Advanced Web Technologies Coursework Report

Roche Alexandra
40404291@napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

The goal of this coursework is to create a prototype of an online directory using the Python Flask micro-framework. It must demonstrate the acquisition of knowledge in Python, Flask, but also in frontend technologies (HTML, CSS, js), in database management and website structures and security. For this prototype, I chose to create a website that indexes Japanese language schools in Japan, which I have decided to name JapLanguageSchools. To do so, I used tools as Python Flask for the website structure, Jinja2 which is Flask's template tool and Bootstrap for the aesthetic aspect of the website. This report contains all conception and implementation steps with screenshots of the final application.

1 Introduction

Concept **JapLanguageSchools** is a web-app linked to a database containing information about Japanese language schools in Japan. That information is about school details (Name, location, website, accommodation...) and school curriculum (Duration, costs). From the app, you can access directly to the list of all schools, or you can use filters to specify your search. It is possible to filter results by determining a location (City), price range or course duration. You can also do a specific search by entering a name or a city in a research field.

Content From the home page of the app (Figure 1), you can access different pages. The first option is to display the entire school list (Figure 2), and the second is to sort results by categories[1]. I chose to implement three categories: City, price range and program duration. There are different options for each. For example, by selecting to sort by price range, it is possible to display only schools which cost less than 1000 euros per month, or more than 1000 euros per month. Once you chose a sorting, all matching schools are displayed. More important information is displayed for each school (Name, city, and duration). By clicking on "More information", the user is redirected on the school page[2], containing all information about the selected establishment.

[1] Screenshot in Appendix 1
[2] Screenshot in Appendix 2

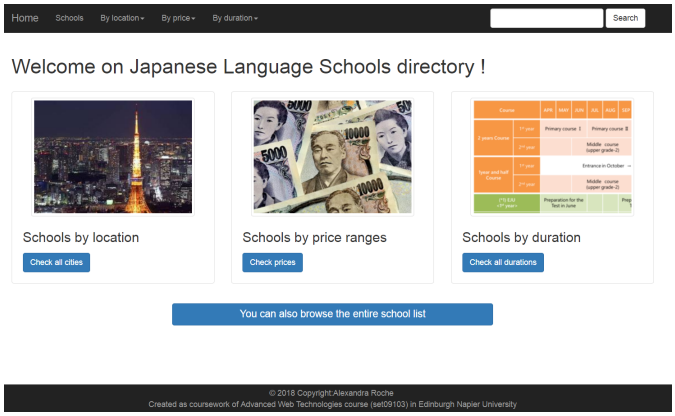


Figure 1: **Main Page -** You can select categories or chose to display all schools

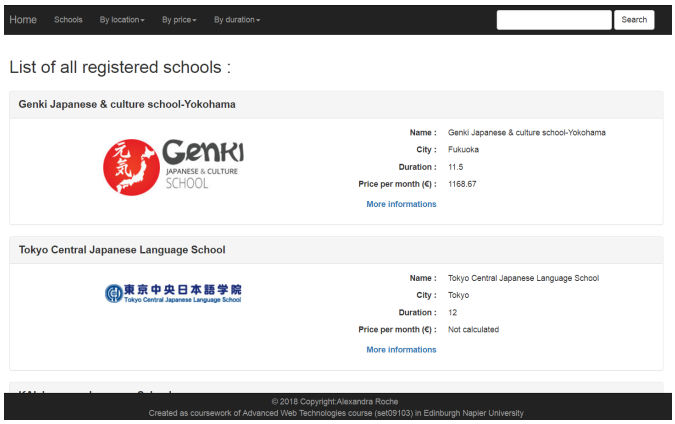


Figure 2: **Schools display page -** List all schools and display some information

2 Design

Website structure Because JapLanguageSchools is an online directory, it has to be efficient, that means that the user should find the requested information as simply and as fast as possible. So the website structure is fundamental in this process, it should be clear and straightforward. So the main page offers all available research options to the user: Display all schools, or chose a category. Those options are accessible from all the website's pages from the navigation bar on the top of the screen. It avoids the user to be lost on the application and to be able to change the sorting option at any moment. Category pages show all sorting options and also show by advance the number of results depending on the sorting choice. The navigation bar also contains a

search field, where the user can search a specific school by entering a city, a district or a part of a school name. By searching, it redirects the user on the search page where results are displayed. From results, there is access to the school page containing school details, prices separated from general information. It also includes a link to the school's website, for even more information.

