

David Church - GameSearch  
40284534@live.napier.ac.uk  
Edinburgh Napier University - Advanced Web Technologies (Set09103)

**Keywords** – GameSearch, python, html, css, js, json, bootstrap, games

## 1 Introduction

My web-app is a site, which allow users to look and sort tables of games. There are multiple ways for the user to interact with a page for example; at the landing page, it requests that the user should enter their name. Once the user hits submit it will send them to the home page.

Here they will see that it says "Hello, (user name)". From here, the user can view the three featured games, which 'GameSearch' offers. The user can then click on the title of any of these titles, which will then take them to the store from where they can buy it.

## 2 Design

**When** designing this web app I planned it out using wire-frames to see how it would look before starting development.

### 2.1 Wire-frames

When creating my wire-frames for this web-app, I thought about which colours how the layout would be used. I decided that I was going with a basic colour scheme using the bootstrap inverse colours, which are the defaults. (*Figure 1*)

For the layout, I put a bit more thought into designing. I did not want the page to that the user would be interacting with to be cluttered with information and links so I planned to make each page use white-space to clear up each page.

I planned on making a simple scroll page with bits of information but as I developed more the page was becoming non appealing to the eye. I then created another wire-frame which I choose to develop as I felt that this was the cleanest layout I could come up with for the design I was wanted to develop.

### 2.2 Navigation

On each page there are multiple links, this is my main use of navigation used throughout this web-app.

- Home
- About
- PC
- Xbox
- PS4
- Featured Games

The style for this layout was to keep it user friendly and not to over complicate things. I decided on using only a few pages which gave the user the best information. Each page uses the full navigation showing all links, I wanted to make it so the user could leave the page they are currently on and redirect to any other page.

I thought that using bootstrap navbar was the best course of action for this web page as I wanted it to be responsive. I also thought that using a list-group and add some formatting so it would show on the left side of the screen was the best way to show links that would go to the main pages. (PC, Xbox PS4) I felt that this was a more friendly and cleaner design instead of listing all the links in the navigation bar for the user to choose.

### 2.3 Page Layout

The pages which are available to the user all have the same look to keep consistency. I didn't want to change the look of each page constantly designing a new look if the one I have picked works. I was hoping that keeping each page consistent would make developing this web app a lot more easier as I am not the best at creating interfaces.

#### 2.3.1 Login

I wanted to create a basic login page but without the full aspect of logging in. I just wanted to take the user name instead of both user-name and password. I decided to use a basic but friendly looking design which is able to collect the users name and get them

to submit, if there is nothing entered it will throw an alert saying to fill their name in. (Figure 2)

### 2.3.2 Home

This page is going to be the main design layout that the user will interact with. (Figure 3) The way the navigation is set up is how it has been designed on every other page but login. From this page the user can either go to another page using the links or get redirected by clicking on the links of the 3 card tiles showing featured games. (Figure 4)

### 2.3.3 About

Following the home page design I once again kept the navigation the same. This page has a piece of information for the user to understand what this web app is about. There is also a button which once pressed opens an email input window. This is for the user to send questions or even any information that they would see that can improve the site. (Figure 5)

### 2.3.4 PC/Xbox/PS4

Each of these three pages consist of the same layout using a table which has JSON data imported. I have used a user friendly table design which the user can interact with in a few ways. Each page has the same colour layout design listing eight games for that specific system. PC (Figure 6) Xbox (Figure 7) PS4 (Figure 8)

### 2.3.5 Tables

The tables that I am using are using data from which I have created in a JSON file. I have created a JSON file for each gaming system listing eight games, the information for each game listing its name, price store. There is also an ID which is for when the user would like to sort the table. (Figure 9)

Sorting the table can be done by clicking on the headers of each section, by clicking on a header you can sort it in ascending order or descending order.

### 2.3.6 Interaction

For interaction I tried to keep it as minimal as possible so that there is not much interaction on a page that could make it feel overdone. I have used a few methods of interaction that the user can use; buttons, text boxes, links sorting tables. I feel that for what my web app is used for this is the only needed interaction that should be used.

When viewing my web page from a mobile device it should work fine but there is a few bugs. I have made the site responsive so that it should be able

to run at normal functionality at any screen size. (Figure 10)

## 3 Enhancements

**With** this coursework I feel that the restriction from using API's was a small let down as I feel my web-page would've benefited from using it. When I was researching things to add and how to fix problems when I was developing I found a few resources which would've been nice to use as instead of multiple lines of code it was able to do what I wanted in a few lines.

