National College of Ireland

Software Project

Business Information Systems

2020/2021

Sam Greenan

X17449342

X17449342@student.ncirl.ie

Sole Seeker

Technical Report

Contents

Executive Summary

Sole Seeker is a community based web application designed for users who are interested in the sneaker industry. This web application encompasses a wide range of services that would attract sneaker enthusiasts. Our three main functionalities include a REST API sneaker search function to help users find their perfect pair, a Scavenger Hunt Giveaway Competition utilising a Maps API and a forum page that allows users to express their opinions and listen to others. The application also includes a secure Login and Registration system developed with Mongo DB, accompanied with encryption of personal information. My application also explores the possible revenue streams that could be facilitated with my website. As a Business Information Systems student, I wanted to focus on this area in depth. Areas such as PayPal Integration, Google AdSense and sponsorships were the focus point of this aspect. The Sole Seeker application was made using a wide range of technologies. One most prominent is Flask, a micro framework that utilises the backend language Python. This partnership of framework and language has grown immensely in the area of web development over the last few years.

1.0 Introduction

1.1. Background

Sole Seeker is a community based web application developed for the growing popularity of designer sneaker enthusiasts. This project was created as a hub for enthusiasts to connect and utilize valuable services. These services would range from giveaway activities, forum communication and services to find the newest pair of sneakers for popular brands through the manipulation of user specification. By utilizing a wide range of web API's, different functions and a secure storage system, these services could be created.

I decided to pursue this topic due to my massive interest in the sneaker industry. I always keep up to date with all sneaker-related news and trends. Through this interest I noticed a gap in the market for this type of application. Over the years, the sneaker community has grown rapidly with the introduction of new exciting brands and shoe types.

After deciding on my project topic, I began researching existing applications surrounding my area of interest. Websites, such as Reddit and Twitter have good instances of bringing like-minded people together, I will take this into account when developing my application, in particular my forum service. Other websites such as Shopstyle.com and TG4 solutions on SwaggerHub.com are good examples of proving users with thousands of different pairs of sneakers. These examples relate to the service of finding the newest pair of sneakers for popular brands. Through this research I found that TG4 Solutions provide an online Sneaker database API to developers. I noted this as it could replace my original idea of manual database creation or web scraping. This would in turn save time and increase efficiency through non manual creation and updates.

I knew straight away that this idea can be commercialised. Pursuing a degree in Business Information Systems I felt as though this is quite an important factor of my project. I want to delve deep into the marketing side of my application. Through research into different streams of revenue for this type of application, it led me to look into PayPal ecommerce, sponsorships and Google AdSense.

1.2. Aims

In short, I want to create a well-designed, secure, and adaptive web application for all types of internet users. With the utilization of my chosen technologies, I hope to achieve this goal.

I aim to use the necessary measures relating to information security that most web applications nowadays cater for. HTTPS will be incorporated into my final deployed website. This will protect user information and destruction of services. Through encryption of personal user information moving from my application to my database, this will ensure that the data is kept confidential. For example, I aim to encrypt all user passwords as soon as they are submitted.

I also want to explore the business side of my idea. Being a Business Information Systems student, I feel as though this is a very important aspect. The market I hope to target is Male and Female, aged 12 – 35 years old. The sneaker market is massive and is a great area to venture into. I hope to gain traction and attention from my project idea. I want to learn about how an application like this can gain revenue. In this case, it can be through Google Ad sense, PayPal integration, sponsorships or even selling the idea.

I aim to use a wide variety of API's to bring a range of different functionality into my web application. These API's can facilitate my services, such as a Maps API, PayPal API and shoe repository database API.

Main Objectives to be completed:

- Build a user-friendly web application.
- Create a secure Login/Registration system that includes encryptions of the data.
- Retrieve and store relevant user information.
- Create a Giveaway function for my users with the utilization of a Map API.
- Create a program to compare users input with a database of sneakers.
- Develop a forum page to bring the sneaker community together to discuss recent news or popular trends.
- Build a detailed and secure database to hold important user information.
- Import a Twitter API for Sneaker news/updates.
- Create a promotional video for the homepage to discuss the main functions of the application.
- Input a donation link using the PayPal API for commercial use.
- Manipulate and Integrate a Sneaker REST API for sneaker results.

1.3. Technology

As soon as I solidified my project idea, I started to think of the different technologies I could use. I knew from the start that I wanted to use the language Python. I never used this language before in any college module; however, I noticed its growing popularity in today's industry. I started learning the language using a Udemy course the summer before final year commenced. Through building mini projects and programs, I started to really enjoy using the language.

With my idea and language of preference in mind, I had to brainstorm the type of application I would create. Through extensive brainstorming into the types of applications I enjoy developing and the suitability with my project idea, I landed on building a web application with the framework Flask. I chose this as I have a lot of experience in web development from different modules throughout my college career. For example, ASP.net development and AWS development. I also feel that the types of services I want to provide within my application are better showcased on a desktop computer rather than a smartphone.

I will now go more into detail on my chosen technologies.

Front end

- HTML, CSS, BOOTSTRAP

In order to create a diverse and user-friendly experience, I must cater for the front end of my web application. In order to create the templates for each web page, HTML and CSS will be used to showcase them. I'll also be using BOOTSTRAP for my base template that will be inherited across all web pages for synergy.

Backend

- Python, JSON, jQuery, JavaScript

To create my specific functions, I require a large amount of back-end functionality. Python is the main language used in the framework Flask. It is a well-known, extremely powerful language. I will also be using JavaScript for perform some functions in my web application. This ties well with the front-end language HTML. To facilitate my REST API, I will need to use JSON. Lastly, for form validation I will require jQuery.

Frameworks

- Flask

Flask is a micro framework used in web application services. It utilizes the back-end language Python. I chose this framework as it will work well with all of my chosen functions. It also has a large community for development, which is quite useful for solving Flask issues. The other framework I considered is Django, this framework also utilizes the backend language Python. However, Django would have taken much longer to master and contains features that my application deems unnecessary.

Other

- Visual Studio Code, Mongo DB, TG4 Sneaker REST API, Heroku, PayPal API, Twitter API

I will develop my project in Visual Studio Code. It is a free code editor that caters for a wide range of frameworks and languages. To hold relevant user information, I will use MongoDB as my database storage system. MongoDB is a document-oriented database storage service. MongoDB's cloud platform ATLAS is extremely useful for creating database clusters. I'll connect my application through the 'MongoClient' package in Flask. Having a cloud supported database ensured high security in maintaining important user information. Also, the online MongoDB interface is extremely user friendly. I can easily create separate clusters for different parts of my application.

To obtain an online repository of sneakers I will use the REST API Sneaker database by TG4 Solutions. For deployment of my website, I will used Heroku. This is a free web hosting service. This service is extremely useful and uncomplicated to update my site when I need to.

For my home page, I will implement the Twitter API for embedded tweets and a tweet function. The embedded tweets will preview tweets coming from large sneaker news pages. The tweet function transfers the user to Twitter, drafting a tweet with the hashtag #SoleSeeker tagged. Also within the homepage will be a donation link provided by the PayPal API. This is an example of a revenue stream that my application can provide.

1.4. Structure

My report is structured as follows:

- Introduction

This section highlights the background, aims and technologies used in my final year project. It gives a detailed run through of where I came up with the idea for Sole Seeker, what I want to achieve in this project and the technologies that could facilitate this.

- System

This section looks at the many different types of requirements that Sole Seeker needs. Firstly, I will discuss each one of my functional requirements, detailing the importance of each one. After that I will show my overall Use case along with all use case scenarios. Each use case scenario includes a description and priority in relation to my overall application. Furthermore, I will highlight my other requirements relating to data, user, environmental, usability, security and non-functionality. After discussing my requirements, the report showcases the design and architecture, describing the system architecture and the components used to develop the Sole Seeker application. The report then moves onto describe the implementation of my main algorithms, functions and methods. I have included snippets of code where appropriate to describes these processes. Following on from implementation are screenshots of the Graphical User Interface with a description accompanied by them. Lastly the report discusses the testing and evaluation carried out in my project.

- Conclusion

This section describes the advantages and disadvantages of my overall application, including the limitations that were present.

- Further development or Research

This section describes what improvements will be seen in the future and the opportunities to grow Sole Seeker.

2.0 System

2.1. Requirements

2.1.1. Functional Requirements

This section focuses on the functional requirements of my application. I will list out the necessary function the system must accomplish. These functional requirements clearly define what is required by the system to let the user interact with the services within the application.

1. Login/Registration function
   - The system should allow the user to either create an account or log in when entering the website. A user shall be able to enter their name, email and password when creating an account. A user shall be able to enter their email and password when logging into their account.

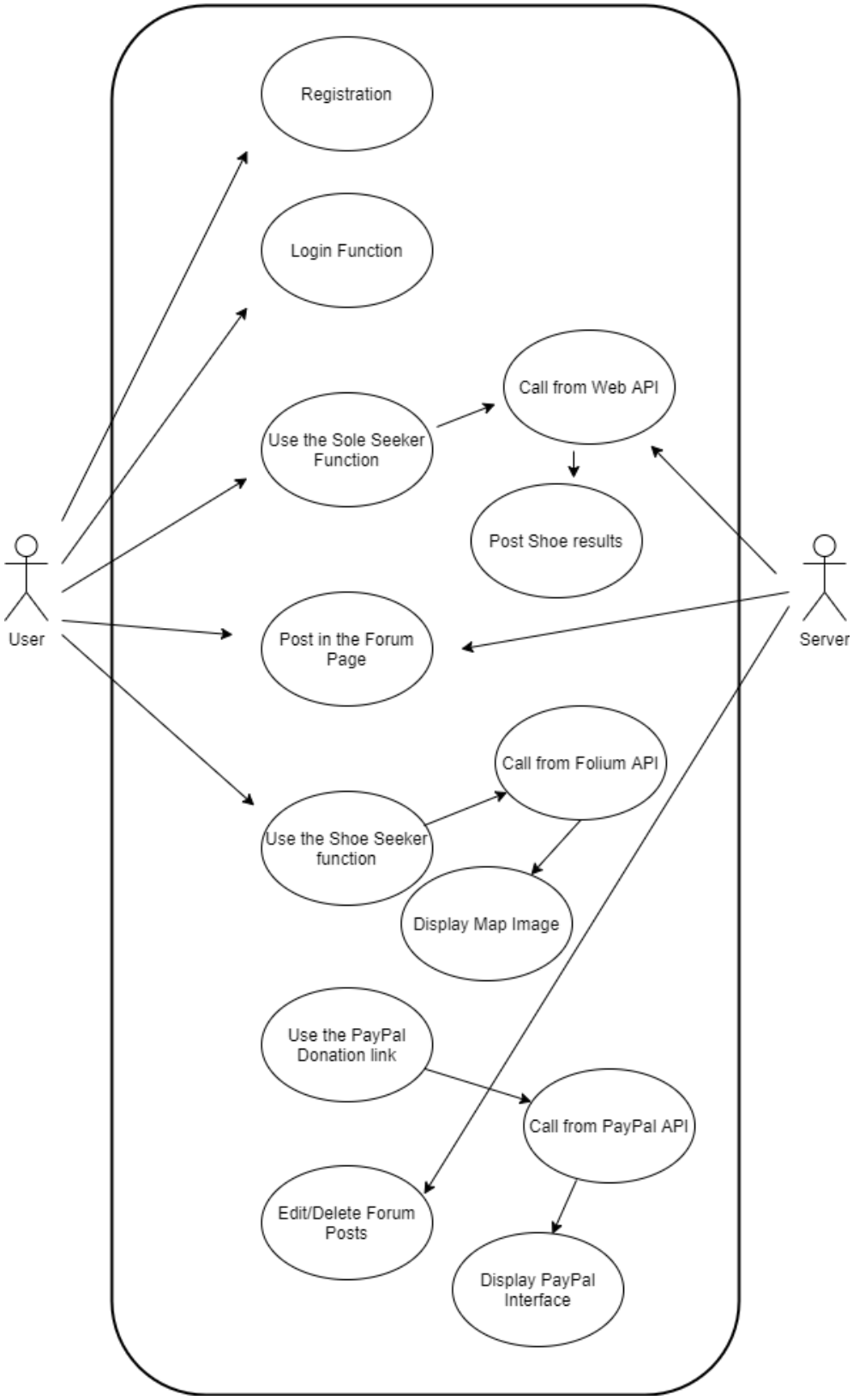
2. Sole Searcher function
   - The system should allow the user to select certain criteria relating to a pair of sneakers. Once submitted by the user, the system shall present at least 50 sneaker results for the user to browse through. The system shall provide relative sneaker information alongside each result given to the user.

3. Shoe Seeker(Giveaway function)
   - The system shall provide steps to follow in order to take part in the giveaway competition. The steps should inform the user about the prize of the competition and the location at where the prize is hidden. The system shall provide a map function to help the user locate the prize. The system shall also notify on the web page, the company in which supplied the prize.

4. Forum function
   - The system shall allow the user to make posts on the forum page once they are signed in. The system shall showcase all posts made by all users on the forum page.

5. PayPal function
   - The system shall allow users to access a link to sign into their PayPal account and donate to the Sole Seeker application using a PayPal button. The current function is sandbox with a dummy sender and receiver account.

2.1.2 Use Case Diagram



2.1.2.1 Requirement 1 <**User Registration and Login**>

Description & Priority

This allows the actor to register their email to my website. Priority is high. A user must register to enter the Sole Seeker web application.

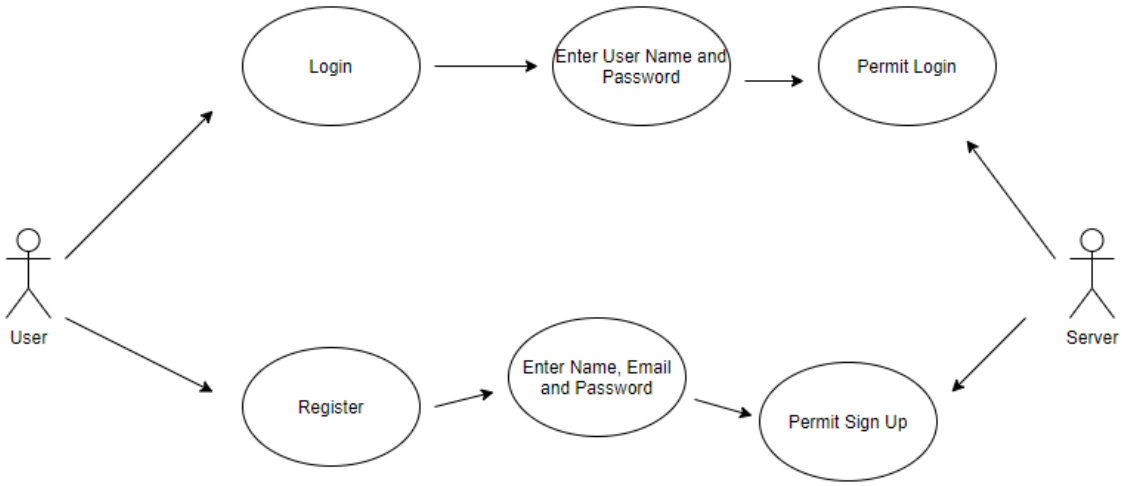
Scope

The scope of this use case is to register and login a user.

Description

This use case describes the action of the user registering or logging into the website. The user would enter a name, email, and password.

Use Case Diagram



Flow Description

Precondition

The system is active.

Activation

This use case starts when a user enters the Registration/Login page of the website.

Main flow

1. The user enters the Login and Registration page.
2. The system loads the Login and Registration page.
3. The user enters their personal details to create an account.
4. The system accepts the user details and stores them into a database.
5. The user account is created.
6. The user is navigated to the User Information page.

Alternate flow

A1: <User Logs into the system>

1. The user enters the Login and Registration page.
2. The system loads the Login and Registration page.
3. The user enters their personal details to log into their account.
4. The system searches the database to locate the user's login details.
5. The system accepts the user Login details.
6. The user is logged in.
7. The user is navigated to the User Information page.

Exceptional flow

E1<User enters invalid information - Registration>

1. The user enters the Login and Registration page.
2. The system loads the Login and Registration page.
3. The user enters their personal details into the Registration fields.
4. The system refuses the user details.
5. The user is rejected to enter the site.
6. The system displays the relevant error message.

E2<User enters invalid information - Login>

1. The user enters the Login and Registration page
2. The system loads the Login and Registration page
3. The user enters their personal details into the Login fields.
4. The system refuses the user details.
5. The user is rejected to enter the site.
6. The system displays the relevant error message.

E3: <User already has an account>

1. The user enters the Login and Registration page.
2. The system loads the Login and Registration page.
3. The user enters their personal details into the Registration fields.
4. The system refuses the user details.
5. The user is rejected to enter the site.
6. The system displays the 'already account registered' error message.

Termination

The system presents the User Information page with a user logged in.

Post condition

The system returns to the User Information page

2.1.2.2 Requirement 2 <**Sole Searcher**>

Description & Priority

This allows the actor to gain an output from the system. The user will input criteria relating to a shoe, e.g., brand, gender and the system will give the user sneakers on the market that relates to them specifics. Priority is high, as this is a large function within my website.

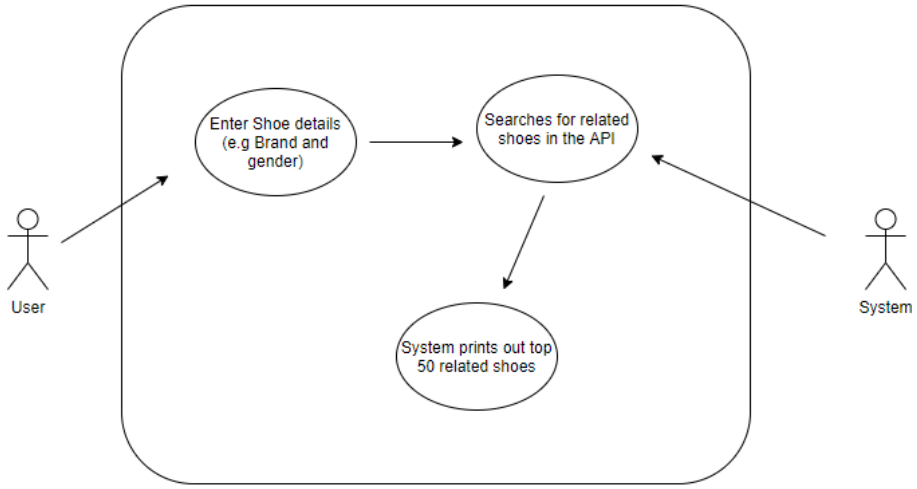
Scope

The scope of this use case is to provide the user with their specified shoe output.

Description

This use case describes the function Sole Searcher. It shows how the user interacts with the function to gather an output.

Use Case Diagram



Flow Description

Precondition

The system is active.

Activation

This use case starts when a user enters the Sole Searcher page of the website.

Main flow

1. The user enters the Sole Searcher page.
2. The system loads the Sole Searcher page.

3. The user enters necessary details on Shoe specifics (e.g., brand and gender).
4. The system searches through the shoe API.
5. The system displays 50 separate outputs of sneakers.

Exceptional flow

E1: < User fails to enter a required field >

1. The user enters the Sole Searcher page
2. The system loads the Sole Searcher page
3. The user does not specify all requirements.
4. The page displays the error message ‘field not entered’.

Termination

The system presents 50 separate outputs to the user.

Post condition

The system goes into a wait state. Available to run the function again.

2.1.2.3 Requirement 3 <**Shoe Seeker**>

Description & Priority

This allows the actor to utilize the features on the Shoe Seeker web page. Priority for this function is medium, as it is not the main function.

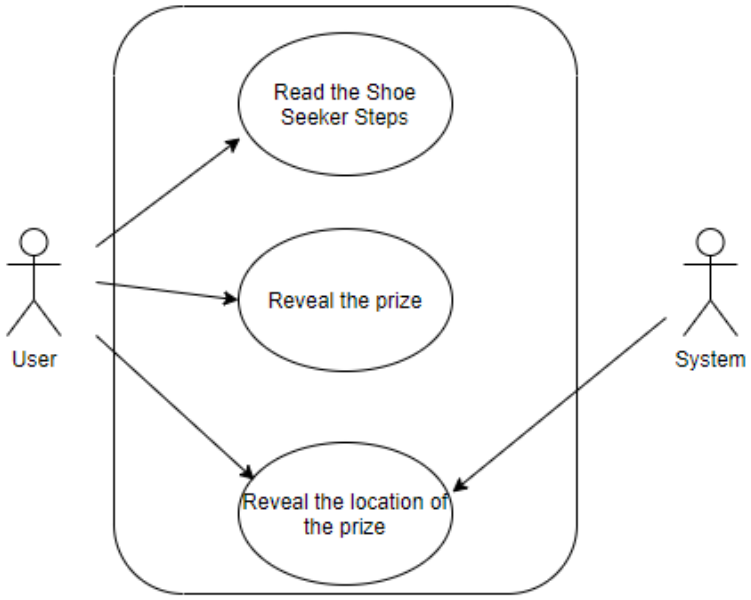
Scope

The scope of this use case is to use the Shoe Seeker Giveaway function

Description

This use case describes the function reading the Shoe Seeker steps, revealing the giveaway prize then showcasing the location of the prize.

Use Case Diagram



Flow Description

Precondition

The system is active.

Activation

This use case starts when a user enters the Shoe Seeker page of the website**.**

Main flow

1. The user enters the Shoe Seeker page.
2. The system loads the Review page.
3. The user reads the steps of the giveaway.
4. The user reveals the prize by hovering over the prize animation.
5. The user clicks the ‘Click me’ button to reveal the location of the prize.

Termination

The system outputs the location of the prize giveaway.

