Westhill Bike Club - Coursework 2 Report

Joanna Mein
40316854@live.napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

This report covers the design of the web-app including URL hierarchy, enhancements, critical evaluation and personal evaluation.

Keywords – Python, Flask, Bootstrap, Jinga2, JSON, Bike Club, Cycling, Membership, Create Ride

1 Introduction

This web-app which has been created for Westhill Bike Club using Python Flask runs in the browser. The web-app has information about the club, the annual Westhill bike ride and membership. Another feature that is available is the ability for people to sign up for an account and login where they will be taken to the dashboard. Once in the dashboard users will have the option to create rides and events which will be posted to the events page. There has also been a contact form implemented which people can fill out with any questions or queries they might have.

2 Design

2.1 Languages

This web-app has been built using Python Flask with the use of Jinga2 HTML templates, CSS and JavaScript. As well as all of these languages a database was also implemented using SQLite3 and SQLAlchemy. This database was used to store the data from the sign up and create an event features. The main running of the web-app can be done using Python Flask through Levinux, Terminal or similar command line software. It is displayed using local host

2.1.1 Python Flask

The structure and running of the web-app has been developed in Python Flask using Terminal on Mac, although it can also be run using Levinux and other command line software. Flask is the framework which allows websites to be built in python and run in any web browser. Using Python Flask lets developers create web-apps without the need to write lines and lines of code, features can normally be executed in just a few lines of code. During development a number of plugins were added to help with the functionality of the web-app. The plugins are imported into the web-app using the following code at the top of the index.py file:

```
from flask_wtf import Form
```

To install the plugins into python flask pip install was used. Below is a list of the plugins imported into flask:

- wtf_forms
- flask_sqlalchemy
- flask_login
- flask_mail

This is an example of how pip install was used:

```
$ pip install wtf_forms

```

2.1.2 Jinga2 HTML Templates

The Layout and structure of the app has been created using a number of different jinga2 HTML templates which are rendered in the index.py file. A base template was created which stored the main features of the pages such as navigation, image banners and the footer. A dashboard base template was also created because the dashboard pages differ slightly in style from the rest of the web-app. The templates are rendered as below, for example the about page of this web-app is rendered as below using one simple line of code:

Listing 1: Render Template Example

```
@app.route('/about/')
def about():
    return render_template('about.html')

```

2.1.3 SQLite3 Database and SQLAlchemy

An SQLite3 database was implemented into the web-app to store user data from the sign-up page and the create an event page. This particular database was chosen because it was the language which was used in the workbook provided. Having also looked at examples and tutorials online SQLAlchemy was chosen to work alongside SQLite3. This decision was made because it meant that the developer wasn't required to write raw SQL because SQLAlchemy does most of it for you which helps to save time and effort. To import SQLAlchemy into the web-app the following code was placed at the top of the index.py file:

Listing 2: Import SQLAlchemy into web-app

```
from flask_sqlalchemy import SQLAlchemy
```

Once SQLAlchemy has been added into the flask web-app the database location needs to be defined. The location will vary from user to user because it depends where they have the web-app located on their computer.

To define the database location the following code is placed in the index.py file:

Listing 3: Defining Database Location

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:////Users/jomein/Desktop/coursework2/sourcecode/database.db'
db = SQLAlchemy(app)
```

Creating the database can be done two ways, either in the termial using sqlite3 and python or adding a single line of code to the bottom of the index.py file. Below is an example of the single line approach which was chosen to make it easier.

Listing 4: db.Create_all()

```
if __name__ == '__main__':
    db.create_all()
    app.run(debug = True)
```

2.1.4 Database Security

Database security is extremely important especially when storing user data such as emails and passwords. Werkzeug.security has been used in the web-app to hash and salt the passwords so that if the someone retrieved the data from the database it would be extremely difficult to decrypt it.

2.2 URL Hierarchy

The design of the URL hierarchy has been developed with the users in mind so that they can easily navigate their way around the web-app. Users can move around using the Navigation bar which is located at the top of the screen. The page the user is on is reflected in the URL. For example, if the user is on the membership page then the URL is "http://localhost:5000/membership/". This makes it clear that the user is on the membership page. Below is an example of how the URL is set in Python.

Listing 5: Recipes URL Hierarchy in Python Flask