### 3.1 Things I would add

**Search bar** - I would like to add a working search bar so that a user could search for a game title and it would show any results with whatever they have entered.

**Contact form** - I would like to add a working PHP contact form but when I was trying to get it to work it kept having errors and as time went on I was wanting to get other things on the web-page finished so I had to stop.

**Account system** - I was looking at creating an account system that would allow a user to create an account using email password. I would like to create this so users could add favourites add reviews to games.

**Hosting** - I would like to add a hosting service so I can have my web app up and running 24/7. I started to research how to host with git hub pages and I think for my next web app I am going to look into this further. I find this interesting as there is a live server and a test server which can host python.

### 3.2 Things I would improve

**Tables** - I would like to improve the data input for my tables, right now I am using a JavaScript script which is inserting the JSON data into the table but I would like to get a working version of doing this by inserting the code using python. When I was trying to insert the code I wasn't able to get it to work and it kept showing a blank table of data.

**Broken pipe** - When I was developing this web app I kept getting a broken pipe error, I would like to be able to stop this or manage to be able to stop it showing so actively.

**Contact form** - Just like in things I would like to add I want a fully working contact form which can contact an email service which can then send automated replies back.

## 4 Critical evaluation

I feel that throughout this web app there are things that could be improved if I had spent maybe more time researching features but at the same time I also feel that for what I have been asked to develop I have done well.

### 4.1 Things that work well

**Login** - I wanted this page to just collect the users name so when they entered the home page they were welcomed. I feel that this accomplished this with the login page that I have created.

**Tables** - Although I was hoping to get the tables to work via python instead of using JavaScript I feel that for the purpose I have used them for they work well. They allow the user to sort either ID, name, price or store.

**Navigation** - I feel that the navigation that I have created within this web app works clearly. I have made it clear where navigation is and which page the user navigates to.

### 4.2 Things that worked poorly

**Contact form** - I wanted to create a form, which used PHP, but I was unable to get it to work. I did however; end up using a simple bootstrap button, which once pressed, opened an email window for the user to use.

I feel that if I got the PHP to work correctly I could have made a fully accessible contact form that asks the user questions could be more in-depth than an email.

**Table JSON data** - When developing the code to get this to work, I originally planned to import the JSON data in the python file but it kept sending out a blank table. I decided that I would use code from which I used in a web app I have already designed to import the data. I was also hoping to create a hyperlink on the "store" data so that once clicked it will take you to the store.

I hope that if this comes up in the next coursework I can will be able to import JSON data using python as I feel that it would be a lot more cleaner code

than pieces of JavaScript inserted into my HTML files.

**Broken pipe** - When developing I kept running into a broken pipe error. I did some research about it and I think I found some solutions to fix it but sadly, I did not get the time to try to implement them. I am hoping I can find a fix so when I start the next web app I can have a way of stopping this error from appearing. (Requests, 2017)

## 5 Personal evaluation

**Whilst** developing this web app its not lie that I came across things I was not able to fix, but there were times where after a little bit of research I was able to change.

### 5.1 Things I could not fix

**These** few things are what cause me most problems whilst developing. I spent most of my time trying to fix these three issues but for some reason and I still haven't found out why they keep appearing.

#### 5.1.1 Broken pipe

As stated in the above section, I do think I have found a solution to this issue but I have not yet tested it. I feel that this challenge can be a really good learning experience as it looks at using Web Server Gateway Interface (WSGI) server capable of handling concurrent requests.

I feel that if I put a little bit of time into learning how to operate this I could fix the broken pipe issue I kept experiencing during development.

#### 5.1.2 JSON data input through python

When the use of JSON files became into play, I first tried to import the data into an HTML table using code in the .py file. I found that it threw no errors but it was always spitting out a blank table with eight rows. I struggled to get this to work for about two days but everything I found on getting it to work still kept throwing out blank data.

I then thought that it might be something to do with my JSON data so I tried a validator but it turned out that it was fine. Eventually I gave up because I wanted to continue so I got a piece of code which I have used before in one of my past web apps which involved using JSON data. I then imported this through a JavaScript script, which did end up working.

Then came the issue of getting the table to sort when the user clicked on one of the headers. I kept getting an issue with the table ID, as the script that I have used was not reading it. This took sometime but I eventually found which was causing the error and it was a missing curly bracket. Once I fixed this issue, my tables of JSON data worked and were sort-able when a header is clicked.

### 5.1.3 Contact form using PHP

When creating the contact/about page I originally wanted a contact form, which the user would enter data in to, but as, I started to develop it I wanted to go further. This caused problems as I tried to get PHP to work but I kept getting errors.