Post condition

The system goes into a wait state. Available to run the function again.

2.1.2.4 Requirement 4 <**Forum Page**>

Description & Priority

This allows the actor to post within the forum page. Priority is high as it is a major function within my application.

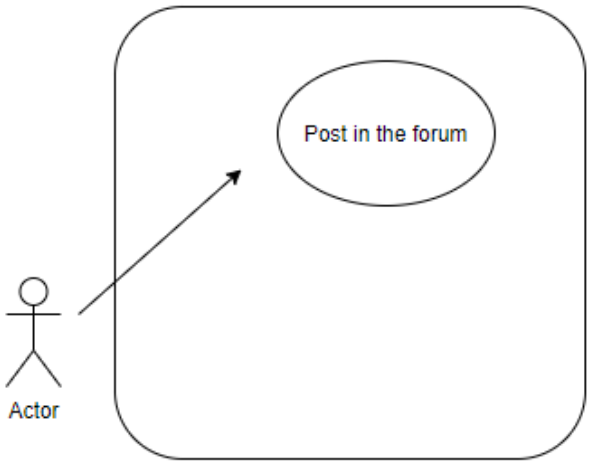
Scope

The scope of this use case is to post in the Forum.

Description

This use case describes a user making a post in the Forum page. A user can upload a Forum Title, paragraph of text and a picture to be displayed on the page.

Use Case Diagram



Flow Description

Precondition

The system is active.

Activation

This use case starts when a user enters the Forum page of the website.

Main flow

1. The user enters the Forum page.
2. The system loads the Forum page.
3. The user enters a post.
4. System adds the forum to the database and tags a timestamp.
5. System adds the post to the Forum list.

Exceptional flow

E1: < Required fields not filled out >

1. The user enters the Forum page.
2. The system loads the Forum page.
3. The user fails to enter all required fields.
4. The system checks that all required fields are entered.
5. The system presents a corresponding error code.

Termination

The system outputs the users post onto the Forum page.

Post condition

The system goes into a wait state. Available to run the function again.

2.1.2.5 Requirement 5 <**PayPal Donation Link**>

Description & Priority

This allows the actor to donate within the home page. Priority is low as it is not a major function within my application.

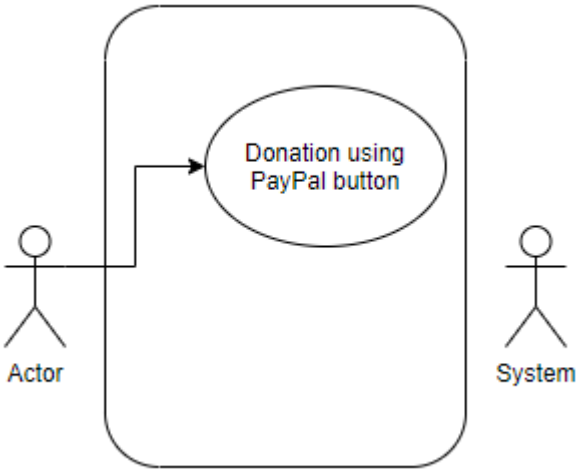
Scope

The scope of this use case is to allow a user to donate using PayPal.

Description

This use case describes a user donating to the application using a PayPal donation link. A user can access this button in the footer of the homepage.

Use Case Diagram



Flow Description

Precondition

The system is active.

Activation

This use case starts when a user enters the home page of the website.

Main flow

1. The user enters the Home page.
2. The system loads the Home page.
3. The user clicks on the PayPal donation button.
4. The user logs into their PayPal account.
5. The user pays the set donation fee of 5 euros to the Sole Seeker account.

Termination

The system retrieves the donation from the user.

Post condition

The system goes into a wait state. Available to run the function again.

2.1.3 Data Requirements

My data requirements provide insight into the data needed to perform my functionalities. While working with any data in my application, I will ensure the highest quality of confidentiality, Integrity and availability. In order for the user to access my application, they must first provide their name, email address and password, which is then stored within a secure MongoDB database.

In relation to my Forum function, my application will take the currently logged in user's name, title of their post, forum post content and an image link if the user wishes to provide one. This data is also stored within a MongoDB database and displayed on the forum page for all users to view.

Another major data requirement is the Sneaker REST API for a repository of sneakers. This API is linked to the Sole Searcher function, whereby a list of 50 pairs of sneakers are displayed on the page relevant to user specification. This API provides multiple variables relating to the results, for example gender, brand name and release date. My application requires a specific API key that is provided by TG4 Solutions. One major challenge with this data requirement is the monthly fee included. In order to use the plan that suits my application (1,000 requests/day), a monthly fee of 20 US Dollars was required.

2.1.4 User Requirements

The system will provide a place in which users can explore a wide range of useful services relating to the sneaker franchise. I will now list out the main user requirements that the application should cater for.

- A user is able to create an account and sign into the website.
- Once logged in a user is able to browse every page on the website.
- A user is able to create a post on the forum and browse through other posts made.
- A user is able to compete in the Shoe Seeker giveaway competition by the system providing the prize being awarded and the location of where the prize is hidden.
- A user is able to make a donation using the PayPal link.
- A user is able to select criteria relating to a pair of sneakers and be provided with at least 50 results.

2.1.5 Environmental Requirements

Environmental requirements are important factors to consider when developing and implementing this application. I will now list the requirements below.

- Desktop Computer
    - A desktop computer is required during development and implementation of the application. A desktop/laptop is required when gaining access to the website.
- Internet Access
    - My application requires internet access at all times in order to use the specific functions involved. The deployed URL address requires internet access to load the information from the server. Internet access is also required when sending and receiving information from the MongoDB database.

2.1.6 Usability Requirements

- **Accessibility:** The application must be accessible from different types of technology,
- **User Friendliness:** The application is easy to use with a simple navigational system to view all application services.
- **User Engagement:** The application allows for consistent and available user engagement.
- **Performance:** All functions must perform as described for the user

(Spacey, 2017)

2.1.7 Security Requirements

In terms of security requirements, our main requirement relates to personal data given by the user upon entering the application. During Login/Registration, a user will give their name, email and password. When this is entered, the data goes straight to an online MongoDB Cloud database. This ensures high confidentiality, minimising the risks of losing personal data. Additional, user passwords are fully encrypted when transferred to the MongoDB database.

2.1.8 Non-Functional Requirements

Encryption

- The system shall encrypt all user passwords that are passed to the user login database. The data shall be kept confidential just to the user.

Promotional video

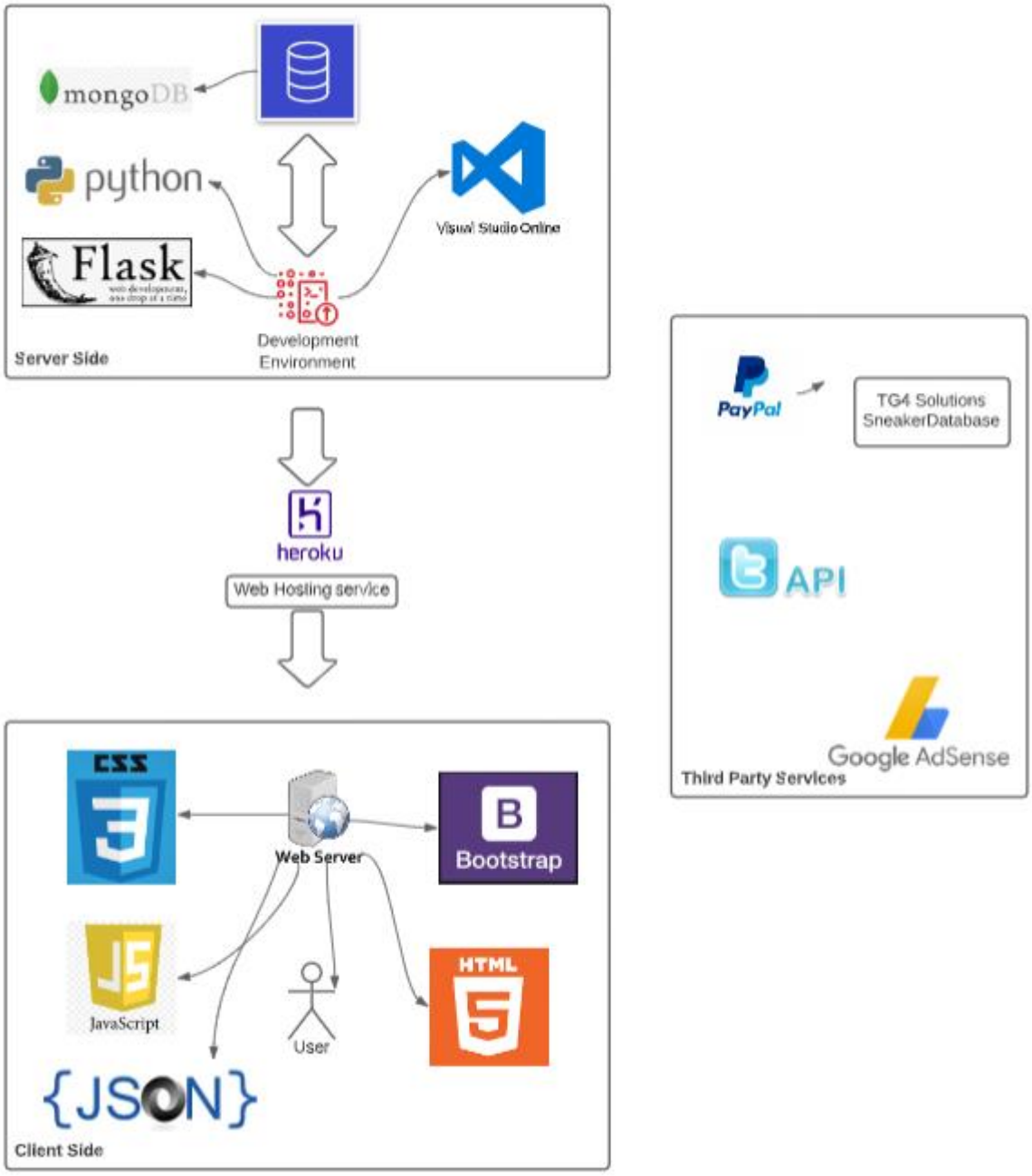
- The system shall display a promotional video highlighted in the home page describing each service that the user can interact with.

Twitter news and updates

- The system shall provide the user with recent sneaker-related news coming from selected twitter accounts.

2.2 Design & Architecture

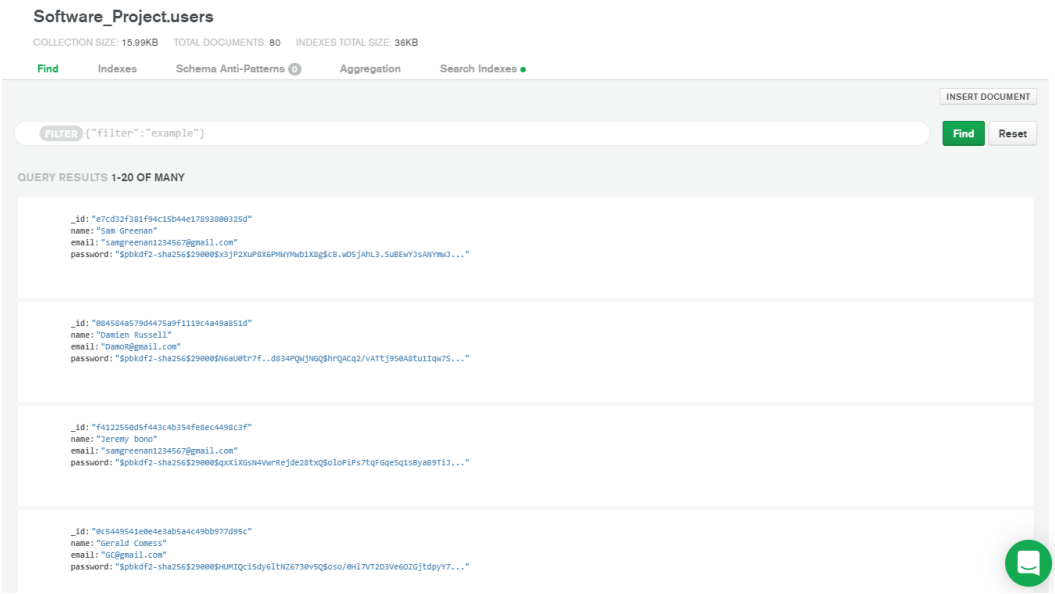
Below is my Design and architecture, that describes how my website will run. It shows the backend, frontend and third-party services of my web application.



2.3 Implementation

- Login and Registration

User accounts are stored in a MongoDB Atlas Cloud Custer as seen below.



Login/Registration Model Script

user > models.py > ...

```python
#ClassName: models.py
#  Date: 04/05/2021
#@author Sam Greenan, x17449342
#
#@reference
# https://www.youtube.com/channel/UC6Oowe4rpbQXo6AmRHmDMYg - Login Functionality User Class creation

from flask import Flask, jsonify, request, session, redirect
from passlib.hash import pbkdf2_sha256
from app import database1
import uuid

class User:

    def session1(self, user):
        del user['password']
        session['Logged_in'] = True
        session['user'] = user
        return jsonify(user), 200

    def signup(self):
        print(request.form)

        #creating the specific user
        user = {
            "_id": uuid.uuid4().hex,
            "name": request.form.get('name'),
            "email": request.form.get('email'),
            "password": request.form.get('password')
        }

        # encryption
        user['password'] = pbkdf2_sha256.encrypt(user['password'])


        #removing duplicates to check whether the email address already exists
        if database1.users.find_one({"email": user['email'] }):
            return jsonify({"error": "Email entered already has an account"}), 400


        if database1.users.insert_one(user):
            return self.session1(user)



        return jsonify({"error": "Signup has failed" }), 400

    #sign out method
    def signout(self):
        session.clear()
        return redirect('/')

    #sign in method
    def login(self):

        user = database1.users.find_one({
            "email": request.form.get('email')
        })

        if user and pbkdf2_sha256.verify(request.form.get('password'), user['password']):
            return self.session1(user)

        return jsonify({ "error": "Login has failed, Email or Password is incorrect" }), 401
```

Login/Registration Routes

user > routes.py > ...

```python
#ClassName: models.py
#  Date: 04/05/2021
#@author Sam Greenan, x17449342
#
#@reference
# https://www.youtube.com/watch?v=w1STSSumoVk - Login routes


from flask import Flask

#from the app file, import the instance of Flask
from app import app

from user.models import User

from forum.forumModels import Forum

@app.route('/user/signup', methods =['POST'])
def signup():
    return User().signup()

@app.route('/user/signout')
def signout():
    return User().signout()

@app.route('/user/login', methods =['POST'])
def login():
    return User().login()

@app.route('/forum/post', methods=['POST'])
def posting():
    return Forum().forumPost()
```

Login/Registration Decorators

```python

# Decorators
def login_manditory(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if 'Logged_in' in session:
            return f(*args, **kwargs)
        else:
            return redirect('/')
    return wrap

```

Login/Registration Database connection and Cluster Creation

```
app.secret_key = '[redacted]'

#connecting to the database
cluster = MongoClient('mongodb+srv://root:[redacted]@cluster0.szmco.mongodb.net/Software_Project?retryWrites=true&w=majority')


database1 = cluster["Software_Project"]
collection = database1["users"]
```

- Shoe Seeker Folium Map API Creation

Using the Folium package to create the radius for the prize location

```
@app.route("/maptest")
@login_manditory
def map():
    map = folium.Map(
        location=[53.3379897, -6.259073],
        tiles='Stamen Terrain',
        zoom_start=17
    )

    folium.Circle([53.3379897, -6.259073],
    radius=200).add_to(map)

    return map._repr_html_()

```

- PayPal API Integration

Importing the PayPal API

```python
import paypalrestsdk
paypalrestsdk.configure({
  "mode": "sandbox", # sandbox or live
  "client_id": "ARli33FAGZdOi6d1LcT3BID4BDyHLYCSJ68OBqQXKxb_9ozyPcaQdOXV7u_MB2QwpsoRwmwQuWE1tYZc",
  "client_secret": "ENYKf54NVyxnSAGjNl71TieaYYincS_PIXN7GAKXqQh8OowOO3UHv6VO86xClt5c0W1UGr1-aUGbQYOn" })


@app.route('/payment', methods=['POST'])
def payment():

    payment = paypalrestsdk.Payment({
        "intent": "sale",
        "payer": {
            "payment_method": "paypal"},
        "redirect_urls": {
            "return_url": "http://localhost:3000/payment/execute",
            "cancel_url": "http://localhost:3000/"},
        "transactions": [{
            "item_list": {
                "items": [{
                    "name": "testdonation",
                    "sku": "item",
                    "price": "5.00",
                    "currency": "EUR",
                    "quantity": 1}]},
            "amount": {
                "total": "5.00",
                "currency": "EUR"},
            "description": "This is a 5 Euro PayPal donation to Sole Seeker."}]})

    if payment.create():
```

```python
        print('successfull payment!')
    else:
        print(payment.error)

    return jsonify({'paymentID': payment.id})


@app.route('/execute', methods=['POST'])
def execute():
    success = False
    payment = paypalrestsdk.Payment.find(request.form['paymentID'])

    if payment.execute({'payer_id' : request.form['payerID']}):
        print('Execute sucess!')
        success = True
    else:
        print(payment.error)



    return jsonify({'success' : success})

```

Integration into the homepage template

```
<center>
  <p class="whitefont">Donation Link</p>
  <div id="paypal-button"></div>

  <script src="https://www.paypalobjects.com/api/checkout.js"></script>

  <script>
      var CREATE_PAYMENT_URL  = 'http://127.0.0.1:5000/payment';
      var EXECUTE_PAYMENT_URL = 'http://127.0.0.1:5000/execute';

      paypal.Button.render({

          env: 'sandbox', // Or 'sandbox'

          commit: true, // Show a 'Pay Now' button

          payment: function() {
              return paypal.request.post(CREATE_PAYMENT_URL).then(function(data) {
                  return data.paymentID;
              });
          },

          onAuthorize: function(data) {
              return paypal.request.post(EXECUTE_PAYMENT_URL, {
                  paymentID: data.paymentID,
                  payerID:   data.payerID
              }).then(function(res) {

                  console.log(res.success)
                  // The payment is complete!
                  // You can now show a confirmation message to the customer
              });
          }

      }, '#paypal-button');
  </script>
  </center>
</div>
```

- Sole Searcher REST API function

As this is one of the applications main functions, I felt it was appropriate to highlight the step by step progression of creating this function.

Step 1: Research

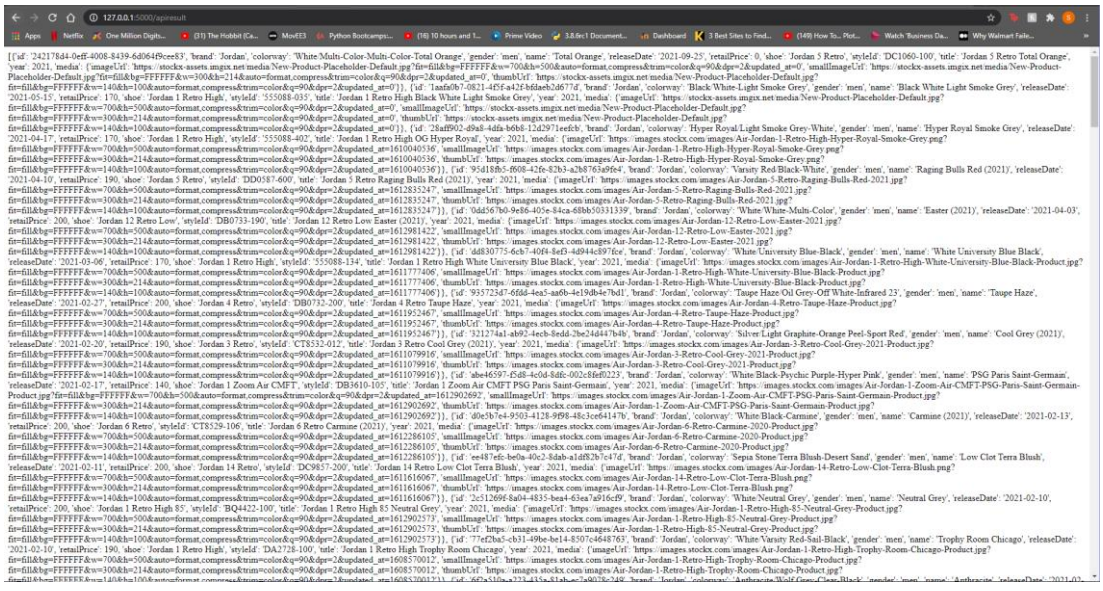
Instead of creating my own sneaker database, I decided on using an already existing one. This reduces the need to manually update the database regularly. The database will be automatically updated over time using a Sneaker database online API. I searched vigorously to try and find the best API to suit my requirements. I decided to use 'The Sneaker database', that can be found on Swaggerhub.com, created by TG4 Solutions.

Step 2: API Connectivity and choosing sneaker specifics.

I received an API key on the Swaggerhub page for the sneaker database. I also created two variables for the 'request.form' for ease of use later on. At the beginning I chose brand and gender as they worked with the API and would display many results.

```
@app.route("/apiresult", methods=["POST"])
@login_manditory
def apiresult():
    #getting the form value
    brandname = request.form['brand']
    gendertype = request.form['gender']
    r = requests.get('https://api.thesneakerdatabase.com/v1/sneakers?limit=100&brand='+brandname+'&gender='+gendertype)
```

Below is the raw data that I received when calling the Plain API key in my application. I then used this data in my next phase when parsing the data.



Step 3: Creating the JSON Object and parsing the data.

