

Advance Web Technologies Coursework 1

Darren Simpson

40284701@napier.ac.uk

Edinburgh Napier University - Advance Web Technologies (SET09103)

1 Introduction

The objective of this coursework was to demonstrate my understanding and mastery of the Python Flask micro-framework by completing a personal project in which I designed, implemented and evaluated the web application which I created on a topic of my choosing.

The web application which I chose to create was a stat tracking site for the game "Overwatch". The aim of the website was to allow users to search players by name and see their top characters played and several stats about their performance. The site itself contains a search page, sign up and log in pages, a profile page to view your own stats and a top players page which shows a list of the current top 100 players by rating.

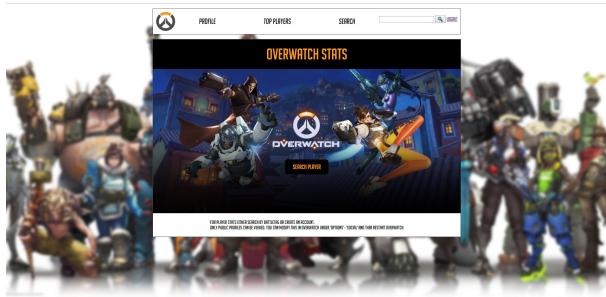


fig.1 Overwatch Stats Homepage

2 Design

The design of my web application is structured in such a way that it allows the user to either simply search the website for players from its main page, which is the simplest function, or, to allow a user to create an account and designate a profile as their own. If the user has not signed in, the navigation bar options allow either to view the top players list or to search either from the search page or from the search bar.

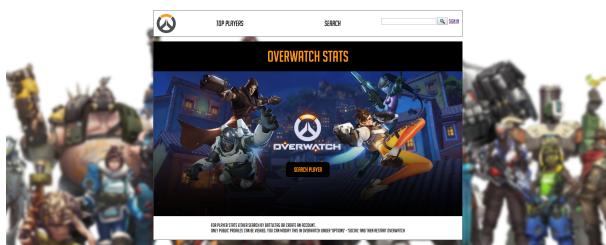


fig.2 Navigation: Logged Out

While logged in, as shown in fig.1, the user is able to use the "Profile" link to navigate directly to the profile that they have assigned either during sign up or selected afterward through

the search. The search allows the user to search either by the full username and battletag to find a specific user, or to use only the username. Using only the username will return a list of all players under this name with their avatar and current level. This can be used to distinguish your own profile or the one you are searching from others.

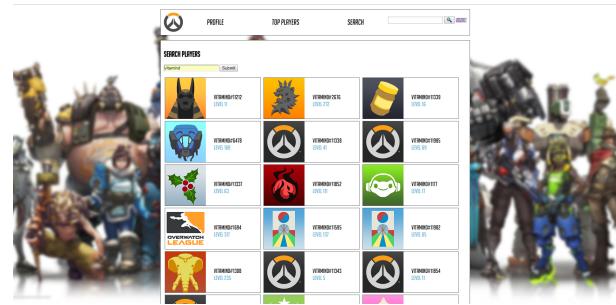


fig.3 Search Page

After navigating to a player profile, the user is given some of the same information from the search, as well as the user rating, top five played heroes, game stats, medal stats and a list of records.

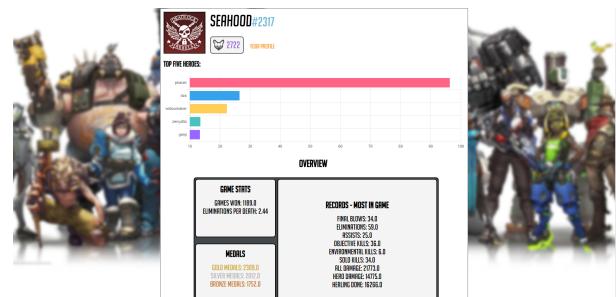


fig 4. Profile Page

From this page, the user can choose to mark this profile as their own (or as the one they wish the "Profile" link to return to), which allows the user to easily navigate to their own page from any point in the website while logged in.



fig 5. Profile Selection

The one area in which user information is entered is through the sign up page. However, no private information is required as the user decides their own username and the battletag that is entered is optional. The battletag can also be used to link to any profile so it does not necessarily link a user account to a Battlenet account. I chose this option as not to create a link between users and their Blizzard accounts, and also to allow anyone to sign up, regardless of whether they play

or not. The passwords which are used are hashed using an external library and stored in their hashed forms to avoid storing passwords in plain text. As stated before, although there is no direct link between the accounts and the profile they choose, if someone were to use the same password for this site and their Blizzard account, it could lead to security issues.

