

1. System name:

Stats Checker

2. Motivation

Reasons for developing system

Sure there are other systems that can be developed. Systems like a weather app, or a shopping app, but most of these are generic ideas that are simpler to make because there are already systems out there that do something similar. Unless you're a Game Developer or something of the sorts, no one seems to take games into consideration, so why not build something in that general field.

So as to the main reason why I'm developing this system. Well for starters the gaming industry has been growing in the recent years, with more and better new games being released every year, multiplayer games becoming more popular, particularly the ones with more of a competitive situation. There are systems in place that give you a hand in games. Take league of legends for example, there are multiple systems that I know of that would recommend you how to build your character base on the most popular trend currently in game, or based on the build with the highest win rate. As for my system I wanted to do something with that general idea in mind. So I looked around, and there really wasn't a singular system that had all games statistics together in one place.

Why are stats important:

- For me personally I like to know where I stand in game. So take for instance a game like Valorant, where the better your stats the better player you are. Take something called K/D score which is an average based on the number of deaths and kills you have acquired throughout your lifetime in that game. The higher this number the better you are at the game.
- Even if you don't want to know your own stats, maybe you found a new person to play with, maybe asking them how good or bad they are at the game can seem a little rude, so you could just take his username and get his information on your own.
- Maybe it's the enemy team that you want to know about. Sometimes you think that a player you've been put up against doesn't match the same level that you are. So instead of beating yourself up about it, simply being able to see that that person has way more experience than you can be a weight off your shoulders.

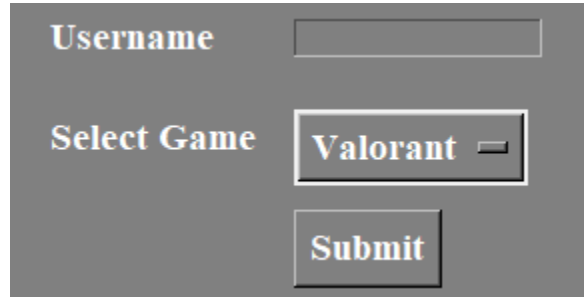
3. Purpose

To develop a system that can allow a player in any game to be able to look up his own or anyone else's statistics in said game. Since most games user's statistics are public all you would need is their username for the game of your choice. Note that not all games have this information publicly available so some games can't be added to the pool

The system condenses multiple games into one location where any user can enter their username or that of any other player to choose and get the main stats from selected game. For instance, in Valorant an important stat to determine how good a player is, is their K/D (kill to death ratio), the higher this value the better. So for someone like me who plays competitive games constantly it is nice to know what kind of teammates I have or what opponent I'm facing. This system not only makes it easier for me to find, but also allows me to choose which game I want to look up.

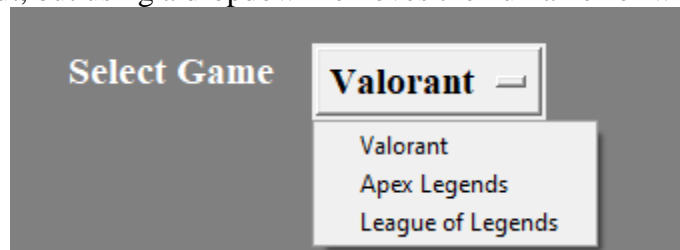
Main Functions:

- i. Simple GUI to be able to quickly and easily select the game of choice and enter the username.



A screenshot of a simple GUI with a dark gray background. It features three elements: a text input field labeled "Username" at the top, a dropdown menu labeled "Select Game" with "Valorant" selected, and a "Submit" button at the bottom.

- a.
- ii. A drop down to be able to select the game of choice, originally it was up to the user input, but using a dropdown removes the human error when entering the game name.



A screenshot showing a dropdown menu for "Select Game". The menu is open, displaying three options: "Valorant", "Apex Legends", and "League of Legends". "Valorant" is currently selected and highlighted.

- a.
- iii. A simple and understandable output that shows you the game you selected, along with the more important stats from that game. Note that different games have different stats so it won't all be the same.



A screenshot of a stats display for the game "Apex Legends". The title "Apex Legends" is at the top. Below it, there are four stats arranged in a 2x2 grid, each with a label and a value in red text.

Level	S9 Wins	Favorit Legend
346	23	Octane
Kills	S9 Kills	
876	562	

a.

