 2 on my teams.html page, t shows a mixture of HTML and Jinja 2 tags which are indicated by {{ }}. An example of Jinja being used on the page is {{ team.shortName }} this tag takes team declared at the top of the page and shortName which is the name of a variable inside one of my json files. Using Jinja 2 here saved me a lot of development time as the page was dynamically created with data inside a json file created earlier. Hard coding the page would have meant that I would have to had placed 20 thumbnails on the page, however using a for loop in jinja it did all the hard work for me, helping to save a lot of development time.

### Generator Function

The generator function is something that I added to the webapp in order to cut down on development time. Previously my function to parase data to my team’s page had been confusing and long. This function here simply loads (json. load) the Json file as data file and returns it on the teams.html page. The 1st image below shows the routes I created for all the leagues on the webapp. By returning page it loads the specific json file declared in the path and outputs onto teams.html which was declared in the generator function. I felt creating this generator function would make it much quicker to add routes for different leagues and saved me a lot of time in this part of the design. I now feel adding more leagues to the webapp in the future would be no problem at all and now requires very little code to do so. The main reason for me creating the generator function was to cut down the complexity of each team route. My thinking behind this was if I was to add dozens of more leagues to the app I did not want my file to be filled with complex routes, so I found the function very helpful.





# Enhancements

Enhancements I would make to improve the app would be to add a basic login system. My plan for this would be to ask users on the home page to sign up and login as users. One of my ideas for logged in users was that they were able to add certain leagues to their favorites so that whenever they logged in the could quickly see the leagues which they had a specific interest in. If I had more time I would have added more leagues for users to view. At present allowing users to add leagues to their favorites would not be worthwhile as there are currently only 3 leagues on the webapp. Adding the login system, I feel would give more control and customization to the user as they can see the leagues which interest them. I feel this would add much needed functionality to the webapp and I feel with more time and slightly more knowledge would have been possible.

Another feature I would have liked to have added to the webapp would have been to add the current league standings along with information about each team. I feel adding a live league table for each league on the app would add much needed content to attract users. Adding this feature to the webapp would require a web api rather than the static json files which it is using at the moment to pass data to the page. Using a web api would allow the page to dynamically update based on current standings in the league giving users the most up to date information.

Once improvement I would like to make to the app is creating is dynamically creating a more-info page for each team. This would allow users to click more info on any team they are viewing then see more detailed information about the team they select. At present there is only basic information on each team in the league I feel that adding more content about each team would enhance the webapp for users and allow them to see more.

# Critical Evaluation

# Evaluation

Building this webapp has been challenging and enjoyable but has furthered my knowledge and understanding of Python and has extended my technical experience. At the start of the coursework I had very little knowledge of Python and other tools used along the way such as Vim. Throughout the build of the webapp my basic skills of Pyhton Flask were stretched and at points was very challenging. This became very apparent when trying to parase the Json files to my HTML pages using jinja. This was by far the most challenging aspect of the build and by far the most time consuming part as well. Although this was most challenging functionality I tried to add to the app I feel it has enhanced my Python Flask and jinja skills. I feel now that I have much deeper understanding of how the code works after spending a lot of time trying to implement it. To overcome this issue, I researched examples online and looked back at examples in the workbook. After doing this I finally got the team name to parase through and realized how the code was actually working. I feel persisting with the jinja here was worthwhile as I now have a deeper understanding of how the code works.

# Resources

# References