I first created a JSON object in order to assist in parsing the data.

```
json_object = r.json()
```

Then to get my results, I created variables for each parsed json object. For example, 'name_result', obtains the raw data then takes the name of the sneaker at position 0. I then replicated this many times to get multiple results.

```
#sneaker 1
    name_result = json_object['results'][0]['name']
    shoe_result = json_object['results'][0]['media']['imageUrl']
    color_result = json_object['results'][0]['colorway']

    #sneaker 2
    name_result2 = json_object['results'][1]['name']
    shoe_result2 = json_object['results'][1]['media']['imageUrl']
    color_result2 = json_object['results'][1]['colorway']

    #sneaker 3
    name_result3 = json_object['results'][2]['name']
    shoe_result3 = json_object['results'][2]['media']['imageUrl']
    color_result3 = json_object['results'][2]['colorway']

    #sneaker 4
    name_result4 = json_object['results'][3]['name']
    shoe_result4 = json_object['results'][3]['media']['imageUrl']
    color_result4 = json_object['results'][3]['colorway']

    #sneaker 5
    name_result5 = json_object['results'][4]['name']
    shoe_result5 = json_object['results'][4]['media']['imageUrl']
    color_result5 = json_object['results'][4]['colorway']
```

Step 4: Taking in User input

In order to relate the API results to the user, I must create a form for the user to submit to the application. The user must select a specific shoe brand and gender to receive results. If the user fails to enter all criteria, an error message will appear.

```
<form method="POST" action="/apiresult" class="center">
    <h1>Select a shoe brand:</h1> <input type="radio" name="brand" value="nike" required>Nike
                                  <input type="radio" name="brand" value="Jordan" required>Jordan
                                  <input type="radio" name="brand" value="vans" required>Vans
                                  <input type="radio" name="brand" value="converse" required>Converse
                                  <input type="radio" name="brand" value="adidas" required>Adidas
                                  <input type="radio" name="brand" value="new balance" required>New Balance


</br>

    <h1>Select your gender: </h1> <input type="radio" name="gender" value="men" required>Male
                                  <input type="radio" name="gender" value="women" required>Female <br>



    <input type="submit" value="Submit" style ="background-color: red" class='btn'>


</form>
```

Step 5: Relating User input to API database and posting the results.

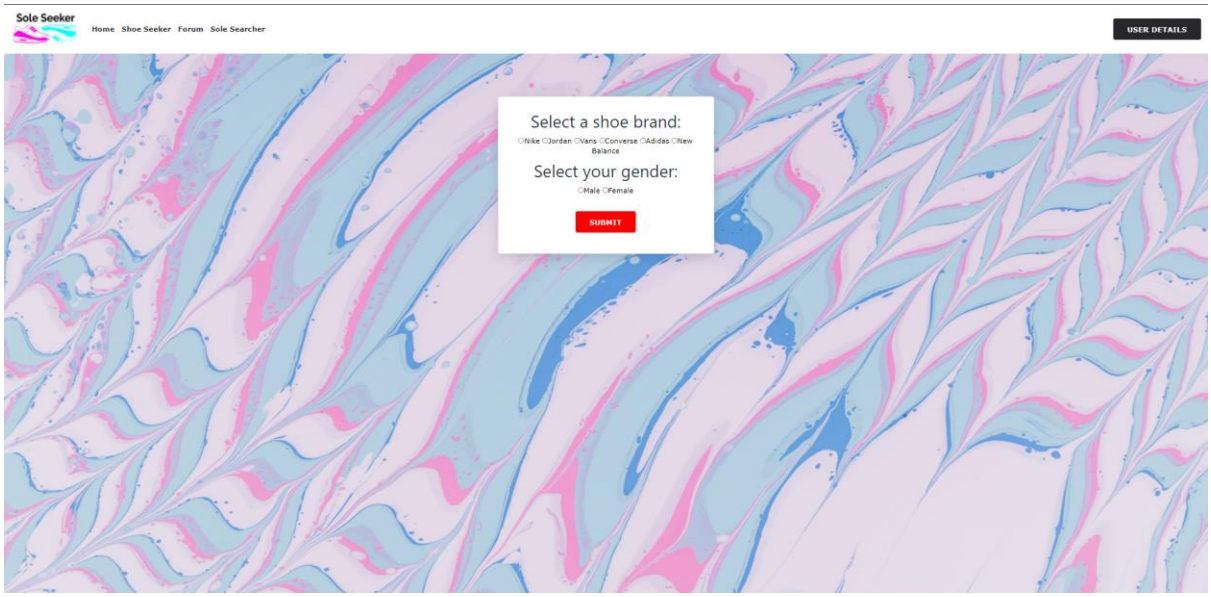
In order to return the sneaker results, we need to redirect the user to the apiresult.html. This page will hold the results coming from the user specification.

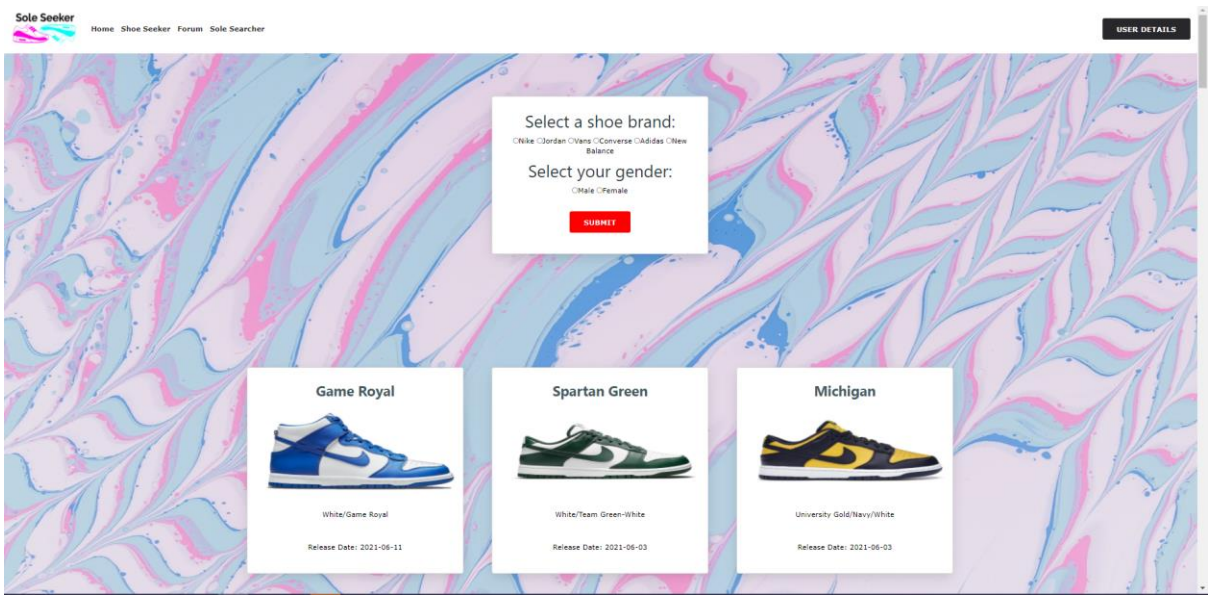
```
#return shoe_result
return render_template('apiresult.html',
imageUrl=shoe_result, name=name_result, color=color_result,
imageUrl2=shoe_result2, name2=name_result2,color2=color_result2,
imageUrl3=shoe_result3, name3=name_result3,color3=color_result3,
imageUrl4=shoe_result4, name4=name_result4,color4=color_result4,
imageUrl5=shoe_result5, name5=name_result5,color5=color_result5,
imageUrl6=shoe_result6, name6=name_result6,color6=color_result6,
imageUrl7=shoe_result7, name7=name_result7,color7=color_result7,
imageUrl8=shoe_result8, name8=name_result8,color8=color_result8,
imageUrl9=shoe_result9, name9=name_result9,color9=color_result9,
imageUrl10=shoe_result10, name10=name_result10,color10=color_result10,
imageUrl11=shoe_result11, name11=name_result11,color11=color_result11,
imageUrl12=shoe_result12, name12=name_result12,color12=color_result12,
imageUrl13=shoe_result13, name13=name_result13,color13=color_result13,
imageUrl14=shoe_result14, name14=name_result14,color14=color_result14,
imageUrl15=shoe_result15, name15=name_result15,color15=color_result15,
imageUrl16=shoe_result16, name16=name_result16,color16=color_result16,
imageUrl17=shoe_result17, name17=name_result17,color17=color_result17,
imageUrl18=shoe_result18, name18=name_result18,color18=color_result18,
imageUrl19=shoe_result19, name19=name_result19,color19=color_result19,
imageUrl20=shoe_result20, name20=name_result20,color20=color_result20,)
```

Below is the code snippet of displaying the results on the ‘apiresult’ page.

```
<!-- result one -->
<div class='card-wrapper'>
    <div class="card">



<h2 class="center"><strong> {{ name }}</strong></h2>
<img class="center" src='{{imageUrl}}' width='400' height='200' alt='Sneaker Image Not Available'>
<p class="center">{{color}}</p>
<p class="center">Release Date: {{releaseDate}}</p>
    </div>



<!-- result two -->
<div class="card">

<h2 class="center"> <strong> {{ name2 }}</strong></h2>
<img class="center" src='{{imageUrl2}}' width='400' height='200' alt='Sneaker Image Not Available'>
<p class="center">{{color2}}</p>
<p class="center">Release Date: {{releaseDate2}}</p>
</div>


<!-- result three -->

    <div class="card">
```

The ends result can be seen below.





Step 6: API Issues and Resolution

While I was conducting User Testing for my web application, a major error emerged from this function. It stated that there was an Internal Server Error relating to the Sneaker results. Through inspection, I found that no results were now being pulled from the TG4 Solutions API.

Due to this I sent an email to the developer of the API, querying if the API was taken down. His response can be seen below.

Hey Sam,

We transitioned over to require api keys to use our api. I apologize for any inconvenience this may have caused. We attempted to contact all known users of our api via email. However, now that we require an api key we now know who all is using our api and can send announcements with more confidence. Below you can find a link to obtain an api key.

https://rapidapi.com/tg4-solutions-tg4-solutions-default/api/v1-sneakers/pricing

Please let me know if you have any questions or concerns.

Due to this I had to alter my old code to facilitate this new API key. It took a while to try figure out how facilitate this new type of API key into my code. However, I got it back working over a couple days. I also now have to pay a monthly fee of 20 dollars to match the amount of API requests my application requires. Below you can see the changes I made to my main python script to adapt to this new API key.

```
111  @app.route("/newapi", methods=["GET","POST"])
112
113  def newapi():
114
115
116      url = 'https://v1-sneakers.p.rapidapi.com/v1/sneakers'
117
118
119      brandname1 = request.form.get('brand')
120      gendertype1 = request.form.get('gender')
121
122
123
124
125      querystring = {"limit":"100", "brand":brandname1, "gender":gendertype1,}
126
127      headers = {
128      'x-rapidapi-key': "[redacted]",
129      'x-rapidapi-host': "v1-sneakers.p.rapidapi.com"
130      }
131
132      response = requests.request("GET", url, headers=headers, params=querystring)
133
134
135      json_object = response.json()
136
137
```

- Forum function

As this function plays a large importance in my application, I will follow the same format of a step by step progression of the development of this service.

Step 1: Template creation

I first created the template for users to enter information about their post.

```
82    @app.route("/forum")
83    @login_manditory
84    def forum():
85        return render_template("forum.html")
```

```
30  <div class="Reviewer">
31      <div class='card-wrapper'>
32      <div class="card">
33      <h1 class="center">Add to the forum</h1>
34
35      <form name="forum_form">
36
37
38
39          <label for="forumTitle">Title</label>
40          <input type="text" name="forumTitle" class="field" maxlength="300" required>
41
42          <label for="forumPost">Post</label>
43          <textarea rows="4" cols="40" name="forumPost"  required></textarea>
44
45          <label for="forumImage">Include an Image  (Image Address Only)</label>
46          <input type="text" name="forumImage" placeholder="*Not Required" class="field" >
47
48
49
50
51
52
53
54
55          <p class="error error--hidden"></p>
56
57          <input onClick="window.location.reload();" type="submit" value="Submit" style ="background-color: red" class='btn'>
```

Step 2: Cluster Collection, Creation and Connection

A Collection was created to store the posts made by the user. The collection then gets passed to the forum template as a new variable ‘collectiontest’.

```
#connecting to the database
cluster = MongoClient('mongodb+srv://root:[redacted]@cluster0.szmco.mongodb.net/Software_Project?retryWrites=true&w=majority')


database1 = cluster["Software_Project"]
collection = database1["users"]

database2 = cluster["Software_Project"]
collection1 = database2["forumCluster"]
collection2 = database2["forums"]
```

```
@app.route("/forum")
@login_manditory
def forum():
    return render_template("forum.html", collectiontest=collection2)
```

Step 3: Making user posts write to the MongoDB collection.

In order to get the posts to be sent to the database, we must create a Model script to facilitate this.

forum > forumModels.py > ...

```
from flask import Flask, jsonify, request, session
import uuid
from app import database2
from app import database1


class Forum:


    def forumPost(self):
        #creation of the forum post object
        forum = {
            "_id": uuid.uuid4().hex,
            "forumName": session['user']['name'],
            "forumTitle": request.form.get('forumTitle'),
            "forumPost": request.form.get('forumPost'),
        }

        if database2.forums.insert_one(forum):
            return jsonify(forum), 200

        return jsonify({"error": "failed"}), 400

```

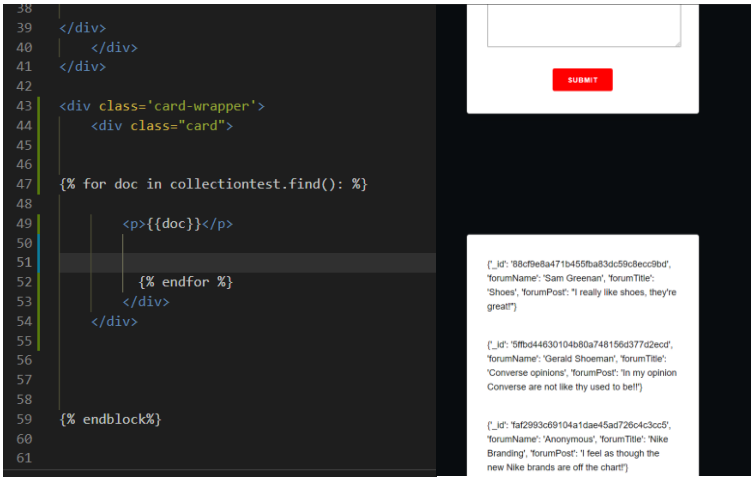
The above Python script firstly imports the necessary packages and features needed to run the functions inside. The 'Forum' class is created to hold the 'forumPost' function. Within this function the creation of the forum post object persists.

- **_id**: The ID is a hex digit that creates an individual ID for each post. This ID classifies each post.
- **forumName**: This variable uses Flask sessions to grabs the name of the currently logged in user.
- **forumTitle**: This variable gets the 'forumTime' from our Forum template in order to grab what the user has entered.
- **forumPost**: This variable gets the 'forumPost' from our Forum template in order to grab what the user has entered.

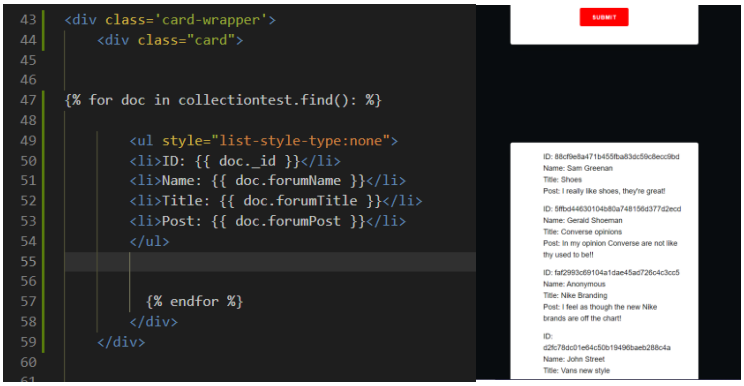
The script then inserts these variables to the Mongo DB database using the inser_one method.

Step 4: Reading from the database and posting the forum posts onto the webpage.

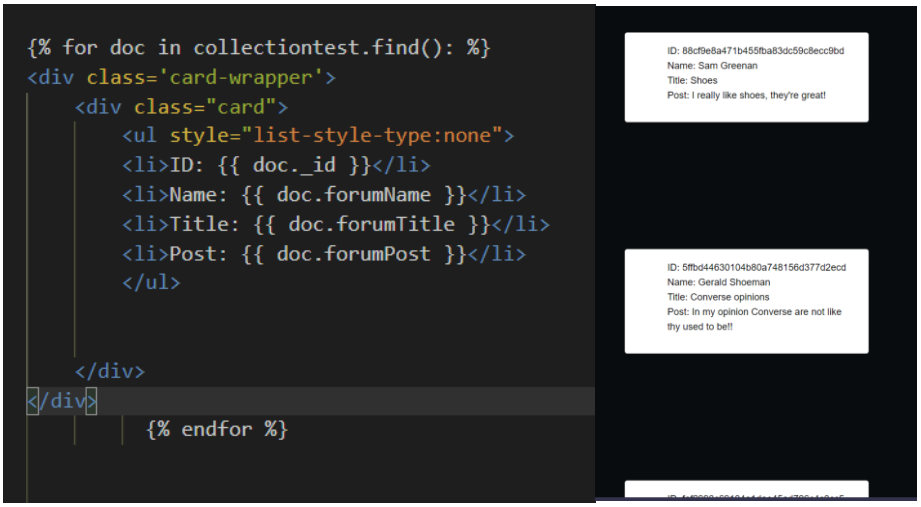
I started off by implementing a simple For loop to grab all posts in the Collection database as seen below.



I then displayed the results in a list format.



Then to get the posts to print separately I wrapped the list in a card wrapper.



```
{% for doc in collectiontest.find(): %}
<div class='card-wrapper'>
    <div class="card">
        <ul style="list-style-type:none">
        <li>ID: {{ doc._id }}</li>
        <li>Name: {{ doc.forumName }}</li>
        <li>Title: {{ doc.forumTitle }}</li>
        <li>Post: {{ doc.forumPost }}</li>
        </ul>


    </div>
</div>
        {% endfor %}
```


Step 5: Adding new variables.

The new variables that I wanted to add to the Forum were **forumTime and forumImage**.

- **forumTime**: In order to print the newest post at the top of the page, I realised I needed to implement a time-variant variable. Through the creation of this I could set the newest post to always be placed at the top of the page.
- **forumImage**: From research I have found that many Forum pages on the internet allow the user to attach an image to their post.