fig 6. Sign Up Page

The site is designed in a way that I felt would allow those who play the game and those who don't to navigate at different depths. People wishing to track their own stats could sign in and associate a player account with their own. Someone just wishing to browse could either search by top players or perhaps by the names of people they know or have heard of in game. The site is designed to direct predominately to a player page in most circumstances. When logging in or using the profile link you are directed to your own, as with signing up if you specify the battletag. Otherwise it is done through the search. As players as the main focus of the site, I felt that it should be easy to navigate to a player with ease.

On the Profile page itself, I would have liked to have added a list of individual heroes and the stats relating to them based on player. Once again, this proved to be an API issue as I could not get the information required. In the end I used three separate APIs to get all the information I needed for the site in its current state. I feel like the main improvement which would improve the quality of the site would be having an API which provided the stats required to bulk out the information given.

Due to the nature of the game, the player uses different heroes and as such, getting the a large amount of broad data such as "Eliminations" doesn't really show as much information about the player as I would have liked due to it being a large mostly meaningless number. Being able to filter the stats already shown by hero would also have been a great improvement as something such as "records" by hero would be interesting as opposed to overall records.

With that in mind, I would have liked to improve the site graphically. Having images of the heroes associated with the stats and being able to associate stats at a glance would have helped to give a greater amount of information per player.

I considered implementing a "compare" feature which would allow users to compare stats from their selected profile with another of their choice. I feel like this would have improved the site as it would give the information per player page more use. Being able to compare your own stats with others that you play with or some of the top players would be interesting and let you see where you are doing well or need to improve. Sadly, due to time constraints on the project, I was unable to implement this feature, though would like to in the future.

4 Critical Evaluation

While reviewing what I have created for this project, I would say that I am overall very pleased with the outcome. I think I have produced an appealing looking site that is easy to navigate and competently does everything that I set out to achieve. I feel that the overall quality aesthetically is high and the quality of the code is good, and a vast improvement on the previous project. I also separated the project into several files (such as separating the main python file from my API and Database files) which made the project more manageable and allowed me to break down what I had easier.

What works well:

I feel that the structure and layout of the site are some of its stronger attributes. It is very clear what leads where and the links between pages. It is laid out in a pleasing way and fits the theme of being "Overwatch based". The player profiles are easy to read, giving good information at a glance, while using the images from the overwatch API and sticking to a colour scheme that allows those familiar with the game to instantly pick out statistics (e.g. Purple for Competitive Rating).

The top five chart using "Chartjs" instantly allows users to see not only the top five heroes but also the differences

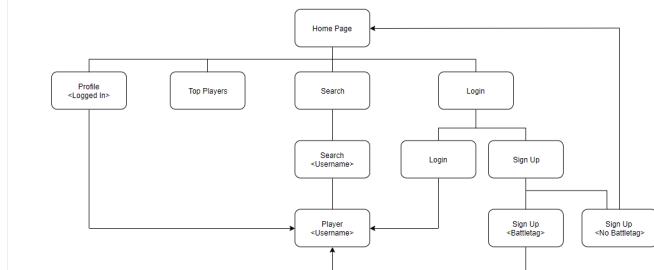


fig 7. URL Hierarchy

3 Enhancements

When looking at what is currently on the site, there are several elements I would add or improve:

I feel that the Top Players list could include links to the profiles of the Players rather than just providing information. The issue with trying to do this was that the API I was using in order to get player profiles was using only players in the 'EU' region. Most of the top 100 players are either in 'US(United States) or 'KR'(Korea). Users are still able to search for the names and those who are within the 'EU' will display so long as their profile is set to public.

in time with ease. I feel that being able to display the information in a clear and appealing way is one of the best strengths of the site in regards to it's Profiles.

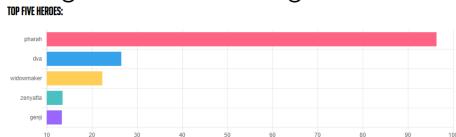


fig. 8 Top Five Chart

The site handles data well and fills each section with the relevant data without causing inconsistencies in layout. While using the search I made sure that images with broken links would have a default fallback image in order to keep the grid of player profiles uniform. Additions such as this keep the site looking good while also making the information required that little bit easier to find as it is laid out in an easy to follow manner.