URL structure The website's homepage is on the root (URL *'/'*). From this page, the user can access on the website's five main pages, as shown in figure 3. From the school display page, you can access to each school pages. Those school pages have a URL which specified the school number (this number corresponding to the index of the school in the database). For the sorting options pages, the URL contains the value of the selected category (price, city or duration). When an option is selected, it appears in the URL as a parameter. When the user clicks on the button to get more information on a school from a sorting results page, he is redirected to the school details page on the URL '*/school/<schoolnumber>*'

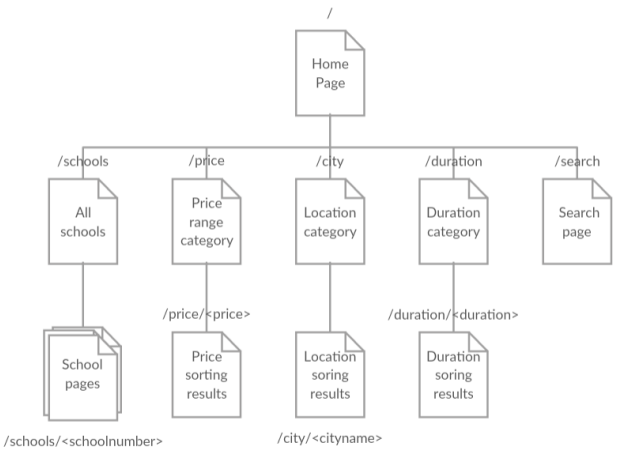


Figure 3: **URL Hierachy** - From the level 0 (Root) to level 2

3 Enhancements

There is a list of improvements that can be implemented on JapLanguageSchool.

- On an individual school page put a map by using Google Maps API showing the location of the school.
- Implements a logging system that allows one user (The website's administrator) to change school information. It would make possible for the administrator to update data without using direct access to the database. Data modification will be easier thanks to the graphics interface of the website. However, implementing this system will force me to keep a look at the security system. Implement a grade system with permissions should be a good idea for possible future improvements but it has to be completely secured. Creation of a user table on the database will have to be created with data encryption to keep personal information. Staying aware of hacking. Be careful about data access, and for example prevent SQL injections or other hacking strategies.
- Create new entries with new schools. If the user has corresponding permissions, it will allow creating new lines in the database. It can also calculate automatically price total, and an API could get the current money currency to convert yens into euros or pounds.

4 Critical Evaluation

When I chose the kind of directory I wanted to implement, I was pleased to find a topic which I am concerned by. That allows me to know what the user's purpose is when using this website. From this point, I deduce how to build my application.
In my opinion, information displaying is a strength of my application. The preview of each school in sorting pages or search pages shows key point information, and it can reduce the necessity for the user to visualize the entire school page or can help for comparison between schools. Also, the search system works well and allows a search by name (or partial name), city or district. Some towns in Japan are so big that just giving only the city do not provide enough information to the user. That is another example of implementation that I have made because I am interested in the topic and because I know that it is useful in some cases.
However, I think that my code could be more constructed. Some points should be improved to make my code more readable for someone else. For example, instead of transferring raw data from my flask code to my template, I could convert it into a Python object with attributes. There are some other points in my code which could be improved like this. Another example is to highlight in python flask code the separation between functions which communicate with the database and others by putting them in different files.