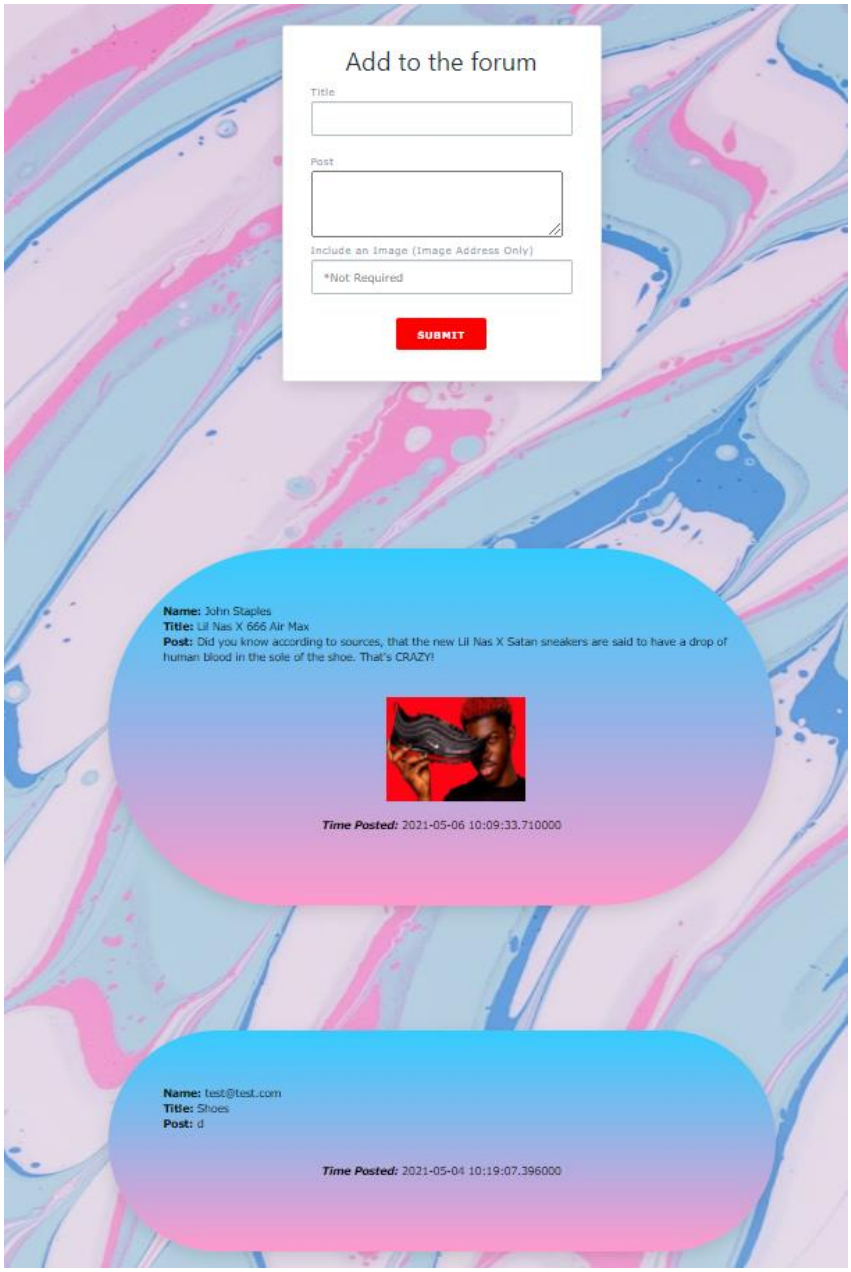
To add in these variables, I went back to my Models script and made additions.

forum > forumModels.py > ...

```
1   from flask import Flask, jsonify, request, session
2   from datetime import datetime
3   import uuid
4   import time
5   from app import database2
6   from app import database1
7
8
9   class Forum:
10
11
12      def forumPost(self):
13          #creation of the forum post object
14          forum = {
15              "_id": uuid.uuid4().hex,
16              "forumName": session['user']['name'],
17              "forumTitle": request.form.get('forumTitle'),
18              "forumPost": request.form.get('forumPost'),
19              "forumTime": datetime.now(),
20              "forumImage": request.form.get('forumImage')
21          }
22
23          if database2.forums.insert_one(forum):
24              return jsonify(forum), 200
25
26          return jsonify({"error": "failed"}), 400
27
28
```

Step 6: Design Implementation

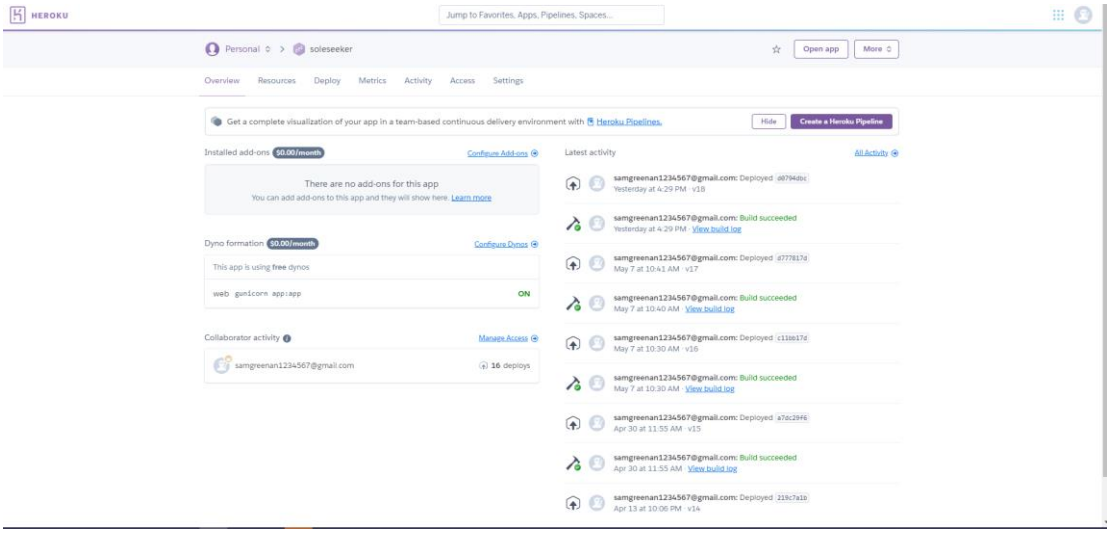
Now that a user can post in the forum, have their post sent to the database then printed onto the webpage, the next stage is design. I used different CSS techniques to create a new post layout.



- Sole Searcher Deployment

As Sole Seeker is a web application, it is essential that it is deployed when completed. In order to deploy Sole Seeker, I used Heroku. Heroku is a cloud based deployment service that performs as a Platform as a Service (PaaS). Using the basic, no fee subscription, I was able to deploy my application allowing for 512MB of RAM. The deployed site runs smoothly; however, it can get slow when a large number of users join at once. In the future as I develop this application further, I will upgrade to allow for more users to visit the site at the same time.

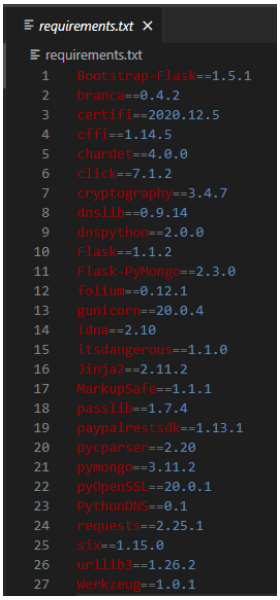
Below is a screenshot of the Heroku Interface, noting the number of deployments and the successful builds over time.



Link to deployed site: https://soleseeker.herokuapp.com/

- Application Packages

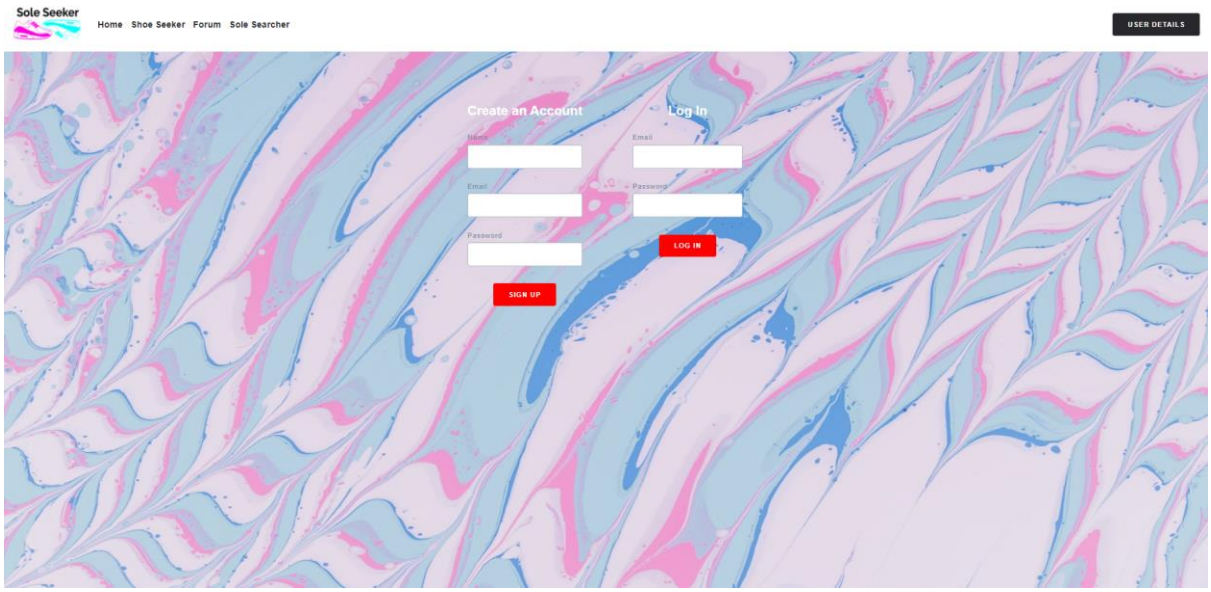
Please see below the list of packages used during development. I used a virtual environment to store these packages in order keep them isolated.



2.4 Graphical User Interface (GUI)

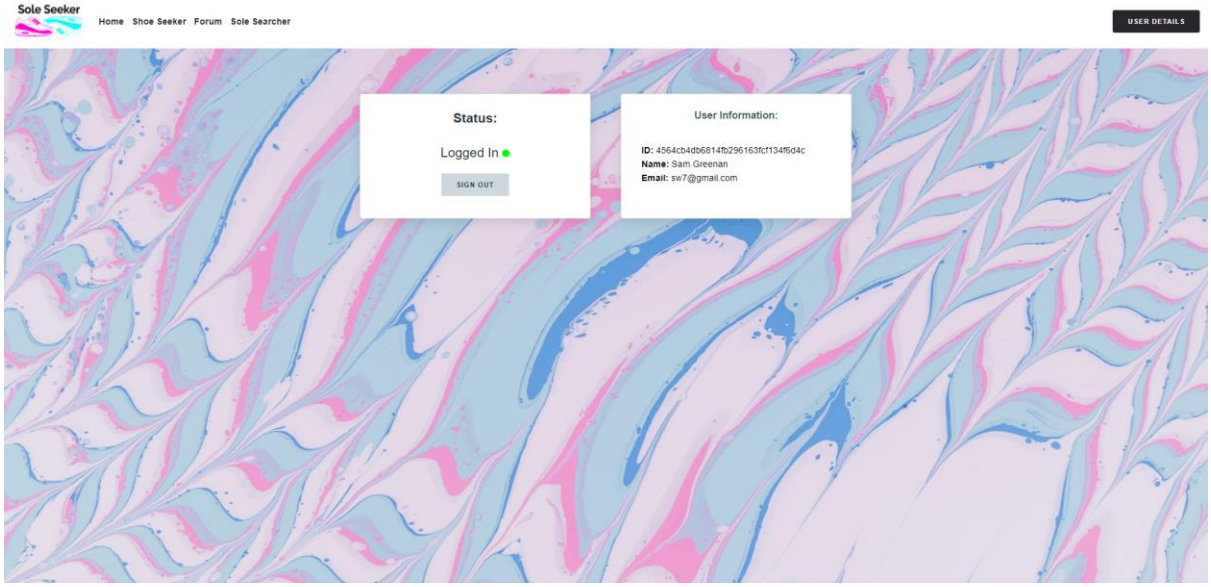
I will now go through each page of my website and highlight the functions included.

User Registration and Login page



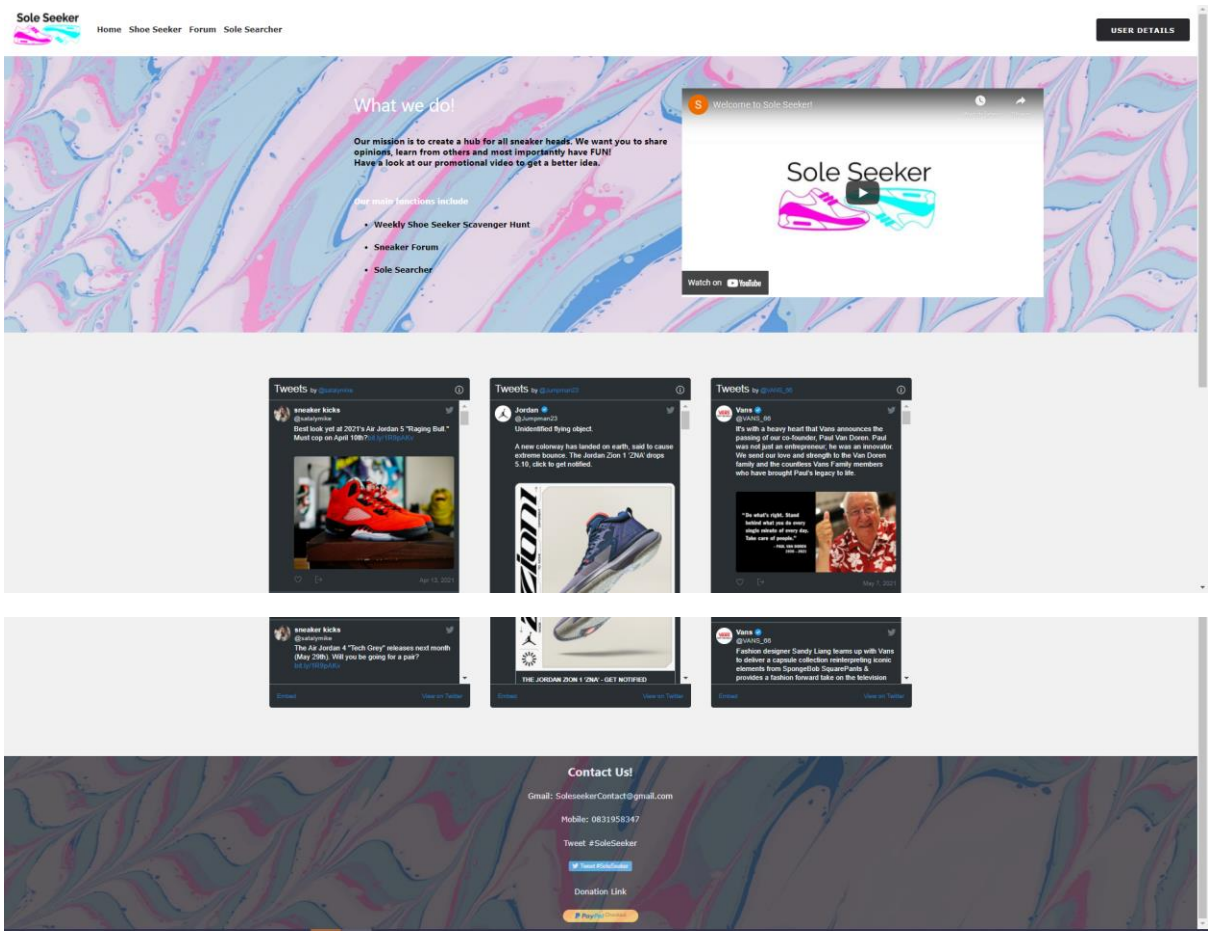
This is the first page the user will see when they enter my website. A user must either log in or register to view any page on the website. User information is validated and sent to an online MongoDB database where it is stored correctly. Passwords are fully encrypted within the database, utilizing the Flask passlib extension. Methods are also in place to ensure users input a valid email address.

User Information page



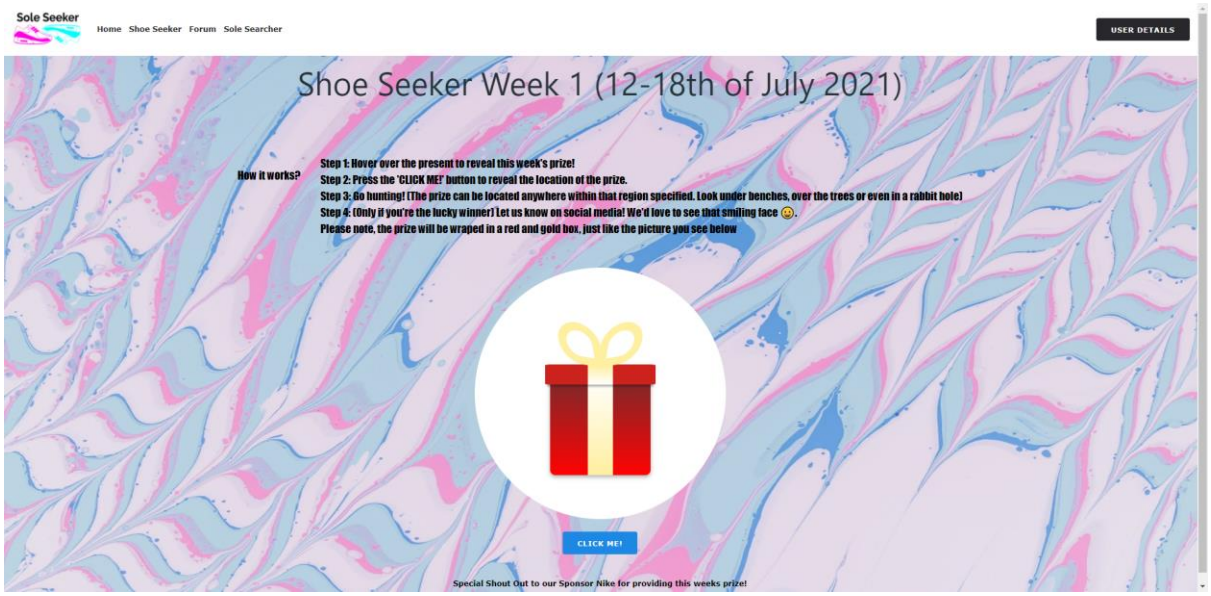
Once a user is logged in, they are presented with their current status and user information that relates to their account. This reassures that a user is logged in and that they can now enter any page on the website.

Home page



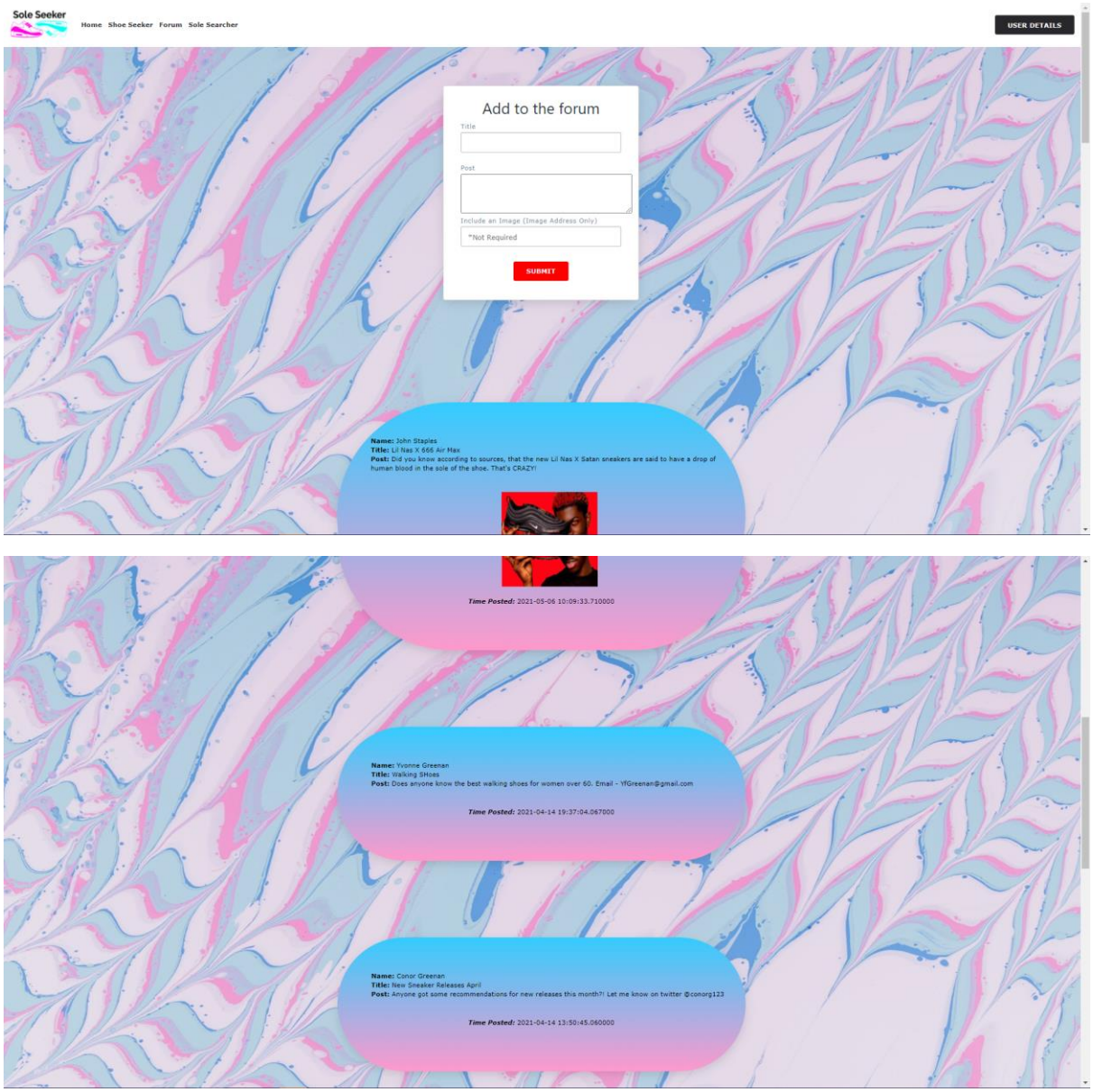
The homepage gives the user a brief overview of the many services within the website. Listed on the page are the main functions that are within the website. The paragraph of text is accompanied by a promotional video that goes into detail on the services and what Sole Seeker is all about. Below this are three popular Twitter accounts that provide Sneaker news and updates. Tweets are updated regularly due to the implementation of the Twitter API. Towards the end of the page are contact details and a #SoleSeeker tweet button. Also, there is a PayPal donation link utilizing the PayPal API.

Shoe Seeker page



The Shoe Seeker page showcases the giveaway competition aspect of my application. This is a chance for one of my users to win a pair of new sneakers. The user should follow the steps laid out in order to take part in the giveaway. These steps gives the user the chance to take part. The user will firstly reveal the prize through hovering over the present image. Using CSS animation, the prize will pop out of the box. After that the user will press the 'Click Me' button to reveal the location of the prize. This will enable a Map API, showing a circular radius around an area of Dublin. The prize can then be found anywhere within that radius in real life. As stated on the page, the pair of sneakers will be wrapped in a red and gold box, just like the one on the webpage. At the end of this web page is a special shout out message to the sponsor of that week's prize.