The navbar, while quite simple, contains links to every section of the site that you would need, while also adjusting depending on whether or not you are logged in. While logged in it makes it that little bit easier to navigate to your own profile without searching.

What needs improvement:

While although I'm pleased with the site overall, there are a few sections that I feel could have been improved upon and perhaps did not work as well as the others.

While although I feel the information of the player profile page is presented well, I do wish that I could have filled the profiles with more relevant information. As stated before, this mostly comes down to what I could get from the API. Due to the nature of the game, having large overall stats is not particularly useful, which is why I opted to use one time stats such as "records". Being able to break the data down further and allow for sorting would have given the user a greater feel for how the player they were viewing performs.

the search function, while although functional, could have searched usernames which included parts of the search query. As it stands right now, the search will only return the usernames of players which include the full name of the account. (e.g. Searching: "Play" would only return people named "Play" rather than "Play, Player, Plays, etc".)

5 Personal Evaluation

I am proud of the work I have put in over the course of this project and of the final product that I have created. This initially felt like a very daunting task to take on as it included not just using python and Flask, but implementing other elements such as a database. Even from the very beginning of the project I found that trying to do tasks such as getting information from the API proved to be an issue as I had never dealt with one before. However, it is because of the scope of these new challenges that I feel that this project has been one of the best learning experiences I have had. Being given the option to create whatever application I wanted lead to me creating something that I have become invested in and wanted to make sure was it was made to a high standard.

This is what made me search harder to solve problems and want to go the extra mile to include more.

Using the information from the previous project and what I had learned then, I was able to expand on it and now feel confident that I have learned to learn a little better and have started to develop beyond just the basics.

Unlike the last project, I decided to not use Bootstrap in order to layout my site as I wanted to create something from the ground up myself. Having used bootstrap before however did help aid my work as while although there weren't as many tools in place, I was able to use the fundamentals I had learned and apply them to my own work.

My main downfall with this project, and what I felt I let myself down on most, was time management. Having just completed the previous project, I felt as though starting from nothing again was a massive task, and as such, this caused me to put it off by more time than I would have liked. Although I still committed a lot of time to the creation of my application, I wish I had used all of the time given to me in order to produce something even better.

Other than that, these are the biggest challenges that I faced:

Integrating Code: Having now done one project in Python, I felt more confident in using it, however, the issues I faced most was integrating different types of code with each other. Since I was using Python, HTML, CSS and also using a Database, I found that a lot of the time I was using the incorrect syntax or calling from the wrong place simply due to having such a large amount of different code. While although it might seem quite trivial, there were often times where I would spend an hour on a problem just to discover that I had used the incorrect brackets, which in itself was a learning experience. Every time this happened I would write down the problem and the syntax and keep it as a reminder to myself whenever the same issue occurred. This helped me compile a list of how to lay out what I needed to use depending on what language I was using.

Databases: Databases had been something of a weak subject for me before this project as they were something I rarely used other than for specific projects I was given. As such, when trying to use a database to allow user accounts and logging in, I often encountered problems that I was unsure of how to fix. The solution to this simply proved to be reading up on the documentation and making sure I was looking at the errors when they came back. I found that adding logging to the code meant that I could track what was being added to the database and what variables were being set to and this made solving where the issues were coming from much easier. This project has encouraged me to use logging for finding errors much more than I was previously and now see how invaluable it is.

Overall, I feel like this project has been a huge help to me. It has cemented certain practices in my mind and has shown how much I have improved in coding in regards to the first project. I feel more confident in my abilities to code and problem solve now, which I was lacking before. I think the greatest thing I have learned to utilize is compartmentalizing tasks. Prior to this project I was terrible for looking at a task and thinking about everything at once, which often lead

to me struggling to find the answer. I have learned now that breaking down bigger issues into smaller ones and doing things line at a time is the key to building up. Initially I felt like this project would be too much in the time given, but after simply breaking down the site into smaller tasks and then doing the same to those tasks, I was surprised to see how quickly it developed. Once again, being able to make something of our own choosing meant that I was more invested in this than I would have been in a brief that we were given, and I feel that is what made the difference in pushing through to solve complicated problems.

6 References

Outside of the workbook provided, my main sources used were:

[Stack Overflow](#)

[Overbuff](#)

[Play Overwatch](#)

All code in this application other than the API's is own.

API:

[Play Overwatch API](#)

[OW API](#)

[OW Ranking API - Top 100](#)