Coursework Report - Student Accommodation Web-app

Tatiana Huskova
40207956@live.napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

This report is about a prototype of an online catalogue of student halls in Edinburgh. It describes the web-app, the design of its code, offers ideas for improvement as well as critical and personal evaluation.

Keywords – advanced, web, technologies, napier, set09103, report

1 Introduction

The Private accommodation for students web-app, as its name suggests, offers a list of student halls, provided by private companies in Edinburgh. These vary in location, price, availability, additional services and the way the rooms look. The app shows relevant information about each of the selected houses in a separate URL address.

2 Design

The app runs on a single python "index.py" file, which is supported by a couple of additional files. The index file contains an errorhandler that redirects to a 404 error page, if URL is not working/known and two additional routes - homepage and house page.

Homepage uses a basic Jinja2 template supported by simple CSS and allows user to see the list of houses. The list contains the name, price and address of the house. Based on those two key factors (price and location), the user can later decide, which house would he be interested in and search for additional information.

These are hidden in the second route. It was designed so that there is no hardcoding of a route for every single house individually needed. By typing a user-friendly URL that is rendered from the name of a house, the user is then redirected to the page of appropriate house, which contains additional information, such as tenancy length, extras offered in addition to the accommodation or a picture of a room. All the data is stored in JSON file and displayed in a simple Jinja2 template. This route also contains a "for" loop, which changes the "not-that-pretty" URL to a neat one therefore, simplifies the navigation for user.

3 Enhancements

It might be useful to add a search bar into the homepage to allow users to use a keyword to search halls. The search result would then select halls that contain desired keyword, rather than showing them all of the possibilities. However, this would require using a proper SQL database.

The current style of the JSON file makes the URL layout process over complicated. Simplifying the file by separating its sections and naming them might ease further process of retrieving data from the storage and therefore make the Python code slightly simpler to come up with and write.

Another, less useful, but still valued thing would be an improved CSS and page design in general. Design style used in this coursework is very simple, nothing too attractive. A well-designed website is one of the key factors of success. Listing houses in a table would make the information at the homepage better readable and more consistent. Whereas the right size of an image at a house page could reduce scrolling.

4 Evaluation

Critical and Personal Evaluation.

4.1 Critical Evaluation

The most interesting part of the code, which I am most proud of, is the generalised route for every single house. It saves many lines of code and allows to perform communication between Python and JSON. This is something, I had in my mind since the very beginning, but was quite tricky to come up with a plan on how to do it and still make it look good and human-readable.

On the other hand, the template together with the CSS do not cooperate very well and this is visible when looking at the house page. Adjusting the image properties may be the answer.

If I kept my data storage structure written as it was originally, I might be able to get rid of the dictionary part in my Python homepage route.

4.2 Personal Evaluation

Once again, the old well-known has proved again: no matter how hard you try, there is always that one colon missing and breaking the entire code. When coding in Flask, the same applies for space counts.

The task was still quite challenging, despite having all the topics covered in practicals. Web development is not as easy as it seems, especially not in a new language. At the beginning, I was struggling a lot. Whether it was loading data or making the code work properly. It took me a time to realize that the idea, which I bear in my mind is not possible to do with so little knowledge, I had about Python. It led me to do further research and background study, which was very helpful and brought me to a couple of interesting websites.

I cannot complain about the lack of time, I just underestimated Python and overestimated my abilities. Something I will not do in the second half of the assessment. Even though there still is space for improvement for the project I submitted, I can see a lot of progress in my performance since I started working on this coursework.

References

Code Wolf, T. (2017, June 15).*Python Flask Web Development: Flask Templates, Jinja2 Tutorial, Bootstrap Integration, and Custom Error Pages*. Retrieved October 23, 2017, from https://www.thecodewolf.com/python-flask-web-development-flask-templates-and-flask-bootstrap-integration/.

Lee, F. (2016, October 16).*Python: Using Python, JSON, and Jinja2 to construct a set of Logstash filters*. Retrieved October 23, 2017, from https://fabianlee.org/2016/10/31/python-using-python-json-and-jinja2-to-construct-a-set-of-logstash-filters/.