Forum page



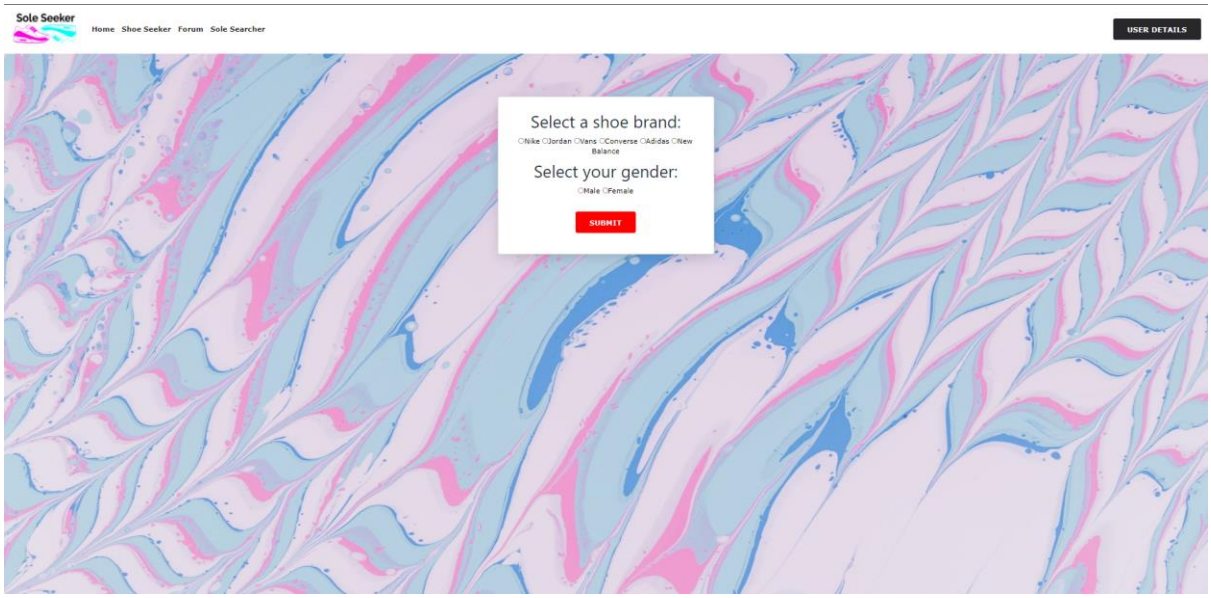
This page provides the service to allow Sole Seeker members to share their thoughts on Sneaker related news and trends. When a user wants to write a post there is certain criteria which must be filled out.

- **Title:** This input type gives the opportunity for the user to display the title of their post. This I what will grab the other user's attention when scrolling through the forum. Users should keep this short and concise and try and convey what they're post will be based upon.
- **Post:** This is the content of the post. The user will enter their full post here.
- **Image:** Users also have the option to include an image with their post. A user shall provide an 'image address' as the way in which to include their image.

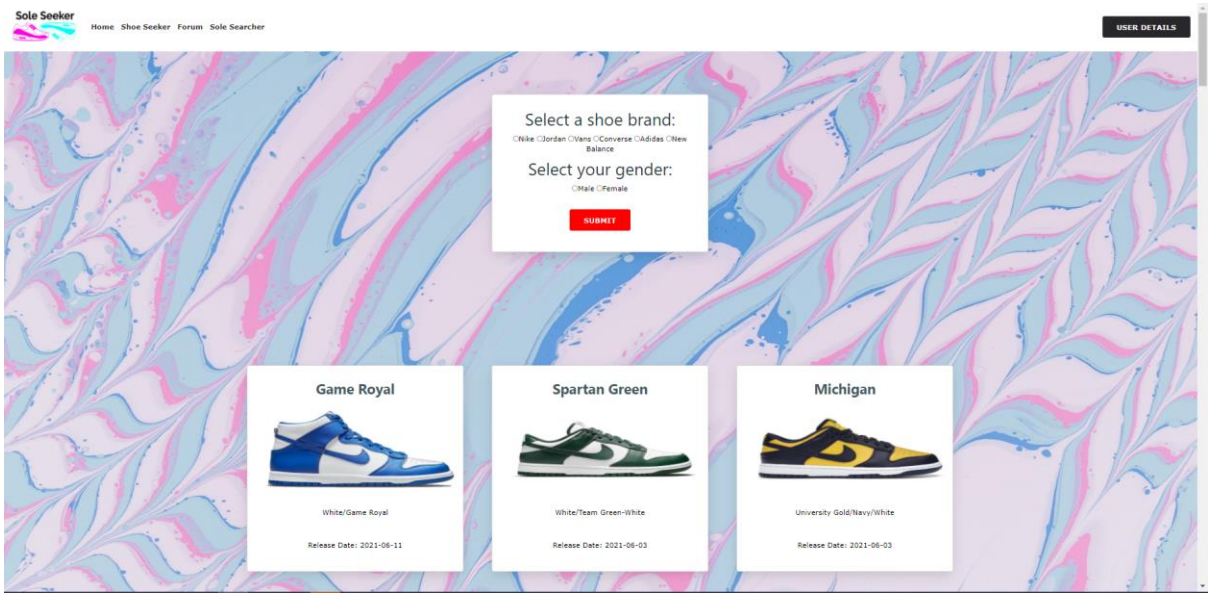
Once the user submit their post, the contents will be sent to a MongoDB cluster for storage. The server will then pull the post from the database and add it to the list of other posts on

the page. Utilizing Flask sessions, the user who made the post will also have their name tagged alongside the post.

Sole Searcher function page



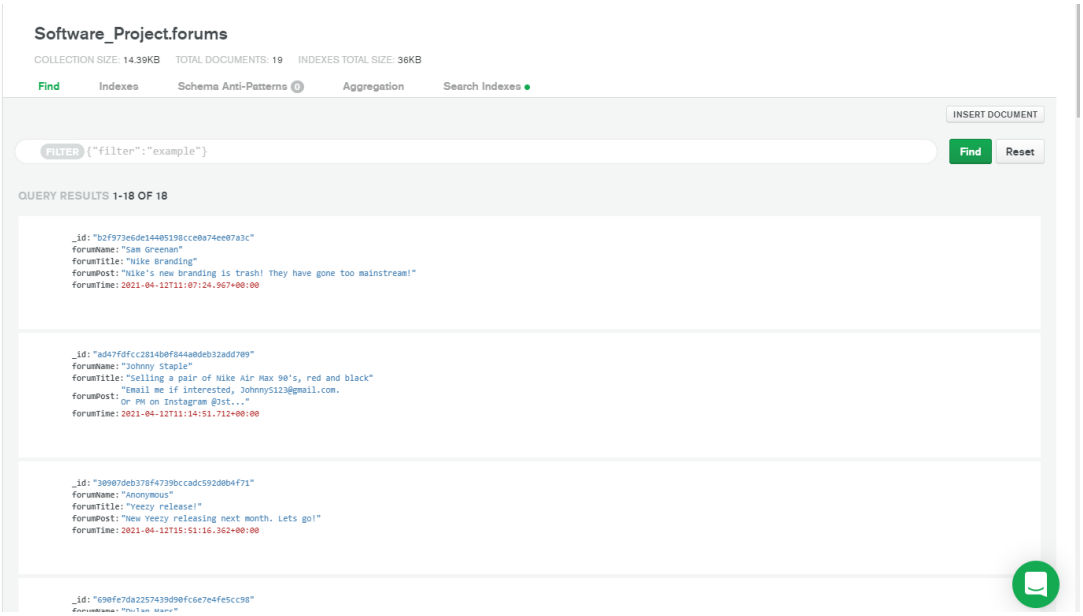
Sole Searcher output Page

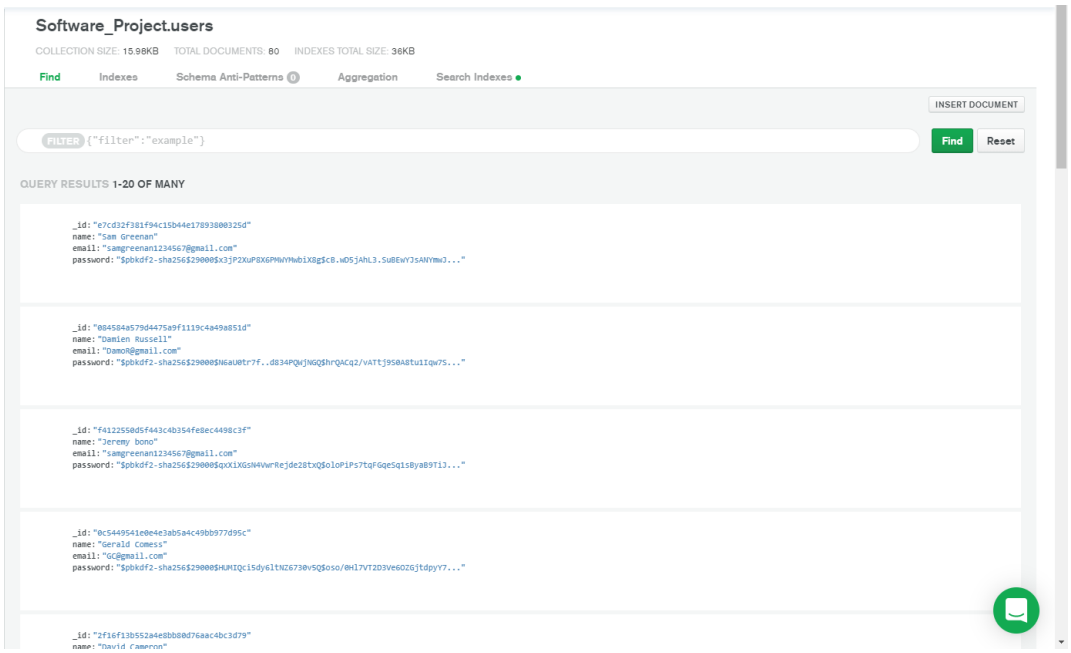


This is quite an important and useful service on Sole Seeker. Using the TG4 Solutions online database, I was able to create a function whereby a user can choose their preferred shoe brand and gender and be presented with 50 sneaker results that match their preference. I used a REST API key to connect with the database and use different functions to provide the service. I chose to only include shoe brand and gender in the search function for simplicity and to maximise results.

Once the API is called, 50 sneaker results are printed onto the web page. Each result previews the sneaker name, sneaker image, colourway and release date. I chose these variables as they best describe each pair of sneakers. I considered using another available variable ‘retailPrice’, however the data provided was not consistent. Many results had a ‘retailPrice’ value of ‘0’ or ‘null’. As this was something I couldn’t change I decided on leaving this variable out to improve data integrity.

Mongo DB Atlas Interface





Above are two snippets of the collection of posts and registered users in the Mongo DB Atlas Cloud database. As an admin I can delete any posts or remove any user accounts.

2.5 Testing

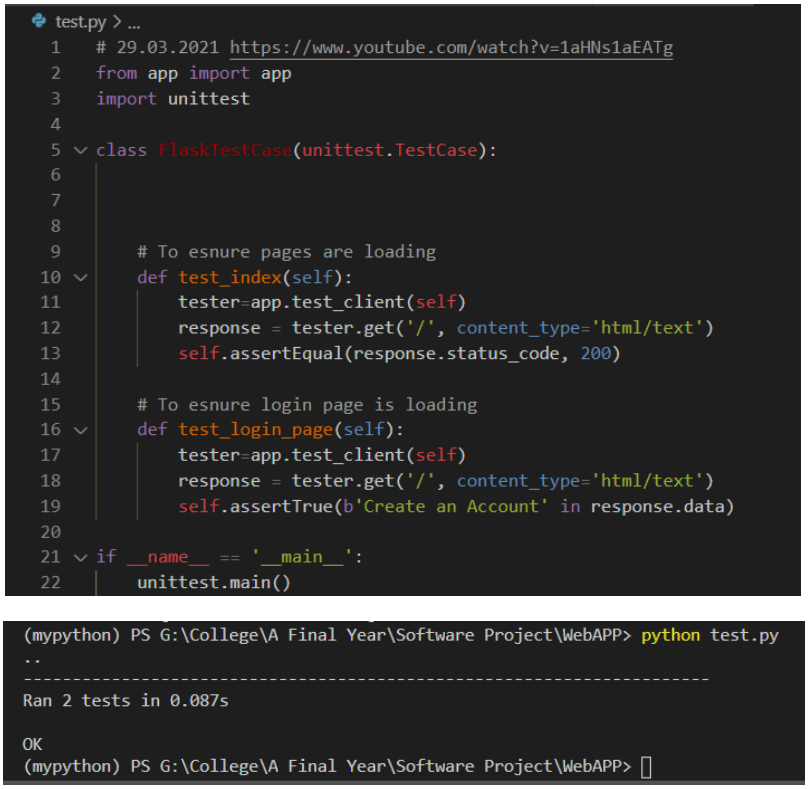
The two type of testing techniques I have chosen to complete are Unit testing and User Testing. I chose these as I feel they are most suitable for my type of project. Unit testing will allow me to test evidence of rendering and functionality with different web pages. Whereas Unit testing will help me garner feedback for my application. I will use this feedback to fix any system bugs or facilitate user preferences. User testing will also let me know what I'm doing right and get second opinions on different ideas.

➢ Unit testing

In order to test whether my pages were loading, I performed a unit test to measure if all pages responded with a status code of 200. This means the HTTP was accepted. We will also test to see if the data on the page is returning. We test this also to make sure the right page was rendered during the test.

In order to test if the application pages were loading, I needed temporarily remove the @login_manditory function from all 'app.routes'. I needed to do this as all pages cannot be accessed unless a user is logged in. After all unit testing was performed, I returned this @login_maditory function to all 'app.routes'.

Login/Registration Page:



Homepage:

```python
from app import app
import unittest

class FlaskTestCase(unittest.TestCase):


    def test_index(self):
        tester=app.test_client(self)
        response = tester.get('/homescreen', content_type='html/text')
        self.assertEqual(response.status_code, 200)

if __name__ == '__main__':
    unittest.main()
```

```
(mypython) PS G:\College\A Final Year\Software Project\WebAPP> python test.py
.
----------------------------------------------------------------------
Ran 1 test in 0.009s

OK
(mypython) PS G:\College\A Final Year\Software Project\WebAPP>
```

Shoe Seeker Page:

```python
from app import app
import unittest

class FlaskTestCase(unittest.TestCase):




    def test_index(self):
        tester=app.test_client(self)
        response = tester.get('/shinder', content_type='html/text')
        self.assertEqual(response.status_code, 200)

if __name__ == '__main__':
    unittest.main()
```

```
(mypython) PS G:\College\A Final Year\Software Project\WebAPP> python test.py
.
----------------------------------------------------------------------
Ran 1 test in 0.009s

OK
(mypython) PS G:\College\A Final Year\Software Project\WebAPP>
```

Forum Page:

```python
from app import app
import unittest

class FlaskTestCase(unittest.TestCase):




    def test_index(self):
        tester=app.test_client(self)
        response = tester.get('/forum', content_type='html/text')
        self.assertEqual(response.status_code, 200)

if __name__ == '__main__':
    unittest.main()
```

```
(mypython) PS G:\College\A Final Year\Software Project\WebAPP> python test.py
.
----------------------------------------------------------------------
Ran 1 test in 0.008s

OK
(mypython) PS G:\College\A Final Year\Software Project\WebAPP>
```

Sole Searcher:

```python
from app import app
import unittest

class FlaskTestCase(unittest.TestCase):




    def test_index(self):
        tester=app.test_client(self)
        response = tester.get('/APIsearch', content_type='html/text')
        self.assertEqual(response.status_code, 200)

if __name__ == '__main__':
    unittest.main()
```

```
(mypython) PS G:\College\A Final Year\Software Project\WebAPP> python test.py
.
----------------------------------------------------------------------
Ran 1 test in 0.009s

OK
(mypython) PS G:\College\A Final Year\Software Project\WebAPP>
```

To conclude this, all application pages return the status code of 200 when tested, therefore all pages are rendered properly.

- **User Testing**

I performed user testing for a sample size of five. I gained very informative feedback that benefited the progress of my web application. It helped me find areas that were lacking and areas in which were thriving. To facilitate the User testing I constructed the following steps for the participants to follow:

Step 1: The user will agree to participate by signing the consent form.

Step 2: The user will receive a link to access the web application or test locally on a development server.

Step 3: The user will browse the website how they please.

Step 4: The user will fill out the User testing survey and submit.

Total Time: 15 minutes

Application Testing: 10 Minutes

Survey: 5 Minutes

- The consent form was made using JOT form. A link of the form was sent to each user participating in the test. The form required and name, last name, email and digital signature.
- Once submitted the user was sent a link to the deployed web application or accessed through a local server whereby, they would browse the website as they please.
- After testing the application, the participant was sent a Google forms survey. They survey contained simple questions relating to their experience using the application.
- I would then take this information to better improve my web application.

A copy of the User Test consent form and Google forms survey can be seen in the appendices of this report.

User Testing Results:

I will now present a few example test cases to observe the feedback from the user testing performed.

Test Case ID: 1
Test Priority: High
Test Title: Test Overall Application
Description: Test all function within the web application
Preconditions: Consent form is signed and submitted
Step 1: User navigates to Sole Seeker URL
Step 2: User creates an account
Step 3: User navigates through application testing all functions available
Step 4: User fills out the application review form and submits.
Expected Result: All feedback Is positive; User specifies that all functions are working correctly.
Actual Result: All feedback Is positive. User specifies that all functions are working correctly.
Status: Passed

Test Case ID: 2
Test Priority: High
Test Title: Test Overall Application
Description: Test all function within the web application
Preconditions: Consent form is signed and submitted
Step 1: User navigates to Sole Seeker URL
Step 2: User creates an account
Step 3: User navigates through application testing all functions available
Step 4: User fills out the application review form and submits.
Expected Result: All feedback Is positive; User specifies that all functions are working correctly.
Actual Result: All feedback Is positive. User specifies that all functions are working correctly.
Status: Passed

Test Case ID: 3
Test Priority: High
Test Title: Test Overall Application
Description: Test all function within the web application
Preconditions: Consent form is signed and submitted
Step 1: User navigates to Sole Seeker URL
Step 2: User creates an account
Step 3: User navigates through application testing all functions available
Step 4: User fills out the application review form and submits.
Expected Result: All feedback Is positive; User specifies that all functions are working correctly.
Actual Result: All feedback Is positive. User specifies that all functions are working correctly.
Status: Passed

Test Case ID: 4
Test Priority: High
Test Title: Test Overall Application
Description: Test all function within the web application
Preconditions: Consent form is signed and submitted
Step 1: User navigates to Sole Seeker URL
Step 2: User creates an account
Step 3: User navigates through application testing all functions available
Step 4: User fills out the application review form and submits.
Expected Result: All feedback Is positive. User specifies that all functions are working correctly.
Actual Result: Not all feedback was positive. User specified that there was an internal server error relating to the Sole Searcher page.
Status: Failed

After test case 4 I went investigating the error that has occurred. At the time the error persisted on both my deployed application and local server. The error stated that 'cannot find "results" '. I double checked my code to further progress with mitigating this error but did not find anything wrong. I concluded that the error must have been coming from the web API I was pulling my results from. It seemed as though the API key I was working with no longer worked.

As highlighted in the Implementation section of this report I sent an email to the developer of the API, querying if the API was taken down and his response can be seen below.

Hey Sam,

We transitioned over to require api keys to use our api. I apologize for any inconvenience this may have caused. We attempted to contact all known users of our api via email. However, now that we require an api key we now know who all is using our api and can send announcements with more confidence. Below you can find a link to obtain an api key.

https://rapidapi.com/tg4-solutions-tg4-solutions-default/api/v1-sneakers/pricing

Please let me know if you have any questions or concerns.

Due to this I had to alter my old code to facilitate this new API key. It took a while to try figure out how facilitate this new type of API key into my code. However, I got it back working over a couple days. I also have to pay a monthly fee of 20 dollars to match the amount of API requests my application requires.

This individual test case was extremely beneficial to conduct as it highlighted a pretty major bug within my application.

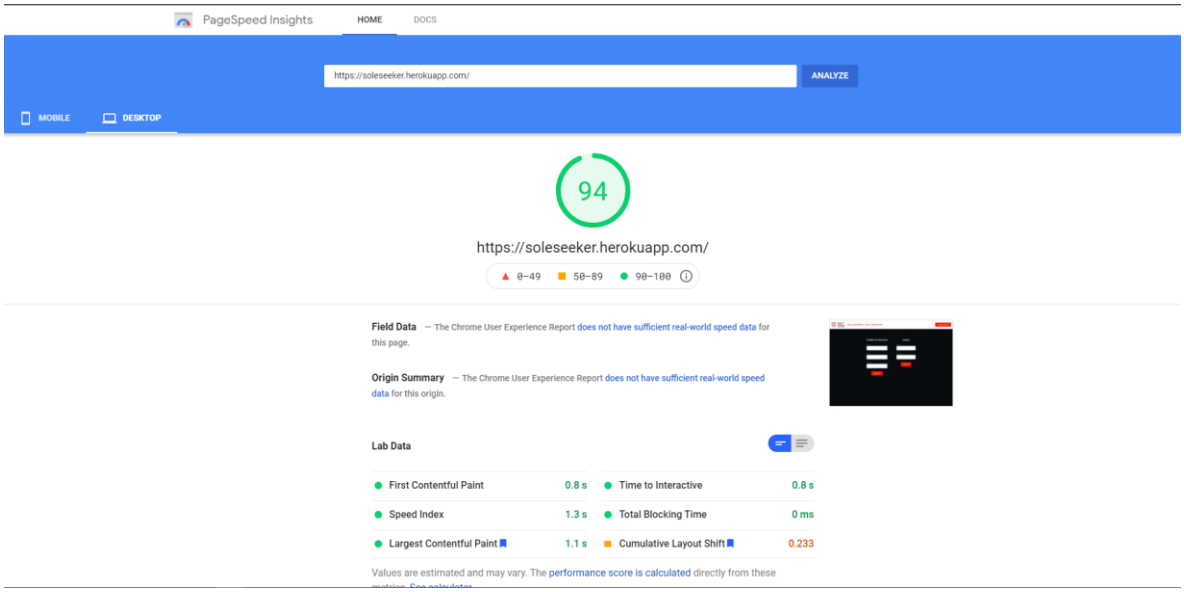
Below are the overall results coming from the User Testing Survey.