I then began to find work arounds but I ended up just making a simple contact button, which once clicked opened an email window allowing the user to enter an email and send to the email address which is set up. (Figure 11)

## 5.2 Things I could fix

The two things that did break when I was developing was some simple fixes with some minimal errors in the code.

### 5.2.1 Table sorting

I came across the issue where the table was not sorting when the user clicked on one of the headers. I kept getting an issue with the table ID, as the script that I have used was not reading it.

This took sometime but I eventually found which was causing the error and it was a missing curly bracket. Once I fixed this issue, my tables of JSON data worked and were sort-able when a header is clicked.

### 5.2.2 Method GET POST

When creating my log in page I had an issue trying to get the users name to show on a different HTML file. After looking at related problems I finally found a way to make it work.

The main problem I had when trying to get it to work was that I didn't have `name=request.form["name"]` after the `render template`. This did take a few try and errors but I feel now that I have a grasp understanding on how it works.

## 5.3 Overall performance

I fell that throughout this web app I have completed the coursework to the best of my abilities. I have learned a few things a long the way which I hope I can apply in to the next coursework I have learned

about some errors which can be easily fixed some new plugins when I was researching for fixes.

There were parts where I could have put more effort into like getting the JSON data to import through python instead of a script but I also feel that with the deadline closing in I didn't want to spend all my time trying to fix one thing.

## 6 References

- 1 Requests, F. (2017). Flask broken pipe with requests. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/12591760/flask-broken-pipe-with-requests> [Accessed 20 Oct. 2017].

## 7 Appendix



Figure 1: **Inverse navbar** - Inverse colour scheme using bootstrap.

Figure 2: **Login page** - Landing page to get user name.

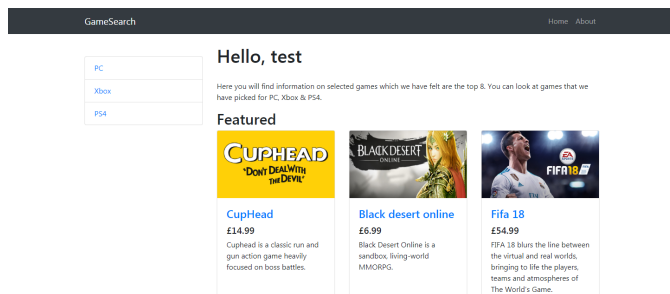


Figure 3: **Home page** - Main page which the user can interact with.

## Featured

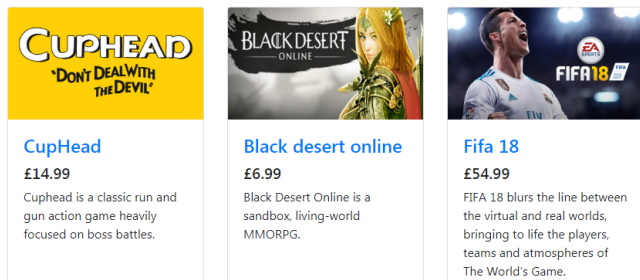


Figure 4: **Featured card-tiles** - Card-tiles which the user can interact with.

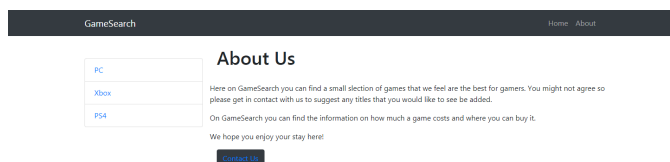


Figure 5: **About page** - Showing the information and contact option.

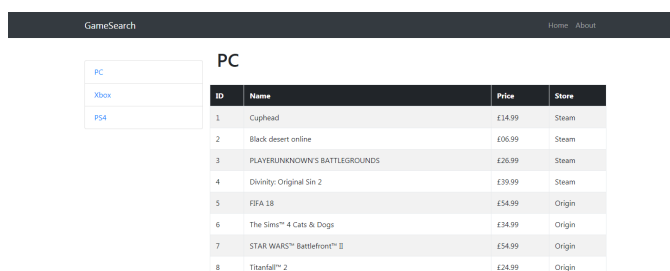


Figure 6: **PC page** - Showing the table data.

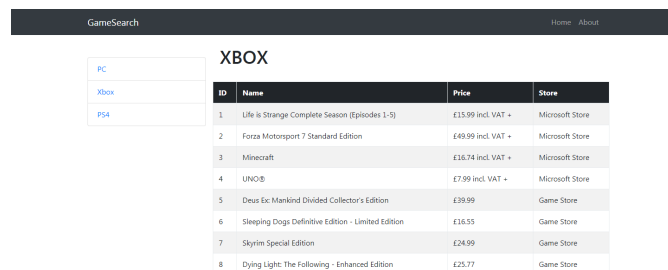


Figure 7: **Xbox page** - Showing the table data.

