

Coursework Report

Boyang Zou

40333231@live.napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

Keywords – Boyang Zou

1 Introduction

This first coursework is a simple catalog web app relate to guitar. The contents of this web including the distinction between electric guitar and classic guitar, some videos and introduction about current famous guitarist, and introduction and pictures about different kinds of guitars. Those collections were divided into four parts and were put into four different pages which were available through the links of the navigation bars in the home page. In addition, when user is under the page that introduce the famous guitarists, they can click the seven pioneers of guitar's name to get the corresponding information about them.

The whole website is consisted of a dozen of different pages, and every single page have a different url. When user jump to another page, they would be able to navigate to another site. The web-app achieved transition among different pages and achieved the function of redirects requests and response. It contains many static files such as images and css and also contain a simplified login interface. When user submit a form whose "username" and "password" are a default value it will redirect to the index page and the value will display in the url and display a "welcome username" behind the login bottom. If not the default value, it will stay in login page and user will receive a error message. When user input a undefined url, it will display a error page which instead of the original error page to remind user is not valid url, and this page inherit the navigation bar from index module by using template inheritance.

```

20      </form>
21      </div>
22      </div>
23      </div>
24  {%- if error %}
25

```

2 Design

There are some steps to build the web-app. The first step of the coursework was using html build a home page and a dozon of its subpages and using links to connect these pages, which achieved the responses between different pages in the browser. Then searching the collections on the internet, putting different types of collection into corresponding pages and utilizing css, bootstrap to make those page look ok.

After finished this, The next step was establishing two folders in Levinux, 'static' folder to contain all of my static files like css file, images and bootstrap tools, and also builded a 'templates' folder to contain all of the modules. The third step was building a app.py file to set up url for every module and using jinja2 tool to template those modules. Now it can achieve the transition from different pages and URL Hierarchy function, for example, when user click a link under the main page, they will be navigate to a new url based on the url of main page.

```

1 @app.route('/guitarist/Paul/')
2 def guitarist_1():
3     return render_template('people1.html')
4
5
6

```

```

1 @app.route('/login', methods=['GET', 'POST'])
2 def login():
3     error = None
4     if request.method == 'POST':
5         if request.form['username']=='Zouboyang' and
request.form['password']=='zby962311':
6             return redirect(url_for('index', username=request.form['username']))
7
8
9
10
11
12
13
14
15
16
17
18
19

```

```

1 @app.route('/')
2 def index():
3     return render_template('index.html', username=request.args.get('username'))
4
5
6

```

And then is to achieve the simple login function. Through using redirects and importing the value of "username" and

"password" which was inputted by user in the login page into index view function and its module. When user input the right password and username, they would turn the index page from login page and the value will be displayed on index page's url and a welcome "username" message behind the login link.

```
1 @app.route('/')
2     def index():
3         return render_template('index.html', username=request.args.get('username'))
4
5
6
```

However, if user input the wrong information, they will stay in the login page with a error message under the page.

Finally a error 404 page was built to remind user when they input the wrong url.

```
1 @app.errorhandler(404)
2     def page_not_found(error):
3         return render_template('error404.html'), 404
4
5
```

3 Enhancement

The first coursework is a very fundamental web-app. Therefore, there are many features need to be added or improved.

First, add more error pages response to different error occurred and appropriate information that enables users to make an informed decision about what to do next. Because most users will be confused when some thing is wrong during they are browsing a webpage. They need some relevant guidance to tell them what is going on and induce them how to fix this. Otherwise, they might lose the interest of continue browsing the web.

Second, the web-app needs a user register function, thus user can register their own account in this web and login by using their account. When user are doing the register, they need to submit the information they inputted in the form, those data will be receipted and stored. Therefore, a database needs to be established to manage those data. In addition, a database can also grant and limit access to data based upon criteria such as user name, password, region or account number, databases also enforce data integrity by ensuring that data is collected and presented using a consistent format. This means user can only register successfully by obeying the request to input their information. Currently, this web-app can just login by input a specific username and password. The establishment of database make every user can login to the web-app by using their account they registered with a consistent format.

Thirdly, it is great to add a simple forum to enhance the interaction among different users. Some people who are good at playing guitar can share their experience and put some tutorials on the forum to help those amateurish people. Also, people can exchange different opinions about guitar and get

knowledge they do not familiar with. All the data will be stored in the database.

Then a file upload function will be set. After users log in, they will have the chance to upload some file related to guitar such as some reference audio and even some advice of how to improve the website.

Finally, a simple picture database search engine will be added to allow user search the images of different types of guitar and guitar master.

Those are the features which are tend to be added and improved.