5 Personal Evaluation

Process and challenges For this project, I choose to work only by the Linux terminal to access the server. Even if I know bash language for two years, doing an entire project with VIM and without any IDE provided with a graphic interface was challenging. I already used some provided tools such as Bootstrap or Python, but I never used Python to create a website (I used Php most often). Thanks to practical classes, Python Flask was easy to handle. The most complicated part was to transfer data between Python flask code and templates using Jinja2. Jinja2, in general, was a new tool for me, and it took me some time to know how to use it. Hopefully, thanks to all documentation available on the internet, I discovered all Jinja content pretty fast, and it didn't block me that much. If I should rework a part of this project, I think I would

spend more time on error handling and the security aspect of the application. If I want to implement a way to modify data on the website, I should take a more in-depth look at it. Even if I have already implemented solutions and responses to most errors that can appear, I should read more flask documentation about it to make my application as secured and functional as possible.

Personnal Feedback First and foremost, I was pleased to see that we had the choice of our topic. I spent a really good time working on JapLanguageSchools, mostly because I needed to research language schools in Japan in the first place. So this work was a way to do a "two-in-one" and maybe this application will allow me to share more easily my research with my family and friends to choose my future studies.
It wasn't "hard" coursework for me, mainly because I made some website before. So I already knew how to proceed with this kind of projects. However, it still is exciting because it is not always the same work environment nor restrictions. So every work is a new challenge.

6 Bibliography

References

[1] *StackOverflow*.
https://stackoverflow.com

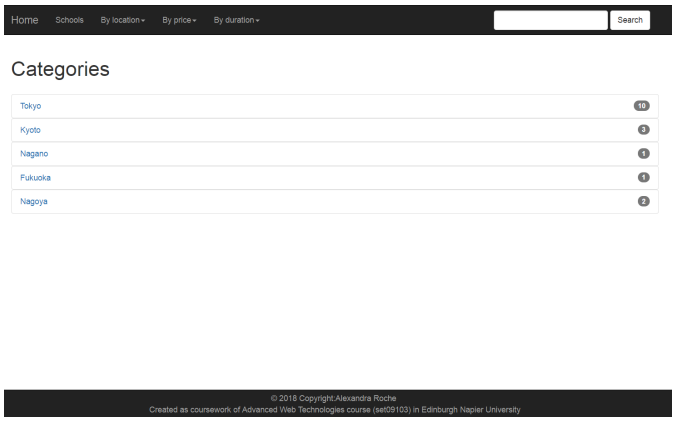
[2] Armin Ronacher
Jinja2 Documentation.
http://jinja.pocoo.org/docs/2.10/

[3] Armin Ronacher
Flask documentation
http://flask.pocoo.org

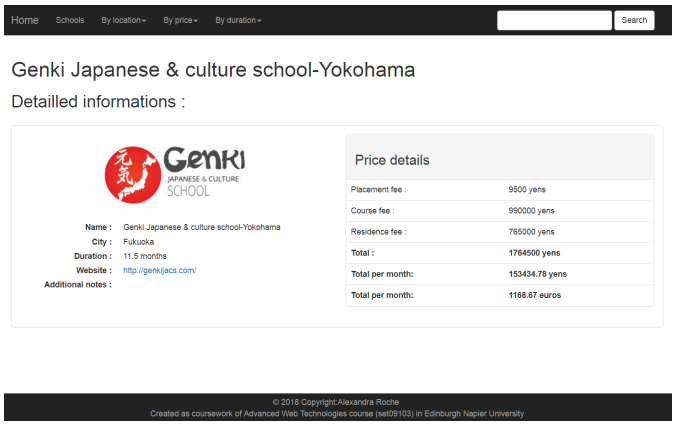
[4] Henrik Huttunen
Vim tutorial
https://www.openvim.com

[5] *Python tutorial (French)*
https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python

7 Appendix



Appendix 1 - Page where categories are displayed



Appendix 2 - School page with information