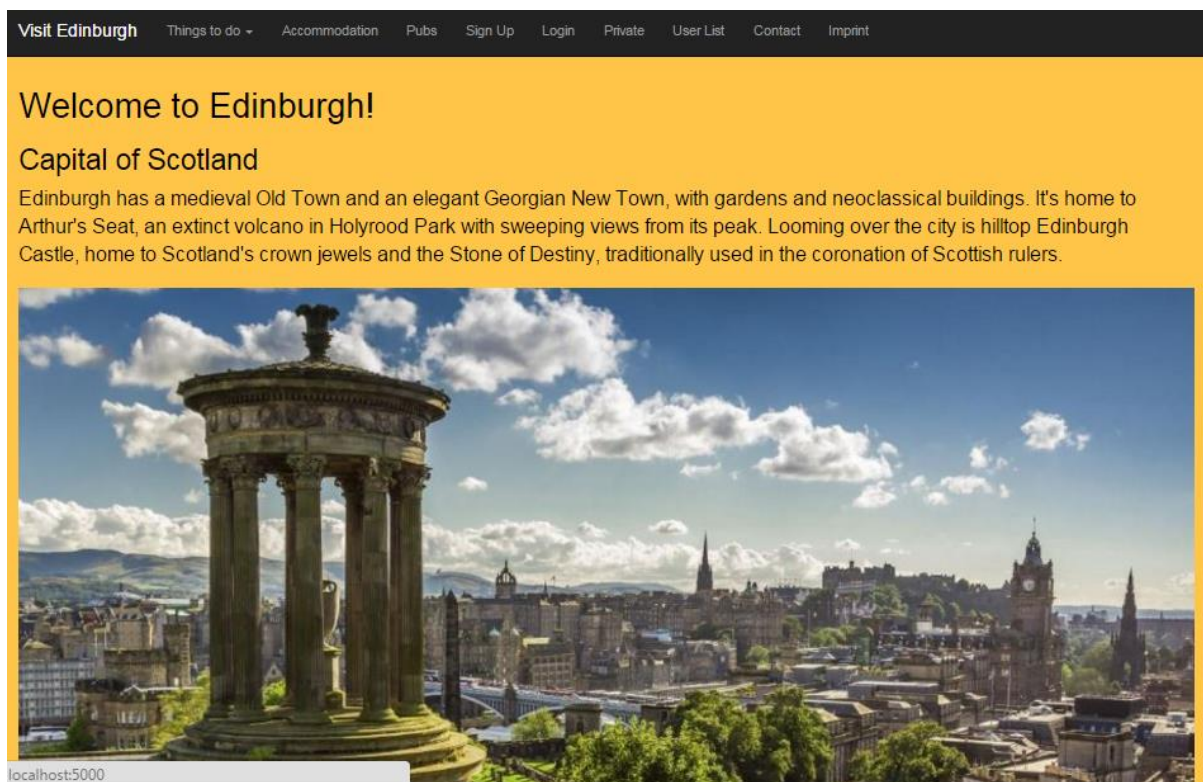


Advanced Web Technologies - SET09103

Module leader: Dr Simon Wells

Coursework 2

Web-app: Visit Edinburgh



Lukas Nachtmann

Matr.No. 40220839

Edinburgh Napier University

Introduction

As part of the lecture “Advanced Web Technologies – SET09103” at Edinburgh Napier University, the students’ task for the second coursework was to develop an own web-app project. The aim of this coursework has been implemented by using the given learning environment.

This web-app is an Information page about Edinburgh. In Edinburgh there are tourists all the time. With this web-app they will get a short overview about things to do in Edinburgh, accommodations and pubs. The web-app will help people for planning a trip to Edinburgh.

The Web-app shows famous attractions, activities, accommodations and pubs. Furthermore, links to the official websites are given for more information, if required.

In order for the web-app to grow and improve, there is also a contact page provided that invites everyone to give requests and feedback about the web-app. With the Multiple User Login users can create their own account and login. The aim for the future is, that users can add favourites to their account and can comment the attractions and activities.

For building the web-app application such tools as Python, Python Flask, Jinja2, HTML, CSS or JS have been used. Python Flask-Login was installed, the license is stored in the folder “licenses”.

Design

For this webpage a menu template called 'menu.html' is used as templated inheritance. With this template the navigation bar at the top is repeated on every page of the web-app. The main routes can be chosen in this menu ('Things to do', 'Accommodation', 'Pubs', 'Sign Up', 'Login', 'Private', 'User List', 'Contact' and 'Imprint') and you can go back to the starting page with the button 'Visit Edinburgh'. The dropdown button 'Things to do' contains the options 'Activities', 'Attractions' and 'Tours'. All other templates extend this navigation bar from the menu template.

The architecture of the web-app starts with a welcome page. This page can be seen by navigating to 'http://localhost:5000' and 'http://localhost:5000/edinburgh' on the browser. Based on this there are different routes designed.

The route '/edinburgh/things-to-do/activities' leads to a list showing the activities in Edinburgh. The route '/edinburgh/things-to-do/attractions' leads to a similar list showing a few great attractions in Edinburgh. On both pages the names of each activity and attraction are linked to external weppages. Own webpages are in the planning process. One example for the Edinburgh Castle is finished. It can be seen on the route '/edinburgh/things-to-do/attractions/edinburgh-castle'. The last dropdown button Tours lead to the route 'edinburgh/things-to-do/tours'. The different city and ghost tours of Edinburgh are listed here. External links, which opens the official websites in a new window, are embedded as well.