Timestamp	How was your overall experience with the Sole Seeker Application?	Were you able to successfully register as a user and log into the website?	How was your experience with the Login and Registration System?	Were you easily able to navigate through each page of the website?	What do you think about the overall design of the website?	You could easily make a post on the forum after being logged into the website?	You could easily use the function on the 'Sole Searcher' page	How was your experience with the 'Shoe Seeker' function?	Did you encounter any bugs during the testing?
2021/04/12 2:29:15 PM GMT+1	Great	Yes	Great	Yes	Good	Strongly agree	Strongly agree	Great	No
2021/04/14 2:54:59 PM GMT+1	Great	Yes	Great	Yes	Great	Strongly agree	Strongly agree	Great	No
2021/04/14 7:39:28 PM GMT+1	Good	Yes	Great	Yes	Great	Strongly agree	Strongly agree	Great	No
2021/04/29 2:13:42 PM GMT+1	Great	Yes	Great	Yes	Great	Strongly agree	Disagree	Great	Yes
2021/05/07 12:12:56 PM GMT+1	Great	Yes	Great	Yes	Great	Strongly agree	Strongly agree	Great	No

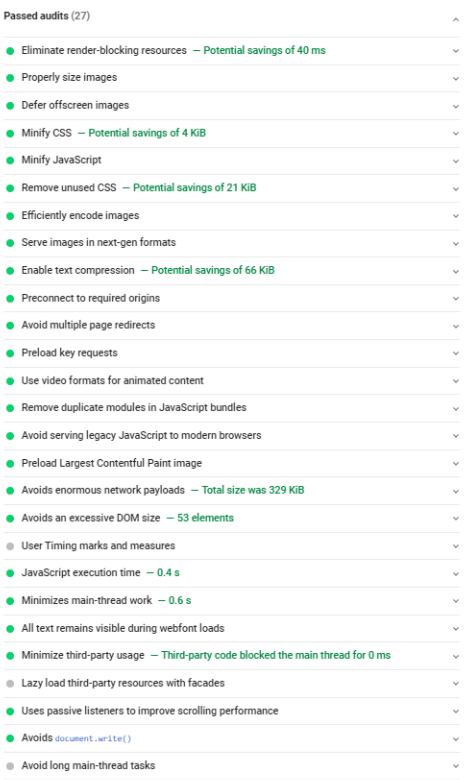
If yes, please explain the bug?	In your own opinion, what brands would you associate this type of application with	Would you recommend this application to a friend?
	Nike, Adidas, Vans etc.	Yes
	YEEZY	Yes
	Adidas	Yes
There was an error related to the server in the sole searcher function.	Nike	Yes
	Nike and Adidas	Yes

2.6 Evaluation

To evaluate my web application, I submitted my deployed website to Page Speed Insights. This service runs a program to test the performance of your application and gives advice on how to improve in specific areas. The program will give a website URL a score out of 100. It's said that a score of 90 to 100 is good, 50 to 90 requires some improvements and below 50 is said to be poor (About PageSpeed Insights | Google Developers, 2021). Below you can see that my application secured a score of 94.



More information can be seen below, listing the different audits my application passed.



3.0 Conclusions

Advantage

Sole Seeker was aimed at providing a place for sneaker enthusiast of all kinds to come together and utilize valuable services. From completion of my application, I feel I have achieved this aim with functions such as Sole Searcher API function and my Forum page.

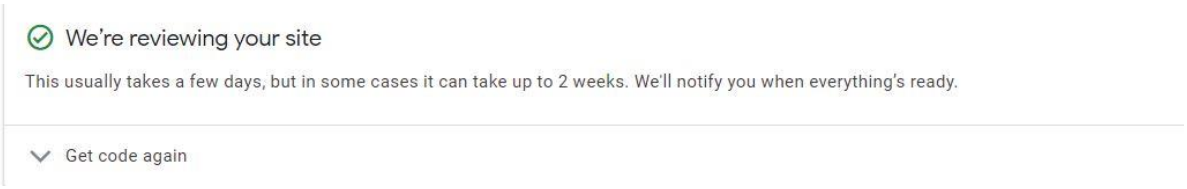
Sole Seeker provides an easy to use experience with great visualizations along with detailed services.

The application highlights various avenues to gain revenue from providing an easy to use application with interesting services. The application has a bright a vibrant theme to attract the user's eye and get them to revisit the site.

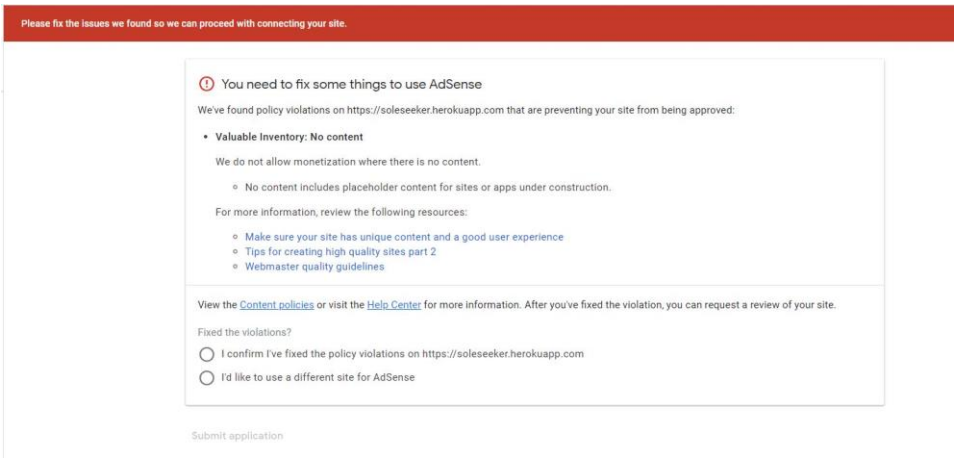
Disadvantage

The disadvantage of Sole Seeker lies around the opportunity to grow our services. Although users can share their opinion on the forum, there's no service to interact with other posts. The forum page is lacking a comment section or like/dislike function. This would hugely improve the quality of the website and broaden its capability.

Another major disadvantage of my application was the failure to include Google AdSense. I submitted my website for approval at the start of April, as you can see below,



However, after a few days I got this reply below.



From research into this error, I found that Google will not put ads on websites in construction or websites without at least 40 to 50 individual posts written onto the website over the period

of several months. As this was the case, I was unable to facilitate this due to my project time frame. Due to this I had to scrap the idea of incorporating Google AdSense into my web application which negatively impacted my idea of integrating different types of revenue streams.

Limitation and opportunities of the project

A large opportunity for this kind of application is the actual selling of sneakers. Incorporating an ecommerce aspect to my application could help gain revenue and reputation. With the already implementation of the PayPal API in my website, I could facilitate this type of service. However, big websites like Schuh or Footlocker limit the chance of this idea becoming a success due to harsh competition.

4.0 Further Development or Research

Just like any application, with more time, comes more improvements. Sole Seeker is no exception. Below are further improvements that can be made in the future.

- **Forum Comment Section.**
  A comment section for user posts on the forum would be beneficial in terms of user connection. Also, a reaction element could be implemented, such as a like/dislike button.
- **Use of QR codes in the giveaway competition.**
  With the growing popularity and development of using QR codes in applications could be considered in the 'Shoe Seeker' service. For example, instead of users trying to find a red and gold box in a certain location, they could find a QR code to be entered into the giveaway. Or even, first to scan the QR code wins.
- **Ecommerce opportunity with the Sneaker API function.**
  This aspect is associated with my Sole Searcher API aspect, whereby users are given 50 sneaker results relating to the criteria they set. A future opportunity could involve the selling of these sneaker results. However, this aspect would involve a large investment and introduce harsh competition.

5.0 References

1. Spacey, J., 2017. *11 Examples of Usability Requirements*. [online] Simplicable. Available at: <https://simplicable.com/new/usability-requirements> [Accessed 5 May 2021].
2. Google Developers. 2021. *About PageSpeed Insights | Google Developers*. [online] Available at: <https://developers.google.com/speed/docs/insights/v5/about> [Accessed 7 May 2021].
3. Herbert, A., 2021. *Pretty Printed*. [online] Prettyprinted.com. Available at: <https://prettyprinted.com/> [Accessed 10 May 2021].
4. MongoDB. 2021. *The most popular database for modern apps*. [online] Available at: <https://www.mongodb.com/> [Accessed 14 May 2021].
5. Flask.palletsprojects.com. 2021. *Welcome to Flask — Flask Documentation (2.0.x)*. [online] Available at: <https://flask.palletsprojects.com/en/2.0.x/> [Accessed 14 May 2021].
6. Mark Otto, a., 2021. *Introduction*. [online] Getbootstrap.com. Available at: <https://getbootstrap.com/docs/5.0/getting-started/introduction/> [Accessed 14 May 2021].
7. Rapidapi.com. 2021. *V1 Sneakers API Documentation (tg4-solutions-tg4-solutions-default) | RapidAPI*. [online] Available at: <https://rapidapi.com/tg4-solutions-tg4-solutions-default/api/v1-sneakers> [Accessed 14 May 2021].
8. Python-visualization.github.io. 2021. *Folium — Folium 0.12.1 documentation*. [online] Available at: <https://python-visualization.github.io/folium/> [Accessed 14 May 2021].
9. Developer.paypal.com. 2021. *PayPal Commerce Platform for Business*. [online] Available at: <https://developer.paypal.com/docs/business/> [Accessed 14 May 2021].

6.0 Appendices

6.1 Project Proposal

National College of Ireland

Project Proposal

Sole Seeker

31/10/2020

Software Project

Business Information Systems

2020/2021

Sam Greenan

X17449342

X17449342@student.ncirl.ie

Objectives

The aim of this project is to develop and create a web application for users who are in the market for a new pair of shoes/sneakers. The user will be able to design their own specific shoe on my web application. The user will then be presented will five pairs of shoes that already exist on the market that are like the users' design. The list of shoes will vary from, most like the designed shoe, to least like the designed shoe. Each pair of shoes presented on the web page will have information alongside with several links to purchase that specific pair.

My main function I want to create is the shoe similarity algorithm. I hope to develop this using Python and present it on the Home page of my website. I plan on doing much more research into algorithms and improve my effectiveness with the language Python. I plan on completing the Python crash course I am currently progressing through.

Another major factor I want to incorporate into my web application is creating a central hub, where the sneaker community can come together to share opinions on sneakers and sneaker trends. Over the last number of years, the interest in designer and collectable sneakers has grown immensely, so I believe there is a missive market for this type of application.

I would also like to research into the business side of my application. I want to find ways of commercialising my idea to gain capital. Some ideas I have in mind include, specific links for websites to purchase sneakers. These specific links will notify a sneaker business that the user has come from my website. This way I can receive commission if one of my users purchases a shoe on their site. I would also like to investigate incorporating this idea into an in-store application. For example, Footlocker or Schuh having a tablet that the shopper can use to customize a pair of shoes. Once customized, it will present the user with shoes that resemble their chosen design that are in stock at that time. Furthermore, I want to explore Google AdSense and how that could possibly be included in my website.

To build my web application. I must use the languages Python, HTML, CSS with the framework Flask. I will also populate any relevant information into a MongoDB database. To retrieve the shoe data, I will use a webAPI or conduct Web scraping.

Main Objectives to be completed:

- Build a user-friendly web application.
- Retrieve and store relevant user information.
- Develop a shoe customization template for user input.
- Create the algorithm to compare the users input with the database of shoes.
- Create other services for shoe lovers on my web application, for e.g. mini games, blog posts, forum page.
- Build a secure and detailed Mongo-DB database.

Background

I came up with this idea before I ever started my college course in NCI. The idea came to me when I wanted to purchase a pair of sneakers. I could visualize the pair I wanted but I couldn't find any like it online or in the local shopping centre. Later, I went online a visited a shoe customization website for Nike sneakers. I customized my own pair and was about to purchase them when a thought came into my head. **What if these already exist?**

That's when I came up with my idea. A way in which a user can customize any pair of shoes, and then be presented with similar shoes that already exist on the market. The similarities would be based on brand, type, and colour.

Also, over the years I have become more and more interested in designer and collectable sneakers. I love talking about them with my friends, so I thought an application relating to this domain would be very enjoyable to develop. I'm aware of the massive community of and the need for this type of application.

In terms of similar software's out there now, there is many different websites that allow users to find their own preferred specified sneakers. However, there isn't a website that would output a 'Top Five' system. I want to make my website a hub for all sneaker lovers.

It is when I started to flesh out my chosen idea, I decided on going down the route of web application. I thought that this would be the best approach for my idea. I have always been very much interested in the area, doing well in my web application and web design modules I knew I was more than capable to develop a well-designed website. My decision to use the framework Flask was through my interest in the backend language, Python. I started learning Python over the summer in preparation for my final year project. I have built multiple mini programs in the language and I am currently enrolled in a python Udemy course.

Technical Approach

To build my application, I must have a structured technical approach. This will set a pathway for the next few months in developing my idea.

Research

Firstly, I plan on researching more about my chosen idea. This will involve looking for web applications that are similar to mine. This way I can see more about my future competitors, gather ideas, and understand what is necessary for a live web application.

Other types of research that I will conduct will be technical. I plan on completing two Udemy courses, one for Rest-API and flask and another for Python. On completion of these courses, I will be able to develop my idea into a live Web application. I will also adhere to the recent trends in web application security and research ways of adapting this into my project. I will also dig deep into cloud-based deployment methods.

Wireframes

To understand what my website will look like; I will create multiple wireframes. A wireframe gives the developer a way of conceptualizing their idea. By using a detailed wireframe software, I can plan what my web page will look like. I hope to have a wireframe for each web page completed for my midpoint Implementation.

Use Case

To map out every function on my website, I will conduct multiple use cases. I will use the website Draw.io to develop all my Use Cases. Every Use Case will be accompanied with a detailed Use Case description, highlighting every aspect of that function. I will create a Use case for every function within my website.

Development

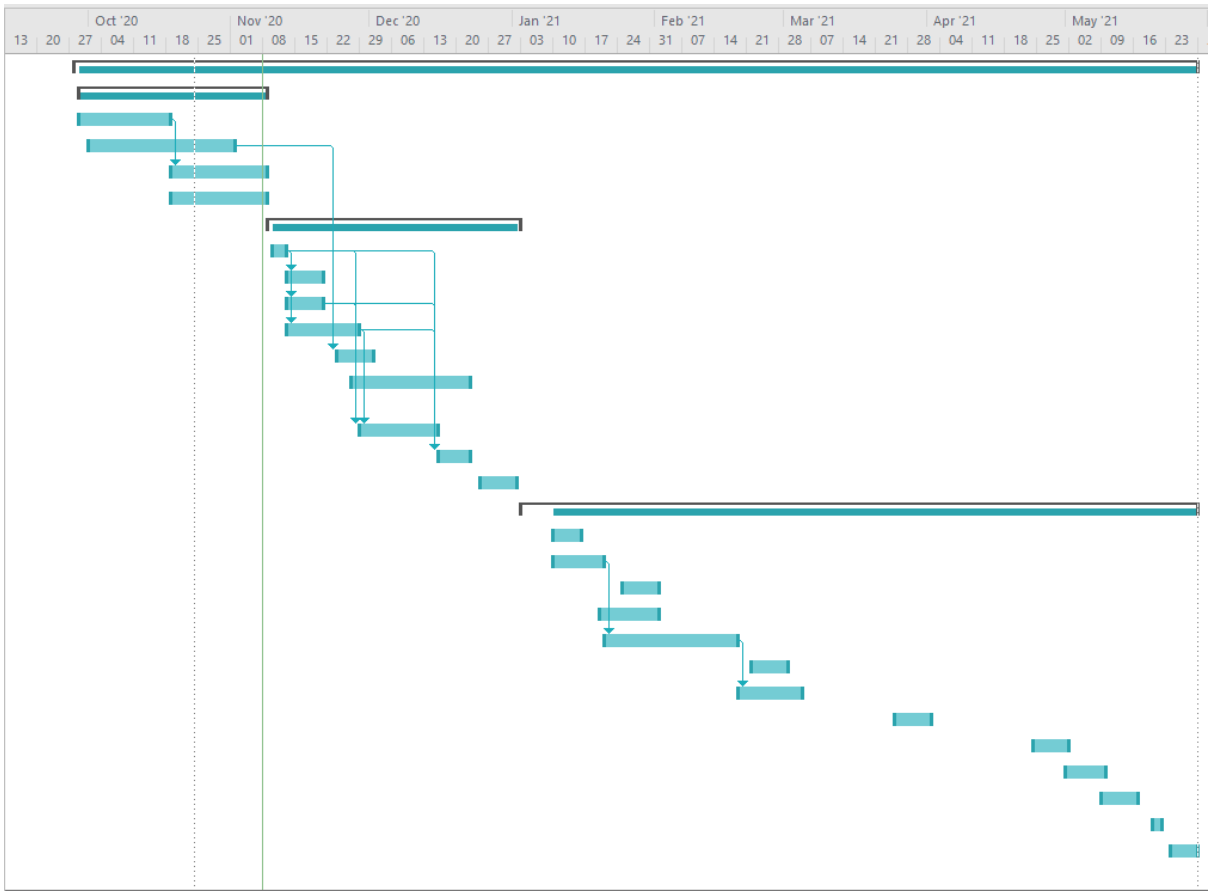
For my development process, I will be using the IDE Visual studio code. I will also be using and manipulating a webAPI and configuring a MongoDB Database. The development process will be most work intensive, so I must ensure all planning beforehand must be completed to the highest standards.

Implementation and deployment

On completion of my software project, I will have my web application hosted on the web. By the midpoint implementation, I hope to have my prototype hosted on a cloud-based server. I am considering using Heroku to support this.

Project Plan

Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Software Project	**244 days?**	**Mon 28/09/20**	**Sat 29/05/21**	
	Pre-development	**41 days**	**Tue 29/09/20**	**Sun 08/11/20**	
	Project Pitch	20 days	Tue 29/09/20	Sun 18/10/20	
	October Reflective Journal	32 days	Thu 01/10/20	Sun 01/11/20	
	Project Proposal	21 days	Mon 19/10/20	Sun 08/11/20	3
	Project Ethics Form	21 days	Mon 19/10/20	Sun 08/11/20	
	Mid Point Implementation	**55 days?**	**Mon 09/11/20**	**Sat 02/01/21**	
	Wireframe	3 days	Tue 10/11/20	Thu 12/11/20	
	Research	8 days	Fri 13/11/20	Fri 20/11/20	8
	Use Case and UML Drafted	8 days	Fri 13/11/20	Fri 20/11/20	8
	Website Navigation and Nav bar Created	16 days	Fri 13/11/20	Sat 28/11/20	8
	November Reflective Journal	8 days	Tue 24/11/20	Tue 01/12/20	4
	Prototype work (Login Page, static pages and cloud deployment)	26 days	Fri 27/11/20	Tue 22/12/20	
	Documentation	17 days	Sun 29/11/20	Tue 15/12/20	8,10,11
	Mid point video presentation Prep	7 days	Wed 16/12/20	Tue 22/12/20	8,10,11
	December Reflective Journal	8 days	Fri 25/12/20	Fri 01/01/21	
	Final Implementation	**147 days?**	**Sun 03/01/21**	**Sat 29/05/21**	
	Ensure the Login Functionality is secure	6 days	Sun 10/01/21	Fri 15/01/21	
	API and Databse Connection	11 days	Sun 10/01/21	Wed 20/01/21	
	January Reflective Journnal	8 days	Mon 25/01/21	Mon 01/02/21	
	Taking user Input	13 days	Wed 20/01/21	Mon 01/02/21	
	Create Matching algoithm	29 days	Thu 21/01/21	Thu 18/02/21	19
	February Reflective Journnal	8 days	Mon 22/02/21	Mon 01/03/21	
	Forums/Blog Pages	14 days	Fri 19/02/21	Thu 04/03/21	22
	March Reflective Journnal	8 days	Thu 25/03/21	Thu 01/04/21	
	April Reflective Journnal	8 days	Sat 24/04/21	Sat 01/05/21	
	Final Touches	9 days	Sat 01/05/21	Sun 09/05/21	
	Video Presentation	8 days	Sun 09/05/21	Sun 16/05/21	
	VIVA Examination	2 days	Thu 20/05/21	Fri 21/05/21	
	Project Showcase	6 days	Mon 24/05/21	Sat 29/05/21	


Oct '20
Nov '20
Dec '20
Jan '21
Feb '21
Mar '21
Apr '21
May '21
13 20 27 04 11 18 25 01 08 15 22 29 06 13 20 27 03 10 17 24 31 07 14 21 28 07 14 21 28 04 11 18 25 02 09 16 23

Technical Details

The technologies that will be used in this application will be based around the necessary languages and frameworks used in web application. The technologies that I plan on using are Visual studio code, MongoDB, HTML, CSS, bootstrap, Flask and Python. I will also be using some sort of webAPI and web scraping.

Visual Studio Code

The main IDE that I plan on using is Visual studio code. Visual studio code is a free source for developing all kind of projects. It is an extremely useful and intelligent IDE that allows for precisive debugging and provides a user-friendly display. I have used this IDE in practice, and I find it very easy to use. It offers a wide range of capabilities.

MongoDB

MongoDB is a NoSQL database storage system that holds data using documents and key-value pairings. I will use this type of database to hold key user information, for example their login details. Yet, I haven't used this type of technology before but am familiar with the concept.

HTML

HTML is the mark-up language that allows users to display data on the web browser. I have used this language on multiple projects, and I am quite confident in my ability.

CSS

CSS is a cascading styling sheet. This revolves around the design of the web page. This language will let me provide the user with a well-designed user interface and eye-grabbing web pages. I have used this technology many of times before and am quite familiar with the syntax.

Bootstrap

Bootstrap is a front-end web framework. It provides web templates for JavaScript, CSS, and HTML. I will be using these templates for my blog and user forum web pages on my web application. I have never used bootstrap before; however, I am excited to get started with it.

Flask

The back-end web framework that I will be using is Flask. Flask utilizes the back-end language Python. Its main competitor would be Django. I considered using Django, however due to my project idea, I felt as though Flask would accommodate my idea better. I have built some small programs using Flask, though I am still at beginner level.