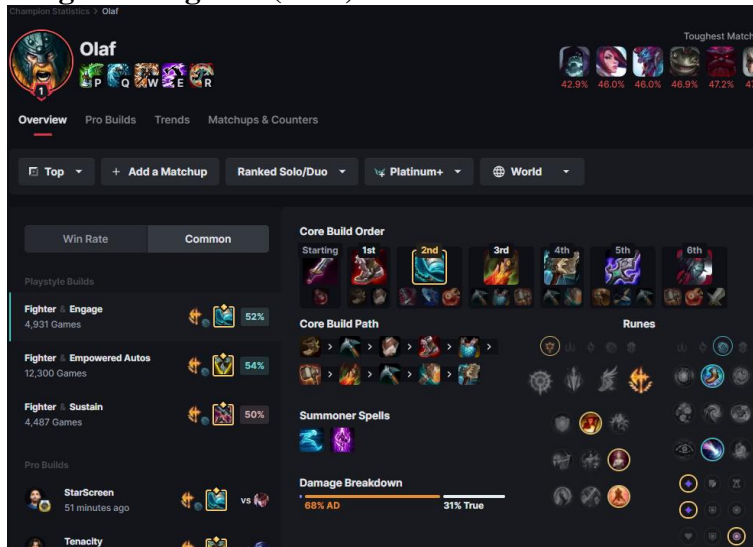
4. Relative work

There are different systems similar to mine that do other functions. For instance, I used a system (Blitz) for League of Legends that would show me the most used build, or the build with the highest win rate for the given champion I select. System like these mostly work at run time, meaning when you are actually playing the game. Others would have more features but they would be locked behind a paywall. The problem for this is that if I were to play another game, I'd have to find another application that does similar. There are some games that, to my knowledge, do not have any other application that would show the user their stats, or another players stat, sometimes this is due to the game already having an inbuilt system, but again only showing you your own.

Javier Castaneda
410721322

The biggest difference from my system to other related systems is that mine condenses multiple games into 1, instead of having to have different application for different games, or having to pay for a similar service online.

League of Legends (Blitz)



5. System description

The system is simpler than one would assume, there is still a lot more that can be done for the system in the future. To start off, there are 4 main function, 3 for the 3 games currently supported and 1 function to display the results.

The 3 main function will have a similar composition to this one, so to save time and space I will only explain one. The 3 game functions being:

```
24  #function to get the stats from valorant
25  > def valorant_stats(username): ...
78
79  #function to get the stats from Apex Legends
80  > def apex_stats(username): ...
128
129  #function to get the stats from League of Legends
130 > def league_stats(username): ...
```

To start of the 3 main python modules used to create this system are:

- Selenium
 - Selenium was used to be able to navigate to the page that has all the users stats for the selected game

```
driver.get("https://tracker.gg/")
sleep(5)
driver.find_element_by_link_text("Valorant").click()
sleep(5)
driver.find_element_by_xpath("//*[@id='app']/div[2]/div[2]/div/main/div[3]
```

-
- Once we reach the required page, the page source is taken to be used in with BeautifulSoup to get the specific data.
- BeautifulSoup
 - Once at the desired page we used BeautifulSoup to crawl through the webpage to get the information we need.

```
try:
    div = soup.find("div", {"class": "main"})

    for link in div.findAll("div", {"class": "stat align-left expandable"}):
        temp = link.find("span", {"class": "value"}).text
        stats.append(temp)

except:
    final_df = pd.DataFrame(df)
    return final_df
```

-
- The above function would get most of the main stats for that game, but for other games some stats are in different parts of the web page so I added some more lines of code to get the rest of the stats.

- Tkinter
 - Tkinter was used for the GUI of this system
 - I chose Tkinter because Tkinter is a simple GUI module with not too much detailing involved and beginner friendly
 - The GUI function is click()
 - This function is only called when the submit button is pressed in the main GUI window.
 - The main GUI function too much to explain line by line but for the most part it works similar to the following

```
#defining the label and the input
username = tk.Label(window, text="Username", background="Gray", fg="White", font=('Times New Roman', 14, 'bold'))
user_entry = tk.Entry(background="Gray", fg="White")
username.place(x=75, y=50)
user_entry.place(x=200, y=55)
```

-
- Defining a Label, or input and assigning them a position in the window block.
- The end result of the GUI looks as follows:



Stats Checker

Username: Shadow0Night0

Select Game: League of Legends

Submit

League of Legends

Overall KD	CS/Min	Win Rate
2.81	2.5	53.2%
Average K/D/A	Vision Score	
8.6 / 10.2 / 20.1	0.2	

Throughout the developing of this system many new ideas started developing, some that I was not able to work on, some that gave me too many problems attempting so I had to discard them. For instance, I wanted to let the user not be limited to just 3 games, but in doing so I would require more knowledge about webscraping that I currently have.

As for the time I spend developing this system, I spend most of my free afternoons for the past week working on it. One of my bigger problems was that tkinter was completely new to me, so I had to learn about it though out the process of this project.

As for how much of the system was developed by me, I would say entirely developed by me, but I did use a major resource for gathering the stats from different game. The rest of the system was mostly me, but whenever I ran into an error with tkinter I would turn to the internet for aid.

6. Conclusion and reflection

At the end of all the work I can say that I am happy with what it turned out to be. Although it may have a bug here or there, the system works in a way I intended it to work. There are still updates that can be added to improve the system, given more time and knowledge it would be doable. Initially the game select would've been user input, but to eliminate human error I changed it to a drop down. As for the GUI, there was none to begin with, but midway through the project I decided it would be best to make it more user friendly. Overall it was a lot more work that I imagined, with me testing ideas that turned out to be a dead end, but that was the fun part for me.

Note: I was unable to run selenium I'm headless mode, when I do it no data would be pulled from the site.