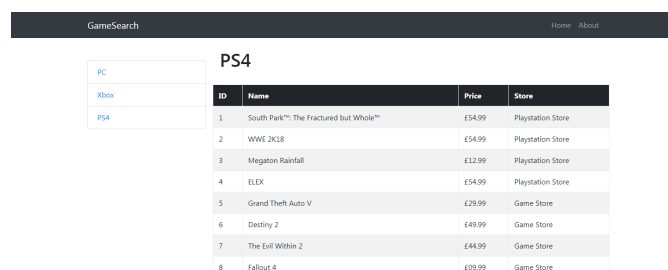


Figure 8: **PS4 page** - Showing the table data.

```

1  [
2  {
3      "id": 1,
4      "name": "Cuphead",
5      "price": "£14.99",
6      "store": "Steam"
7  },
8  {
9      "id": 2,
10     "name": "Black desert online",
11     "price": "£06.99",
12     "store": "Steam"
13 },
14 {
15     "id": 3,
16     "name": "PLAYERUNKNOWN'S BATTLEGROUNDS",
17     "price": "£26.99",
18     "store": "Steam"
19 },
20 {
21     "id": 4,
22     "name": "Divinity: Original sin 2",
23     "price": "£39.99",
24     "store": "Steam"
25 },
26 {
27     "id": 5,
28     "name": "FIFA 18",
29     "price": "£54.99",
30     "store": "Origin"
31 },
32 {
33     "id": 6,
34     "name": "The Sims™ 4 Cats & Dogs",
35     "price": "£34.99",
36     "store": "Origin"
37 },
38 {
39     "id": 7,
40     "name": "STAR WARS™ Battlefront™ II",
41     "price": "£54.99",
42     "store": "Origin"
43 },
44 {
45     "id": 8,
46     "name": "Titanfall™ 2",
47     "price": "£24.99",
48     "store": "Origin"
49 }
50 ]

```

Figure 9: PC json - Example json data.

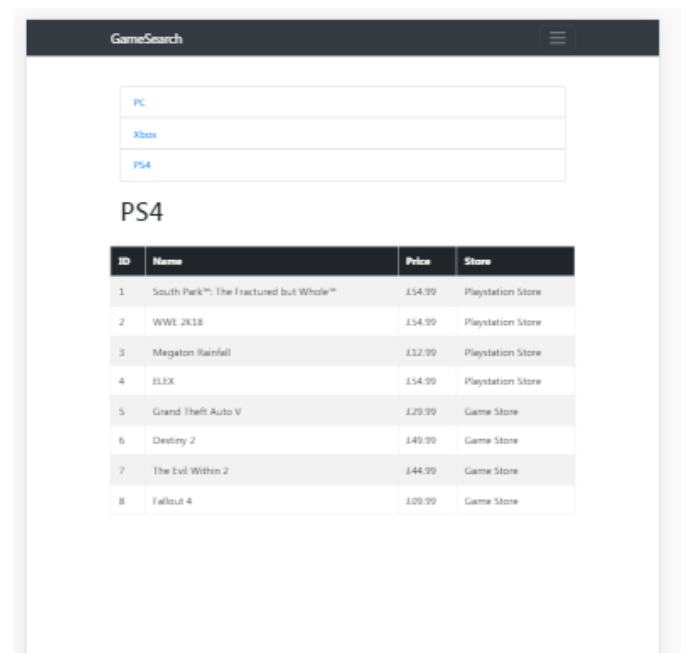


Figure 10: iPhone 6 - Screen from an iPhone 6.

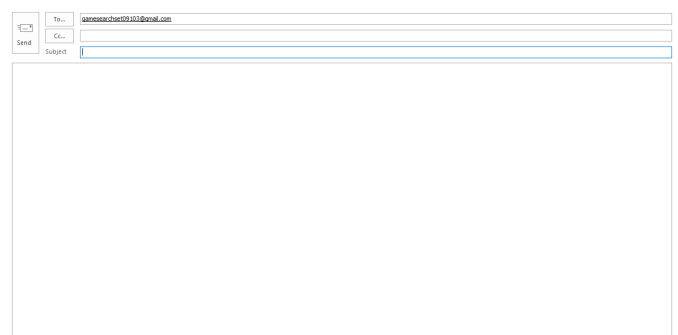


Figure 11: Email screen - What the user sees when they click contact.