Python

Python is a high-level programming language. Python can be used in a wide range of varieties. For my project I will be using it as the backend language for my web application. I started learning this language over the summer using an Online Udemy course and different YouTube tutorials. I find the language extremely useful and admire the emersed capabilities. Yet, I have

not built a large full-scale Python project. Due to this, my project will be difficult to develop, but I am up for the challenge.

Web API / Web Scraping

I will use an API or conduct some sort of web scraping to gather the data to be displayed on the webpage. I will be using a sneaker API to match with the user input. I have researched and found multiple Sneaker API's/databases. The next step is to choose the most suitable resource.

Evaluation

Upon completion of my final year project, I hope to complete many different types of testing. Testing is essential with every type of software development project.

Some of these testing techniques include user testing, functionality testing, UI testing, usability testing and security testing. I will definitely prioritize user testing. This type of test will give me feedback and help me improve in areas that are lacking and maintain areas that are striving. At different points in the development process of my application, I will conduct multiple User testing's. These would involve gathering opinions and advice for my web application. Securing advice on where to improve will be extremely helpful.

Another example of testing I will conduct is Netsparker. This software solution tool can help find security threats in a web application. It could help me review my security overview and ways of improving it. The software presents its results with multiple figures and helpful charts.

Invention Disclosure Form

Please fill in the following sections, if you think your idea is innovative:

1. Title of Invention

Sole Seeker

2. Inventors

Name	School/Research Institute	Affiliation with Institute (i.e. department, student, staff, visitor)	Address, contact phone no., e-mail	% Contribution to the Invention
Sam Greenan	National College of Ireland	Student	7 Old Yellow Walls Road, Malahide, Co. Dublin	100

3. Contribution to the Invention

Each contributor/potential inventor should write a paragraph relating to his/her contribution and include a signature and date at the end of the paragraph.

Sam Greenan – I am the only contributor to this project. I came up with the project idea and I am the only one who will develop the project. Signature:  Date: 08/11/2020

4. Description of Invention

(Please highlight the novelty/patentable aspect. Attach extra sheets if necessary including diagrams where appropriate). What is novel, the 'inventive step'? For more information on patents, please look at http://www.patentsoffice.ie/en/patents.aspx

My invention is a web-based application surrounding sneakers. My application allows users to fully customize any type of shoes. Once the user submits their shoe, they will be presented with five pairs of shoes that are like the users input. These shoes are available on the market and will be accompanied with a link to purchase them. The idea is to stop the user from paying a high customization fee on another site.

5. Why is this invention more advantageous than present technology?

What is its novel or unusual features? What problems does it solve? What are the problems associated with these technologies, products or processes? Explain how this invention overcomes these problems (*i.e*. what are its advantages).

The problem that my application is trying to address is to stop users from buying overpriced customizable shoes when their customized shoes could already exist somewhere on the market. I also want to create a hub for all sneaker lovers to share their opinions and learn from other like-minded individuals in the sneaker community. My website will notify all users the retail price on hundreds of pairs of shoes, so that they never over-pay for a certain pair.

6. What is the current stage of development / testing of the invention?

I am currently in the planning process of development.

7. List the names of companies which you think would be interested in using, developing or marketing this invention

Schuh, Footlocker, JD Sports, Intersport Elverys, Lifestyle Sports, Nike, Adidas, Lacoste.

8. Funding Partner(s)

Government Agency & Department	N/A
% Support	N/A
Contract/Grant No.	N/A
Contact Name	N/A
Phone No.	N/A
Address	N/A

Industry or other Sponsor	N/A
% Support	N/A
Contract/Grant No.	N/A
Contact Name	N/A
Phone No.	N/A
Address	N/A

9. Where was the research carried out?

Research was carried out on multiple websites, include Brooks, Nike, and Footlocker. Research for technologies was conducted on Udemy and YouTube.

10. What is the potential commercial application of this invention?

The potential commercialization of this project is Google AdSense or commission-based revenue. Commission can come from linking shoes on my website to specific websites.

11. Was there transfer of any materials/information to or from other institutions regarding this invention?

If so please give details and provide signed agreements where relevant.

N/A

12. Have any third parties any rights to this invention?

If yes, give names and addresses and a brief explanation of involvement.

N/A

13. Are there any existing or planned disclosures regarding this invention?

Please give details.

N/A

14. Has any patent application been made? Yes/No

If yes, give date: _____N/A_____ Application No.: _____N/A_____

Name of patent agent: _____N/A_____

Please supply copy of specification.

15. Is a model or prototype available? Has the invention been demonstrated practically?

There is no prototype available now.

I/we acknowledge that I/we have read, understood and agree with this form and the Institute's *Intellectual Property and Procedures* and that all the information provided in this disclosure is complete and correct.

I/we shall take all reasonable precautions to protect the integrity and confidentiality of the IP in question.

Inventor: Sam Greenan

Signature

Date: 08/11/2020

_____ Signature

6.2 Ethics Approval Application

National College of Ireland

Ethical Guidelines and Procedures for Research involving Human Participants

SEPTEMBER 2017

1. Introduction

All research involving human participants that is conducted by students or staff at the National College of Ireland should be done so in an ethical manner. The college has therefore developed an Ethics Committee, which acts as a sub-committee of the Research Committee, to ensure that ethical principles pertaining to research involving human participants are upheld and adhered to. All researchers intending to use human participants as part of their projects are thus required to reflect upon any potential ethical issues and submit their research proposals for ethical review before commencing data collection.

This document gives an overview of the core ethical principles guiding research in NCI, while also documenting the procedures required for seeking ethical approval of research involving human participants.

Am I conducting research?

Research is defined as "the attempt to derive generalisable new knowledge by addressing clearly-defined questions with systematic and rigorous methods" (NHS Health Research Authority). Sometimes, we collect data in order to evaluate a service or practice we are engaged in ("service evaluation"). The main difference between research and service evaluation is in the aim: research is trying to create new generalisable knowledge, and service evaluation is trying to evaluate whether a delivered service/practice is working well. One project may have both aims included in it. It can be confusing if a service or intervention is involved, whether or not research is being conducted. If new or competing interventions are being evaluated, then it is likely to be research, whereas if an existing service is being conducted anyway, with an evaluative component, then it is likely to be a service evaluation. Research requires consideration of the below guiding principles, whereas service evaluation does not require approval from an ethics committee.

2. Guiding Principles

In line with other research institutions, there are three core guiding principles governing the ethical conductance of research involving human participants at NCI. These principles stem from the *Belmont Report* (1979) published by the National Commission for the Protection of Human Subjects of Biomedical and Behavioural Research. While it is recognised that these principles may be operationalised differently depending on the specific research discipline, it is recommended that these are consulted as a starting point for any research involving human participants.

2.1 Principle 1: Respect for Persons

This principle entails recognition that participants should be treated as autonomous individuals and hence should never be coerced or swayed into participating in a research project against their will. The participant's right to withdraw from a research study at any time should be respected, as well as their right to dignity and protection from harm.

Respect for individuals can often be implemented in practice via the process of informed consent, whereby potential participants are made fully aware of the requirements involved in participation. While it is recognised that in certain cases deception (i.e. the withholding of certain information from participants) may take place, this should only occur when it is robustly justified for the validity of the research. In cases where deception is justified, researchers should ensure that any potential risk

resulting from this measure is minimised. Participants should also be fully debriefed on the nature of the research after it has taken place.

The principle of respect also requires researchers to protect individuals from vulnerable groups who may have diminished autonomy (see section 4.2 for more detail as to what constitutes vulnerable groups). Where full informed consent is not possible for such population groups, consent may instead be sought from their guardians. In all cases however clear assent, or willingness to participate, should be demonstrated from participants.

2.2 Principle 2: Beneficence and non-maleficence

This principle specifically focuses on the need to protect the well-being of participants. Any potential risk to participants should be minimised, whether that be risk of physical discomfort or of any psychological, emotional or social distress, while possible benefits should be maximised. Researchers adhering to this principle should thus ensure that any potential benefits derived from carrying out the study (e.g. in terms of knowledge gained) should outweigh potential risks. Even in cases where there is only a slight potential risk of harm, participants should be provided with appropriate support to alleviate this.

2.3 Principle 3: Justice

This principle emphasises the need to employ fairness in the distribution of benefits and risks to participants. The way in which participants are selected to take part in research should relate to the purpose of the study, as opposed to other factors such as availability or manipulability of participants. The exploitation of vulnerable populations should be avoided.

Where applicable, researchers are encouraged to consult guidelines stemming from their own professional bodies (e.g. The Psychological Society of Ireland) in addition to the general guiding principles above when planning their research. Researchers should also be sensitive to those issues which are specific to the population under investigation and the methodology that is employed in the project (e.g. qualitative methodologies involving the recording of data may raise issues relating to participants' right to anonymity, as well as the ethical management and use of data). Detailed consideration should be given to all these issues when planning research and when completing the Ethical Review Application form.

3. Ethics Committee

The NCI Ethics Committee was established by the Academic Council in 2012. Acting as a sub-committee to the Research Committee, its role is to oversee ethical issues arising from all research involving human participants that is conducted by students and staff of the college. The key purpose of this committee is to safeguard against any potential harm to participants, and to ensure that their rights are recognised in line with the guiding principles outlined above.

The Ethics Committee reviews all research proposals posing ethical risk to the participants involved, however the decision as to whether projects pose ethical risk is firstly made via the appropriate Filter Committee which operates at School level (see organisational structure in Figure 1 below). The Filter

Committees may review and approve research proposals which are of low ethical risk, while referring those of high ethical risk to be considered by the Ethics Committee (see categories of ethical risk in section 4.1).

While the Filter Committees are made up of staff members with subject-specific knowledge, membership of the Ethics Committee should comprise of no less than five representatives from both the School of Computing and the School of Business, including representatives from the Research Committee.

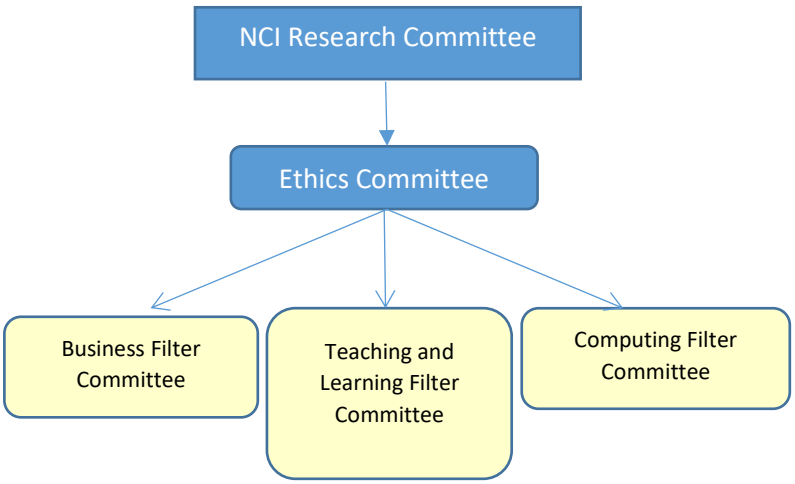


Figure 1: Committee Structures.

4. Review Process

Any staff or student of NCI wishing to conduct a study involving human participants should first submit the Ethical Review Application Form (included at the end of this document), to the relevant School Filter Committee at proposal stage. This initial review will result in a graded categorisation of ethical risk, as outlined below.

4.1 Categorisation of Ethical Risk

Research category A
Research in this category poses little ethical risk to the participants involved. Specifically, it refers to research involving human volunteers, but **excluding** studies involving:

- therapeutic interventions
- new research methodologies
- vulnerable populations (see section 4.2)
- deception of the participants
- any other significant physical, social or psychological risk to participants

Research category B
Research in this category involves human volunteers **including** studies involving:

- therapeutic interventions
- new research methodologies

- vulnerable populations (see section 4.2)
- deception of the participants
- any potentially significant risk to participants

Research Category C
This specifically refers to research involving human volunteers who are service users, patients, staff, records, etc., within the sphere of the HSE or similar setting (but not including clinical trials of investigative medicinal products).

4.2 Vulnerable groups

There are a number of participant populations that may fall under the heading of 'vulnerable groups'. These groups require consideration of unique ethical challenges regardless of the nature of the project. Research involving such populations should therefore always be reviewed by the Ethics Committee.

Groups that may be classed as vulnerable include, but are not limited to:

- Children (under 18 years of age)
- The older old (aged 85+)
- People with an intellectual or learning disability
- Individuals or groups receiving help through the voluntary sector
- Those in a subordinate position to the researcher (e.g. employees)
- Any other groups who might not understand the research and consent process

Note: in addition to the Ethical Review process, any researchers intending to work directly with children will be required to undergo Garda Vetting in advance of the proposed research.

4.3 Exemption from Full Ethical Review

In certain limited cases, researchers can apply for an exemption from full ethical review. In such cases, the Ethical Review Exemption form should be completed, explicitly detailing why the exemption is sought.

In completing this form, researchers must declare that the research does not involve any of the following:

- Vulnerable groups
- Sensitive topics
- Risk of psychological or mental distress
- Risk of physical stress or discomfort
- Any other risk to participants
- Use of drugs or invasive procedures (e.g. blood sampling)
- Deception or withholding of information from participants
- Conflict of interest issues
- Access to data by individuals or organisations other than the researchers

- Any other ethical dilemmas

4.4 Outcomes of Review Process

Following consideration of research projects submitted for Ethical Review, each Filter Committee will submit a report to the Ethics Committee summarising the applications considered and the decisions made.

For research that is deemed to fall under Research Category A (low ethical risk), a favourable outcome at the relevant Filter Committee will be sufficient to secure ethical approval. Research falling under the other two categories must however be considered by the Ethics Committee before approval may be granted.

On the basis of this review, four key outcomes may arise:

1. Research proposal approved (no recommendations)
2. Research proposal approved pending minor revisions (to be accepted by the Chair and Research Supervisor)
3. Research proposal approved pending major revisions (to be resubmitted and approved by the Ethics Committee)
4. Research proposal rejected (resubmission necessary)

A summary of the processes involved in applying for ethical approval can be seen in Figure 2.

Appeals

Appeals against the Committee's decision may be made within ten working days. In this case, at least three members of the Ethics Committee, none of whom will have reviewed the initial application, may review this along with any additional information submitted by the applicant.

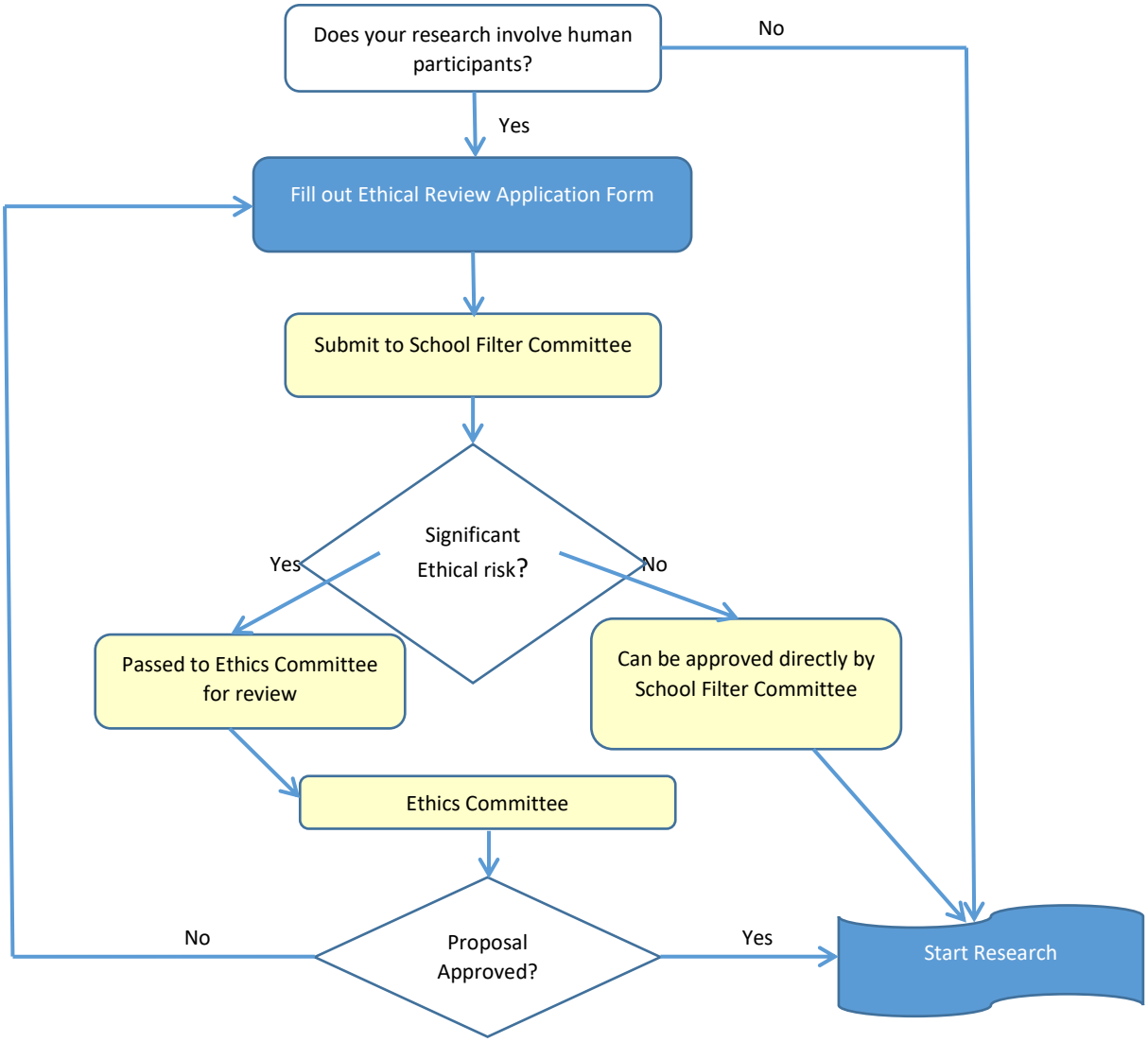


Figure 2: Process chart for seeking Ethical Approval

Ethics Application Checklist

To be submitted alongside the ethics application.

Please complete the below checklist, ticking each item to confirm that it has been addressed.

1. I agree to obtain informed written consent from all human participants aged over 18 who are involved in this research (or if circulating digitally, I will ensure that informed consent is completed, and will have the participants indicate their informed consent by continuing with their study engagement).	□ □
2. I agree to obtain informed written consent from the parents of anyone aged under 18 in this research (or from the schools if appropriate), and informed written assent from those under 18 in this research.	
3. I include a letter of agreement from a clinically responsible individual agreeing to (where appropriate) help me recruit/provide clinical support in the event that participants become distressed/host the study data collection.	□
4. I append a letter of agreement from an external institution or organisation agreeing to host the study.	□
5. I agree to comply with NCI's Data Retention Policy.	
6. I have appended a) information sheet, b) consent form/assent form, c) debriefing sheet.	□
7. I have provided details of how non-anonymised data will be stored, in a safe and encrypted manner.	□
8. I have included my contact details and those of my supervisor (where appropriate). I have only included my NCI email address and not included any personal contact information.	□ □
9. I have given sufficient details on the proposed study design, methodology, and data collection procedures, to allow a full ethical review, and I understand that my failure to give sufficient detail may result in a resubmission being required.	□
10. I understand that if I make changes to my study following ethical approval, it is my responsibility to seek an ethics amendment if the change merits ethical consideration.	□ □

National College of Ireland

Human Participants Ethical Review Application Form

All parts of the below form must be completed. However in certain cases where sections are not relevant to the proposed study, clearly mark NA in the box provided.

Part A: Title of Project and Contact Information

Name

Sam Greenan

Student Number (if applicable)

X17449342

Email

X17449342@student.ncirl.ie

Status:

Undergraduate ☐
Postgraduate ☐
Staff ☐

Supervisor (if applicable)

Sean Heeney

Title of Research Project

Sole Seeker

Category into which the proposed research falls (see guidelines)

Research Category A ☐
Research Category B ☐
Research Category C ☐

Have you read the NCI Ethical Guidelines for Research with Human Participants?

Yes ☐
No ☐

Please indicate any other ethical guidelines or codes of conduct you have consulted

NA

Has this research been submitted to any other research ethics committee?

Yes ☐
No ☐

If yes please provide details, and the outcomes of this process, if applicable:

NA

Is this research supported by any form of research funding?

Yes □
No □

If yes please provide details, and indicate whether any restrictions exist on the freedom of the researcher to publish the results:

NA

Part B: Research Proposal

Briefly outline the following information (not more than 200 words in any section).

Proposed starting date and duration of project

28/09/2020 – 09/05/2020

The rationale for the project

Final Year Project

The research aims and objectives

To gain user recommendations and opinions for a software application.

The research design

To allow users to test my application and receive feedback from their experiences.

The research sample and sample size
Please indicate the sample size and your justification of this sample size. Describe the age range of participants, and whether they belong to medical groups (those currently receiving medical treatment, those not in remission from previous medical treatment, those recruited because of a previous medical condition, healthy controls recruited for a medical study) or clinical groups (those undergoing non-medical treatment such as counselling, psychoanalysis, in treatment centres, rehabilitation centres, or similar, or those with a DSM disorder diagnosis).

The sample size will be no more than 10 users, all 18+

If the study involves a MEDICAL or CLINICAL group, the following details are required:

a) **Do you have approval from a hospital/medical/specialist ethics committee?**
If YES, please append the letter of approval. Also required is a letter from a clinically responsible authority at the host institution, supporting the study, detailing the support

mechanisms in place for individuals who may become distressed as a result of participating in the study, and the potential risk to participants.
If NO, please detail why this approval cannot or has not been saught.

b) Does the study impact on participant's medical condition, wellbeing, or health?
If YES, please append a letter of approval from a specialist ethics committee.
If NO, please give a detailed explanation about why you do not expect there to be an impact on medical condition, wellbeing, or health.