The next two buttons on the navigation bar are created for the routes '/edinburgh/accommodation' and '/edinburgh/pubs'. On each of this pages should be stored a database, but it isn't finished because there was no time left.

A Multiple User Login was realised with the python flask-login function. The button 'Sign Up' leads to the route '/createaccount'. There users are able to

create their own account. They can choose a username, a password and can enter their email address. The created user accounts can be seen on the route `‘/display_users’` with help of the button User List.

The button `‘Login’` leads to the admin login with the path `‘/login’`. There it is possible to sign in, if the user have created an account before.

After login, page is redirected to the private sector with the path `‘/private’`. There you get also with the last button `‘private’`, but you need to sign up to see the content.

There are two more buttons in the navigation bar.

The route button `‘contact’` opens a page for requests and feedback. Users are able to write their ideas or requests in a text field and can submit them to the email address of the web-app developer. Microsoft Outlook is required for this function.

The button `‘imprint’` contains a page with information about the web-app.

Last but not least, two error pages has been created for the 404 error and the 401 error. The first one can be tested with the route `‘/force404’` or with any not existing route like `‘/errortest’`. The error page shows that the entered path does not exist and one is asked to go back to the welcome page of the web-app. It also contains a direct link leading back to the starting page. The second one will be displayed on the path `‘/private’` without logged in before. The user will be required to log in otherwise he is not authorized to see the content. A direct link leading to the `‘/login’` path is given as well.

Enhancements

Because of the limited time frame that is given to work on the coursework, this web-app includes only a few things about Edinburgh.

The idea of this web-app is to integrate databases with lists and information about pubs, accommodation etc. The Users should be able to bookmarking their favourite things about Edinburgh and commenting them. So other users could see the comments and take them into consideration. Therefore the above named databased have to be finished.

In addition, it could be necessary to integrate a search function in the navigation bar, if there are big databases on the web-app.

An Admin Login with Flask-Admin is also conceivable.

Another possible idea is that, users can add other users as friends and write them text messages.

Of course a lot more Information could be stored on the web-app.

Finally, the appearance of the web-app could be changed for a better web-design. There are so many ways to do this.

Critical Evaluation

The critical evaluation of this coursework is very difficult for me. Because I don't know exactly how complex the web-app has to be.

All requirements of the task have been fulfilled. For the chosen theme, an information page of Edinburgh, all really needed information has already been implemented.

The web application is easy to operate and self-explanatory, special knowledge to use it is not needed, except using the learning environment.

An important feature of this web-app is that, users are able to provide constructive feedback. This helps to extend the web-app in a direction the users would like to see. With this information the web application should be improved.

As already mentioned, the web application will get more efficient with the named improvements in the enhancements section. The missing databases would be enrich the web-app.

Personal Evaluation

For this coursework and the first coursework it was essential to be familiar with the given learning environment. I never used 'levinux', 'putty' and 'github' before and in the beginning it seemed to be difficult. With the help of the exercises in the workbook I became acquainted with the learning environment.

Equally I didn't know the Python language, Flask and Jinja2. I learned to work with them. During the design of the web-app I had to learn a lot of HTML and CSS tags, there were many things, I used a long time before, but I forgot them.

The biggest challenge for me on this second coursework was to create the Multiple User Login. I had to read lot of things and spent so much time to get the Multiple User Login working. To realise it, I used python flask-login. First I had to install it and after that I had to get familiar with it. My source code was getting more complex and it's really hard to look through it. The time went by in a flash, therefore I had no time left to design my other ideas or to finish the databases.

In conclusion, I can say that I tried to keep it simple and simultaneous I learned lot of things of Web Technologies from this coursework and I have enjoyed the work with it. I think I have created a well-designed web-app, which contains all aspects of Flask that we have covered. Also my web-app has a well-designed URL hierarchy and associated routes. I tried to implement more advanced features which I investigated myself, e.g. the Multiple User Login with python flask-login.

Because of the github connection doesn't work on my computer, I couldn't push the files regularly to my github account. So I only pushed the final versions to github at the Kilby Computing Centre.

Resources and References

Information about the learning environment, python flask, flask-login:

Workbook (Notes & Workbook 2015-2016) SET09103 – Advanced Web Technologies

<http://chrn.info/dokus/vim-anleitung.txt>

<http://www.fullstackpython.com/flask.html>

<http://flask.pocoo.org/>

<http://flask.pocoo.org/docs/0.10/patterns/sqlite3/>

<https://github.com/maxcountryman/flask-login/>

<https://flask-login.readthedocs.org/en/latest/>

<http://stackoverflow.com/>

<https://realpython.com/blog/python/using-flask-login-for-user-management-with-flask/>

Information about HTML:

<https://wiki.selfhtml.org/wiki/HTML>

<http://www.webmaster-crashkurs.de/externe-links-erstellen.htm>

Information about the Edinburgh:

<http://www.edinburghcastle.gov.uk/>

<http://www.visitscotland.com/see-do/attractions/>

<http://www.visitscotland.com/see-do/activities/>