

WEB-APPLICATION: FINAL REPORT

INTRODUCTION

My CookBook is a web-application running with Python within the Levinux environment. The concept of my web-app is to share my recipes collection with the other users. Users will have to sign up on my web-app to see my recipes collection. After signing in, they will be redirected to my home page and they can start enjoying the visit by choosing one of the category of recipes they would like to see. The route of the link will redirect them to the category page that will show the different recipes we had. Finally, to contribute to my collection and make a more enjoyable visit on my web-app, users would be able to add their own recipe to my collection if they would like to.

DESIGN OF MY WEB-APP

The architecture of my app is composed with 5 different routes running through Python Flask. The first welcome page is the sign up page where a form will be requested to login to my collection. The second page is the home page where the user is redirected from the signup page. Then You will find 3 other routes that redirects to our different category page (Starter, Meal, Dessert).

For designing my web-app, I have used Bootstrap with Jinja2. I found a nice fluid bootstrap theme on startbootstrap.com. It makes my web-app more attractive for the user that he will like to stay longer to find out more information.

Root = Sign up page → Home page → Starter page – Meal page – Dessert page

CRITICAL EVALUATION

To build my web-application, I analyse different type of recipes website. The advantage of creating a web-application for a cooking book is that it is mostly websites and not application. I have decided to innovate into a digital cooking book.

When the user arrives on my main application, he is directed to sign up by giving a name, email and password before entering on the website. When he submits the form, he is redirected to the home page which will be customized with your name. The advantage of customize with the user's name make it more personal for the user as he received a nice welcome message and attracts him to stay. The problem with the form is that the user will still be able to access to the website even if he doesn't write any name or email. It might due as it is not connecting to a database. Otherwise, the form still managed to work and redirect the user to the information they are looking for, the application is straightforward.

On the main home page, the user is able to choose the category that redirects him to a list of the different recipe. The list has not been managed to be displayed horizontally, but the user is able to

understand each recipe. The only elements missing would be more data for each item. The user is only able to access to the name of different recipes and image corresponding to it, the right side of the page could be used for the rest of the data. The same information for the other categories can be found on the other pages.

Moreover, I decided to use a yellowish colour for my theme to influence a friendly atmosphere as I want him to stay on my web-app. The colour yellow will also be more relaxing for the eyes.

The only defaults of this application is that it would need some connection to a database to be able to access to more data and it would help the web-app to interact with the users and become more user-friendly. For example, the form that requires to ask the user to add their recipe would send a notification by email to the administration of the web-app. It would also need more content that would have the same information as in a real cooking book. For the moment, the book only has a look of recipe catalogue.

PERSONAL EVALUATION

Creating a web application from scratch, with a new language that I was not familiar with two months ago, has been a first real challenge for me. I have seen my progress in my coding skills especially in the back-end programming.

I have faced different challenges during the project. A large problem I got stuck with was when I tried to add a specific image with each item of the list. The main issue was to enter a list inside another list already created in python. The solution I have managed was by entering a value for the key of the list in flask (ex: meal = { 'key item' : 'value img' }) and don't forget to insert the source of the images from the static file in the template of the page.

Another issue was the problem of uploading to a non-existing file as Python would not create a file for you so we had to not forget creating a file for the destination of your file or change the address of your URL by an existing one.

I also had problems with my routing URL as I didn't define the same variable in the @app.route and in the development of this route. When you define the route @app.route(/home/), you will have to start your development by def home(): otherwise it won't recognise the content of the page.

Some other small issue was by forgetting to define request, url_for methods in the variable to import. In conclusion, I can say I have learned a lot of new skills although I still need to improve my application with bugs or problems I still not have fixed yet. I made the impression of Python is a sensible language that needs a lot of attention on your code as it can break very easily.

SUMMARY

REFERENCES

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

<http://jinja.pocoo.org/docs/dev/templates/#expressions>

<https://docs.python.org/2/tutorial/datastructures.html#dictionaries>

<https://www.codecademy.com/courses/python-beginner-en-pwmb1/o/1>

<https://code.tutsplus.com/tutorials/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972>

<http://opentechschoool.github.io/python-flask/>

<https://learnpythonthehardway.org/book/ex32.html>

<https://startbootstrap.com/template-overviews/agency/>