The nature of any proposed pilot study. Pilot studies are usually required if a) a new intervention is being used, b) a new questionnaire, scale or item is being used, or c) established interventions or questionnaires, scales or items are being used on a new population. If no such study is planned, explain why it is not necessary.

NA

The methods of data analysis. Give details here of the analytic process (e.g. the statistical procedures planned if quantitative, and the approach taken if qualitative. It is not sufficient to name the software to be used).

NA

Study Procedure
Please give as detailed an account as possible of a participant's likely experience in engaging with the study, from point of first learning about the study, to study completion. State how long project participation is likely to take, and whether participants will be offered breaks. Please attach all questionnaires, interview schedules, scales, surveys, and demographic questions, etc. in the Appendix.

Step 1: The user will agree to participate by signing the consent form. Step 2: The user will receive a link to access the web application or test locally on a development server. Step 3: The user will browse the website how they please. Step 4: The user will fill out the User testing survey and submit. Total Time: 15 minutes Application Testing: 10 Minutes Survey: 5 Minutes

Part C: Ethical Risk

Please identify any ethical issues or risks of harm or distress which may arise during the proposed research, and how you will address this risk. Here you need to consider the potential for physical risk, social risk (i.e. loss of social status, privacy, or reputation), outside of that expected in everyday life, and whether the participant is likely to feel distress as a result of taking part in the study. Debriefing sheets must be included in the appendix if required. These should detail the participant's right to withdraw from the study, the statutory limits upon confidentiality, and the obligations of the researcher in relation to Freedom of Information legislation. Debriefing sheets should also include details of helplines and avenues for receiving support in the event that participants become distressed as a result of their involvement in this study.

NA

Do the participants belong to any of the following vulnerable groups?
(Please tick all those involved).

- □ Children;
- □ The older old (85+)
- □ People with an intellectual or learning disability
- □ Individuals or groups receiving help through the voluntary sector
- □ Those in a subordinate position to the researchers such as employees
- □ Other groups who might not understand the research and consent process
- □ Other vulnerable groups

How will the research participants in this study be selected, approached and recruited? From where will participants be recruited? If recruiting via an institution or organisation other than NCI please attach a letter of agreement from the host institution agreeing to host the study and circulate recruitment advertisements/email etc.

The participants will be friends, family and other NCI students.

What inclusion or exclusion criteria will be used?

NA

How will participants be informed of the nature of the study and participation?

In person or by Email.

Does the study involve deception or the withholding of information? If so, provide justification for this decision.

NA

What procedures will be used to document the participants' consent to participate?

JOT Consent Form

Can study participants withdraw at any time without penalty? If so, how will this be communicated to participants?

NA

If vulnerable groups are participating, what special arrangements will be made to deal with issues of informed consent/assent?

NA

Please include copies of any information letters, debriefing sheets, and consent forms with the application.

Part D: Confidentiality and Data Protection

Please indicate the form in which the data will be collected.

□ Identified □ Potentially Identifiable □ **De-Identified**

What arrangements are in place to ensure that the identity of participants is protected?

Particpants only give their name in the consent form, this data will be deleted after use.

Will any information about illegal behaviours be collected as part of the research process? If so, detail your consideration of how this information will be treated.

NA

Please indicate any recording devices being used to collect data (e.g. audio/video).

NA

Please describe the procedures for securing specific permission for the use of these recording devices in advance.

NA

Please indicate the form in which the data will be stored.

□ Identified □ Potentially Identifiable □ **De-Identified**

Who will have responsibility for the data generated by the research?

NA

Is there a possibility that the data will be archived for secondary data analysis? If so, has this been included in the informed consent process? Also include information on how and where the data will be stored for secondary analytic purposes.

NA

If not to be stored for secondary data analysis, will the data be stored for 5 years and then destroyed, in accordance with NCI policy?

□ Yes □ No

Dissemination and Reporting

Please describe how the participants will be informed of dissemination and reporting (e.g. submission for examination, reporting, publications, presentations)?

By email or in-person

If any dissemination entails the use of audio, video and/or photographic records (including direct quotes), please describe how participants will be informed of this in advance.

NA

Part E: Signed Declaration

I confirm that I have read the NCI Ethical Guidelines for Research with Human Participants, and agree to abide by them in conducting this research. I also confirm that the information provided on this form is correct (Electronic signature is acceptable).

Signature of Applicant Sam Greene

Date 10/04/2021

Signature of Supervisor (where appropriate): SHeeney

Date 10/04/2021

Any other information the committee should be aware of?

NA

National College of Ireland

DECLARATION OF ETHICS CONSIDERATION

School of Computing

Student Name: Sam Greenan

Student ID: X17449342

Programme Business Information Systems **Year:** 2017 - 2021

Module: Software Project

Project Title: Sole Seeker

Please circle (or highlight) as appropriate

This project involves human participants	Yes / No

Introduction

Secondary data refers to data that is collected by someone other than the current researcher. Common sources of secondary data for social science include censuses, information collected by government departments, organizational records and data originally collected for other research purposes. Primary data, by contrast, is collected by the investigator conducting the research.

A project that does not involve human participants requires ONLY completion of Declaration of Ethics Consideration Form and submission of the form on module's Moodle page

A project that involves human participants requires ethical clearance and an Ethics Application Form must be submitted through the module's Moodle page. Please refer to and ensure compliance with the ethical principles stated in NCI Ethics Form available on the Moodle page.

The following decision table will assist you in deciding if you have to complete the Declaration of Ethics Consideration Form or/and the Ethics Application Form.

Public Data	Y	Y	Y	Y	N	N	N	N
Private Data	Y	Y	N	N	Y	Y	N	N

Human Participants	Y	N	Y	N	Y	N	Y	N
Declaration of Ethics Consideration Form	x	X	x	X	X	X	x	
Ethics Application Form	X		X		X		X	

Please circle (or highlight) as appropriate

The project makes use of secondary dataset(s) created by the researcher	Yes / No
The project makes use of public secondary dataset(s)	Yes / No
The project makes use of non-public secondary dataset(s)	Yes / No
Approval letter from non-public secondary dataset(s) owner received	Yes / No

Sources of Data:

It is students' responsibility to ensure that they have the correct permissions/authorizations to use any data in a study. Projects that make use of data that does not have authorization to be used, will not be graded for that portion of the study that makes use of such data.

Public Data

A project that makes use of public secondary dataset(s) ***does not need ethics permission****, but* ***needs a letter/email from the copyright holder*** *regarding potential use.*

Some websites and data sources allow their data sets to be used under certain conditions. In these cases, a letter/email from the copyright holder is NOT necessary, but the researcher should cite the source of this permission and indicate under what conditions the data are allowed to be used. See Appendix I for examples of permissions granted by Fingal Open Data, and Eurostat website.

Where websites or data sources indicate that they do not grant permission for data to be used, you will still need a letter/email from the copyright holder. For example, see Appendix II for an example from the Journal of Statistics Education.

Private Data

A project that makes use of non-public (private) secondary dataset(s) must receive data usage permission from School of Computing.

An approval letter/email from the owner *(e.g. institution, company, etc.)* ***of the non-public secondary dataset must be attached to the Declaration of Ethics Consideration****. The letter/email must confirm that the dataset is anonymised and permission for data processing, analysis and public dissemination is granted.*

Evidence for use of secondary dataset(s)

Include dataset(s) owner letter/email or cite the source for usage permission

NA

CHECKLIST

Non-public/private secondary dataset(s) -Owner letter/email is attached to this form ***OR*** Citation and link to the web site where permission is granted – provided in this form	Yes / No Yes / No

ETHICS CLEARANCE GUIDELINES WHEN HUMAN PARTICIPANTS ARE INVOLVED

The Ethics Application Form must be submitted on Moodle for approval prior to conducting the work.

Considerations in data collection

- Participants will not be identified, directly or through identifiers linked to the subjects in any reports produced by the study
- Responses will not place the participants at risk of professional liability or be damaging to the participants' financial standing, employability or reputation
- No confidential data will be used for personal advantage or that of a third party

Informed consent

- Consent to participate in the study has been given freely by the participants
- participants have the capacity to understand the project goals.
- Participants have been given information sheets that are understandable

- Likely benefits of the project itself have been explained to potential participants
- Risks and benefits of the project have been explained to potential participants
- Participants have been assured they will not suffer physical stress or discomfort or psychological or mental stress
- The participant has been assured s/he may withdraw at any time from the study without loss of benefit or penalty
- Special care has been taken where participants are unable to consent for themselves (e.g children under the age of 18, elders with age 85+, people with intellectual or learning disability, individuals or groups receiving help through the voluntary sector, those in a subordinate position to the researcher, groups who do not understand the consent and research process)
- Participants have been informed of potential conflict of interest issues
- The onus is on the researcher to inform participants if deception methods have to be used in a line of research

I have read, understood, and will adhere to the ethical principles described above in the conduct of the project work.

Signature: 

Date: 10/04/2021

Appendix I

1) Fingal Open Data: http://data.fingal.ie/About

Licence

Citizens are free to access and use this data as they wish, free of charge, in accordance with the Creative Commons Attribution 4.0 International License (CC-BY).

Note: From November 2010 to July 2015, data on Fingal Open Data was published in accordance with the PSI general licence.

Use of any published data is subject to Data Protection legislation.

Licence Statement

Under the CC-BY Licence, users must acknowledge the source of the Information in their product or application by including or linking to this attribution statement: "Contains Fingal County Council Data licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence".

Multiple Attributions

If using data from several Information Providers and listing multiple attributions is not practical in a product or application, users may include a URI or hyperlink to a resource that contains the required attribution statements.

2) Eurostat: https://ec.europa.eu/eurostat/about/policies/copyright

COPYRIGHT NOTICE AND FREE RE-USE OF DATA

Eurostat has a policy of encouraging free re-use of its data, both for non-commercial and commercial purposes. All statistical data, metadata, content of web pages or other dissemination tools, official publications and other documents published on its website, with the exceptions listed below, can be reused without any payment or written licence provided that:

- the source is indicated as Eurostat
- when re-use involves modifications to the data or text, this must be stated clearly to the end user of the information

Appendix II

Journal of Statistics Education: http://jse.amstat.org/jse_users.htm

JSE Copyright and Usage Policy

Unlike other American Statistical Association journals, the Journal of Statistics Education (JSE) does not require authors to transfer copyright for the published material to JSE. Authors maintain copyright of published material. Because copyright is not transferred from the author, permission to use materials published by JSE remains with the author. Therefore, to use published material from a JSE article the requesting person must get approval from the author.

Consent Form – Debriefing sheet and Information sheet

https://form.jotform.com/210993328435056

User Testing Form

https://docs.google.com/forms/d/e/1FAIpQLSd1_G6Q27Lgl_ENHF9ePDXUNkm76TQ9wRX4sVOaE5InHO9YOQ/viewform?usp=sf_link

6.3 Reflective Journals

October Monthly Reflective Journal

When I began my software project module at the end of September, I was introduced to the necessary requirements for the module. The module would span over our two semesters. As soon as I attended the first Software Project module, I started immediately thinking about possible project ideas.

During the first couple of weeks, it was all about idea generation. I would have to prepare a project idea pitch. This would either go on to be accepted or rejected based on the lecturer's opinion of the idea. Before I started in NCI, I had a specific project that I've always wanted to investigate. It was based around a generation algorithm related to sneakers. The idea was that a user would customize their own specific shoe on an application and the output would be a list of already existing shoes available for the user to buy. I understood at this point, I would need some sort of webAPI or perform web scraping. This brought me to investigate different possible technologies.

When I started to think more into technologies, I knew that I did not want to use the language Java or the IDE Android Studio. I have used these resources previously and I didn't enjoy that type of project. I was thinking a small bit about using the language Python. I have been learning python over the summer and am really enjoying it. I felt as this would be a good challenge for me as I have never developed a large-scale python project.

As I was thinking more about my idea over the number of weeks, I had to decide what kind of project to develop. Did I want to create a mobile app, a web application, a game or even a research/data project? With heavy research I found a good framework called 'Flask'. This is a web-based framework that uses python as a back-end language. I believe a web application would be the best approach for my chosen idea.

Over the last number of weeks, I have been continuing to practice in python. I have been using a Udemy course as well as different YouTube tutorials to learn different aspects of Python. I've also started looking further into the framework of Flask. I've been following YouTube tutorials and started making basic programs, loading data onto a local webpage.

In week 3, I conducted my Project pitch. This involved a 5-minute brief overview of my project idea. I feel as though my video went well. I went deep into detail on my idea with possible technologies in mind. Now I must wait for my idea to be accepted or not. Once I receive this information, I can really start to make some serious progress on my application and start my project proposal.

Moving into November, I will now focus a lot of my attention on my project proposal. I also hope to do more research into my project topic and build a well-detailed plan for the year. I will continue to do Python and Flask tutorials to improve my efficiency in that area.

November Reflective Journal

What I've done

This month I focused a lot of my time towards pre-development and the beginning of my mid-point implementation.

At the start of the month, I submitted a draft of my project proposal and ethics form. I included all areas of requirement. However, when I submit my full draft at the end of December, I want to add more detail into the Testing and Evaluation section. I need to talk more about the specific testing I will conduct. I also want to add more detail to other areas, such as Objectives and Project Plan.

I created wireframes for every one of my web pages. This will help me visualize what I want to create. I also started to draft my Use Cases and UML's.

In regard to research, I've watched more tutorials on Python and Flask to improve my effectiveness in the technologies. I have looked at other ways of implementing a login feature. I'm still considering using Python sessions or using a database to store the information.

I also created my Website navigation, with 4 separate pages. I used bootstrap for the nav bar.

What I have learned

I have learned more about the capabilities of the framework flask. I learned more about Bootstrap from implementing it when conducting my navigation bar.

I also learned about the necessary headings in the technical report and the importance of focusing on that for the mid-point implementation. I now know about the importance on focusing on the security and business side of my idea.

What I'm currently doing

Currently I am focusing on making sure everything is working in flask and that my webpage is loading correctly. I am thinking more about the other functions in my web app. At the moment I have inputted a Forum and a Blog Page. In the next few days, I want to focus more on my Use case Diagrams and UML.

What I plan on doing next

- Research on deployment methods, for example Heroku
- Acquire a domain name for my website, for example Shinder.com
- Fully draft UML and Use Case Diagram
- Research other services I can include in my website
- Develop a login function
- Start drafting my Technical document

December Reflective Journal

What I've done

This month I focused on completing my mid-point implementation and demonstration.

At the beginning of the month, I focused on the many different requirements for the Technical report, such as my Use case scenarios, user requirements and mock-ups. I constructed my Use Case Diagram around the different functions I was planning to develop. I wrote up my four use case descriptions for each of my functions.

I also spent time developing my midpoint prototype. Using Visual Studio Code, I created static pages for each of my web pages. Using bootstrap, CSS and HTML I added a format to each page. I also altered the design of my prototype, using many different CSS techniques.

I continued working on my database connectivity, using an URI to connect to a test script. I researched multiple ways of connecting my Mongo DB database. By using either a URI or connecting to local host, I would be able to send input to the database.

I also deployed my website using a Heroku server. Using Git bash I was able to push my current project up on the web.

What I have learned

From deployment, I found Heroku to be extremely user friendly. You can easily make a change with a git push. Also, the ability to have a specific domain name is very handy.

I learned so much about the use of virtual environments and how they operate. I also learned about the many different packages I can use to facilitate the functions within my application.

What I'm currently doing

Currently I am focusing on the functionality of my login system. Also, researching ways of tackling my main function.

What I plan on doing next

- Research ways of storing my unstructured data
- More design implementation into my web pages
- Look into implementing Google AdSense
- Review more capabilities of Bootstrap
- Adding functionality to my other services, e.g., forum, review

January Reflective Journal

What I've done

The majority of this month, I catered more of my time towards my TABA's. After submitting my midpoint implementation and demonstration, I did not complete much in my project. At the start of week one, I begun developing my project again.

I have fixed a major error within my website. The error was to do with taking user input relating to log in details. I have now found the solution taking the user details, encrypting the user passwords, and creating a function to assign a certain uuid for each user.

What I have learned

I have learned more about different python and flask packages and the use of uuid's for users. I've learned about the importance of incorporating encryption for users' passwords. Also, how to conduct encryption in flask. I now understand different methods for writing to a Mongo DB database.

What I'm currently doing

Currently I'm working on my login/register function. I'm focusing on linking the user information with the created MongoDB database. I aim to get this function completed by the end of week 1.
What I plan on doing next

- Completing the Login/register function.
- Look more into my main function.
- Look into implementing Google AdSense into my website.
- Start developing my main function.
- I want to start developing my sub-services by researching CRUD functionality in Python and flask.

February Reflective Journal

What I've done

This month I focused a lot on the main functionalities of my project. Firstly, I developed my sneaker searcher scavenger page. This involved inputting text to inform the user, a prize reveal animation, and a folium map API. I firstly considered using Google maps, but found folium to be much better. I was able to set a waypoint on a part of Dublin City centre and input a radius for that specific location.

I have fully completed my login/registration function. A user can now register or login to my website to access the numerous functions. Once a user is registered, their data will be passed to a mongo DB database. When written to the database, the password is encrypted to ensure proper user confidentiality.

I also developed my Sole seeker search function using a sneaker database API. I research heavily to get this functionality to work. As of now, a user is able to enter specifics related to a shoe (such as brand and gender) and be presented with the top 20 pairs of sneakers that relate to them.

Lastly, I started designing the UI for my homepage. I have structured the layout with a template and found a way of showcasing my promotional YouTube video right on the homepage.

What I have learned

I have learned so much more about the flask and python capabilities. After completing the login/registration function, I now know the requirements to build a secure login system for all types of users. I have also learned about using a map API in flask with the utilisation of Folium. After completion of the sole seeker search function, it has taught me all about enabling database API's and relating the output to user requirements.

What I'm currently doing

Currently I am designing the sole seeker search function to make it look nice and user friendly. I am also still designing the homepage and brainstorming what to include in my promotional video.

What I plan on doing next

- Implementing a PayPal API for a donation link in order to gain revenue.
- Start developing my Forum page.
- Try out a few Testing techniques.
- Focus on documentation.

March Reflective Journal

What I've done

This month I focused on many different aspects of my project. One being the implementation of the forum page. Right now, I have an application that allows users to input their name, title of the post and a post section. This data then gets sent to a mongo DB collection. I designed my sole seeker function. I used different colours and HTML div's make it look presentable. Also, I researched different ways of importing a PayPal function into my website.

In terms of documentation, I have created a step by step explanation for one of my main functions. It highlights each step of the process and explains the code snippets.

In terms of testing, I have completed and documented a render unit test for every page on my website.

What I have learned

I have learned a lot about the necessary testing for a large application and the importance of it. I have also learned more about Flask's capability as a micro-framework. I now understand how to connect two separate MongoDB collections to one Python Script.

What I'm currently doing

Currently I am looking at ways of displaying all Forum posts in the MongoDB collection to a page on my website. I am also conducting out different testing techniques, such as user testing and unit testing. One of these types, being Flasks unittest package to make sure all templates in my application render onto the page successfully. I am also trying different ways of implementing the Twitter API for my homepage. I am also looking again into implementing Google AdSense from my deployed site.

What I plan on doing next

- Implementation of my PayPal system.
- Perform advanced unit testing and user testing.
- Make all pages similar in respect to design.
- Focus on documentation.
- Re-edit technical doc from December submission to align with current application.
- Create my project Poster.
- Shoe seeker designing.
- Promotional video.
- Deployment update on Heroku.

April Reflective Journal

What I've done

In the month of April, I completed many necessary aspects of my project. One being the implementation of my forum page. As of now a user can make a post on the forum page, have that data be sent to a mongo DB database and the post be printed permanently on that page. I also have the ability of delete any posts from the backend database which will in turn delete the post off the page.

I also implemented a Twitter API to the homepage of my website. I have three separate twitter account postings that surround the idea of Sneaker news and updates. I feel these look perfect for the homepage.

I successfully deployed my application on Heroku and tested that all functions were working correctly, and they were.

A major issue I was having this month was related to my Sole Searcher web API page. An internal server error occurred presenting no API output. I investigated to see the error and it seemed as though it was calling an API that wasn't there anymore. Due to this I sent an email to the developer on Swaggerhub. He replied saying that they shifted over to another site, so I had to obtain a new API key along with implementing different parts of code. This was quite difficult as it was hard to get my old code to work with this new API key.

After a while I got the API working again and added 20 more results for the user. The user will now be presented with about 50 sneaker results.

In terms of testing, this month I conducted multiple user testing for my website. I created a consent form and testing form to get feedback for my site. This testing is how I found the internal server error with my Sole Searcher API page. I have conducted a unit test for all pages on my website, testing different functions such as the login function.

In terms of Google AdSense, I submitted my application to try and get adverts for my application. However, this was unsuccessful as the response stated there wasn't enough valuable information on the page to display advertisements.

Lastly, I added more design to the Shoe Seeker Map API page. The text is laid out better for the user and steps are clearly defined.

What I have learned

This month I have learned much more about manual user testing. I now understand the steps involved with consent and gathering opinions of the participants. I also learned much more about working with online API keys and the management of errors that occur. I have also learned about updating a Heroku deployment with new changes to my application.

What I'm currently doing

Currently I am looking into implementing a donation link with the PayPal API into my application. I am also designing the forum page to have the posts wrapped within a forum-like layout.

What I plan on doing next

- Promotional video for the homepage.
- Write test cases for my user testing.
- Focus on areas of documentation.
- Project Poster.
- Reference code.

6.4 User Testing Consent and Survey form