4 Critical evaluation

The first coursework just used a little knowledge of python and flask. The web-app is almost like a static webpage but achieved the requests. The transition form different url which is sensible and will not confused the user. Using two types of request methods "GET" "POST". If method is post, and input the default username and password the page will redirect to the home page. If input the wrong information, there will have a hint to remind the user have inputted the wrong thing. And the use of template inheritance properly reduce the instance of HTML embedded in Python code. Many static files such as pictures and CSS were cited for decorating the pages. However, this web-app is far away from a eligible website, it can not manage the data that user uploaded and even have no function to offer users a way to upload their data. Therefore, a good website need a web database to manage and store those data user uploaded. Futhermore, most of the interaction between user and server can not without a database. To sum up, it is a very simplified web-app that just achieved some basic function of a website. There are many place can be improved in next coursework.

5 Personal Evaluation

This coursework is a big challenge, there are many problems must overcome. First I must know something about the bootstrap and use it to build those static pages. It spend a much time to learn this and make those page look good. Then I was stuck in how to display the value user input in login page in home page after login successfully, and how to display a hint after input a wrong value. But it was finally solved through several times attempt.

To sum up, this coursework helped me learn how to use bootstrap building a website, how to use python and flask, and help me strength the knowleage I have learned in the last few weeks. It is a very good experience. Helped me figure out many problems.

6 Appendix

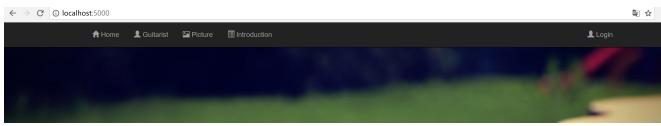


Figure 1: **ImageTitle** - Some Descriptive Text

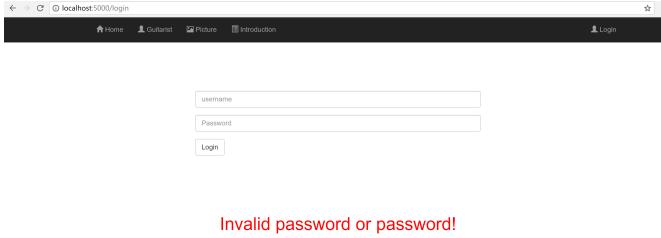


Figure 2: **ImageTitle** - Some Descriptive Text

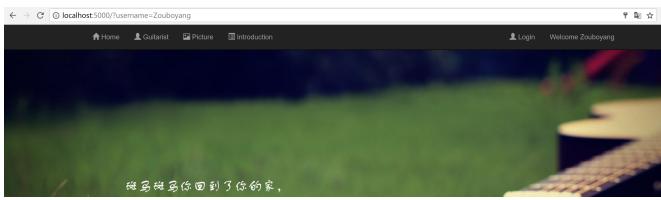


Figure 3: **ImageTitle** - Some Descriptive Text

A screenshot of a web application showing a banner for 'THE LIVE EXPERIENCE' featuring guitars. Below it is a section titled '7 Pioneers of Guitar' with cards for Kotaro Oshio and Sungha Jung, each with a small guitar icon and a brief bio.

Figure 4: **ImageTitle** - Some Descriptive Text



Figure 4: **ImageTitle** - Some Descriptive Text



Figure 5: **ImageTitle** - Some Descriptive Text



Figure 6: **ImageTitle** - Some Descriptive Text

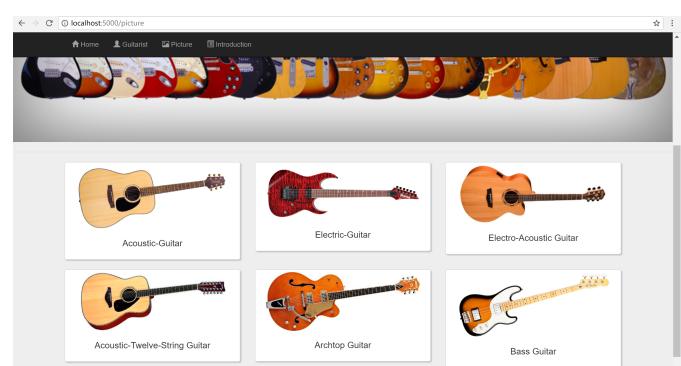


Figure 7: **ImageTitle** - Some Descriptive Text

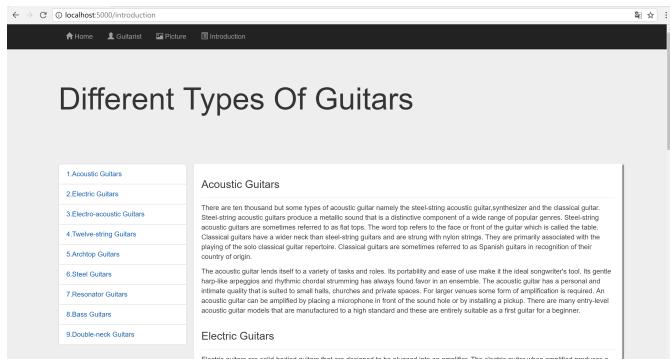


Figure 8: **ImageTitle** - Some Descriptive Text