```
@app.route('/membership/')
def recipes():
    return render_template('membership.html')

```

The hierarchy of the web-app is as below:

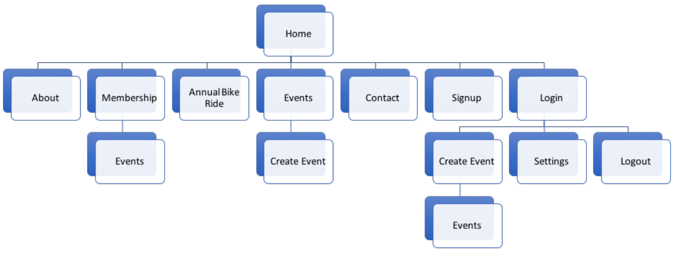

Figure 1: **URL Hierarchy** - Westhill Bike Club Web-app

2.3 Styling

2.3.1 Bootstrap

To style each page of the web-app Bootstrap has been used. Using this tool helps to makes the overall design consistent on every page. It also allows for easy design layouts using the style sheet provided by Bootstrap. However, custom styling has been implemented in this web-app to make it more appealing for the users. For example some of the custom styling which has been done is to the navigation, changing the colour to fit with the colour scheme of the bike club. Other custom styling which has been done is to the forms on the various pages such as the sign-up, log-in and create event forms. One of the other reasons Bootstrap has been used is because using the classes created it makes the pages responsive for all devices which is important to users when using a web-app or website. Having this functionality means less time is spent styling and making every aspect of the web-app responsive.

2.3.2 Imagery

Throughout the web-app there are banner images on pages. This helps to make it more interesting and eye catching for the user, making their experience more enjoyable. The images selected are all related to cycling. Images were taken from Unsplash[1] and the westhill bike club Facebook page[2].

2.3.3 Colour Scheme

The colour scheme implemented was taken from the club colours which are already used. These colours are yellow and black. The online tool Coolor [3] was also used to find complementary colours. The full colour scheme used in this particular web-app is yellow, black, white. The specific colours which have been used are:

- Yellow - #EFA942
- Black - #000000
- White - #FFFFFF
- Blue - #3B73A1
- Grey/Blue - #565B6C

3 Enhancements

3.0.1 Forgotten Password

One of the enhancements which could be added to the web-app is to have a forgotten password link for users. This feature would enable users to click the link forgotten password and be taken to a form where they enter the email that they signed up with. This would then send an email to the user with a link to create a new password. Having this feature would greatly improve the usability of the web-app because at the moment there is no way to change the password. This feature is important to users because many user sign up for a number of different sites and services and forget the password they created.

3.0.2 Membership Form

Another enhancement would be to have a membership form displaying on the membership page. This form would enable user to easily and quickly fill out the form which would then be sent straight to the club administrator. A benefit of having this feature would mean that users don't have to go to another site, download the form, fill it out and then email it to the club. This would cut out a few steps and making the users experience of the web-app more enjoyable.

3.0.3 Admin Users

Having admin users would be an extremely beneficial feature of the web-app. the admin users would be able to see all the current signed up users. They would also have the permission to edit and delete the events created. Another permission they might have it to only allow certain users the ability to create an event. Having these permissions in place would give the web-app more security and stop unnecessary events from being posted by those who are not members of the club.

3.0.4 User Dashboard Features

Enhancements could be made the the users dashboard. Features which could be added might be settings. This would let the user edit their personal details held by the club and the function to create a new password. The dashboard could also show all the events that, that particular user have created giving them the option to edit any of the events they have posted.

3.0.5 Share To Facebook

Another enhancement which would prove to be extremely beneficial for users would be the ability to post a newly created event straight to Facebook. Having this feature would mean that duplicate events wouldn't need to be created on both the web-app and the Facebook group.

3.0.6 Strava API

Something else which would again be beneficial would be to have a Strava API so that when users are creating rides for the events page they can show the route which the training ride is going to be using. Having the API would mean that users wouldn't have to leave the web-app to see the route because they would be able to see it on the events page under the description of the ride.

4 Critical Evaluation

One of the main features of the web-app is the ability for users to sign-up and create an account which will then allow them to log-in. The developer feels that this feature works well although there could be improvements made. Some improvements might be to add verifying the user so that a forgotten password can be implemented.

The other main feature is the create an event feature in the dashboard. The developer feels that also works well because of the simple and easy to use layout of the form. However when the event is created the date does not display the way the developer had hoped and could cause confusion for the users.

The contact form works well although the developer feels that it might have been better to implement it a different way. This is because the contact takes a while to submit and send the email to the specified address.

The developer feels that the overall design of the web-app is suitable for the bike club. They kept it design simple and modern looking with the banner images of pages and the content displayed on all the different pages. Although the developer is happy withe the design and layout they would have liked to improve the dashboard if more features had been implemented into the web-app.

5 Personal Evaluation

When creating this web-app there were a few challenges to face. These included learning more advanced Python Flask and database integration to display on HTML pages.

5.0.1 Python Flask

To build on the knowledge I already had of Python Flask I worked my way through the Workbook provided[4] which gave examples of the skills and knowledge required in order add more functionality to web-apps. To find out more information about the plugins available and how to use them I used documentation websites[5][6][7][8]. This gave me the necessary information about how each plugin works with flask.

5.0.2 SQLite3 and SQLAlchemy

SQLite3 and SQLAlchemy were both new to me, however I had some some basic MySQL in the past so their were similarities when looking into SQLite3.To help me understand SQLite3 it used the examples in the workbook provided as well as a number of different websites[5] and YouTube videos. This research gave me valuable examples of how to implement a database and the best ways to achieve what it was I wanted to do in my web-app, which was creating a log-in and event creator. Once I felt I had looked at enough examples and documentation I started to put into practice what I had read and seen using a mixture of what I found online to get my desired outcome.

5.0.3 Overall Personal Reflection

Overall I feel the developing of the web-app went well without too many challenges or problems. Although some challenges were faced I was able to overcome them easily with the help from articles and website online. I also feel the skills and knowledge I have gained through this development of the web-app will help me later on in university and my career after university.

References

[1] “Unsplash,” November 2017.

[2] “Westhill bike club facebook group,” November 2017.

[3] “Coolors,” November 2017.

[4] S. Wells, “Notes & workbook 2017-2018,” November 2017.

[5] “Sqlalchemy documentation,” November 2017.

[6] “Flask_mail,” November 2017.

[7] “wtf-forms,” November 2017.

[8] “Flask$_{l}ogin,'' November$2017.