

Cousework Report

Bowen Zhang

40333150@live.napier.ac.uk

Eninburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

The second coursework is a improvement on the first coursework. It have more features that is stronger than the first coursework. In the second coursework there are mainly contain these features. routing, requests, redirects, responses, custom error code handling, static files, templates, sessions, logging, testing, data storage, encryption, and data transports.

Key words: HTML, JavaScript, CSS, python, URL, Router, error handling, request, redirects, templates, database, static, sessions

1.Introduction

This coursework is a web application which is mainly about a singer. There are mainly have six pages in this web application. The first page is some introduction about the singer. And the second page is mainly some music which is written by this singer, and also there are also have some link can click to listen his music, and also there are videos could be watch from this page. The third page is a message board, from this message board we can leave a message, and others can find the message you have left. For leave a message in the message board we need to login first. Another page is a sign up page. From this page we can create a new account. After logging, there will have a message flashing to remanded the user have logged. And then we can leave a message on the board.

2.Design

on this step mainly need to design the whole web application. For example, first i need to consider which kind of web application i want to build. And then, i need to consider the layout of the whole frame. Last, which page would be contained in this web application. And then consider some details about each page. After a long time, I decide to create a web application about a Chinese singer.

After all of this have been finished, and then I start to design my flask. This web have some similar to the first coursework¹, but there are many features that the first one did not

have. For example, the session key, database, encryption....

In this web application, This web application mainly contain 5 pages.

Home page: this page is mainly contain some introduction about the singer;

Music page: in this page we can listen music and watch some MV, In addition we can listen some music.

Message board page: the third page is a message board. In this page, the fans of this singer could leave some message for others to read.

Login page: this page is for users to login. After they login they will get a flashed message to welcome them, and then they can get addition rights. For example, they can leave a message.

Sign up page: this page is for new user to register a new account.

After all of this done. I create a python file, which mainly contain a session key, database, and a errorhandler page. After the router have finished, and then create some directory to store material. For example, static directory, var directory, and templates directory. On the static directory, there are mainly contain some CSS,templates, fonts, images. These material are mainly used to decorate the HTML page. And in the templates directory, there are some templates could be render when we configuration router.

In the render template we can use the inherit. For example, each page will have the same navigation bar. So we can create one index.html, and then render the index.html.

```
{% extends "index.html" %}
{% block body %}
```

```
{% endblock %}
```

At first, i have add a user in this page, and also add a password on this page, also set the secret key. If the user can type the correct user name and password and then he can log in and leave a message on the message board, or he will received a message to tell he he wrong username or password. If he login, there will have a message to tell he he have been login.

And then, I connect the message board to the database. On this step, first download the sqlsite. And then create the database, and then connect the database. Therefore the message board can keep the users message. And then i create a redirect command redirect to login page if someone did not login and still want to leave message.

```

from flask import Flask, g, session, render_template, flash, redirect, url_for, request
import sqlite3

from flask import Flask

SECRET_KEY = 'ssssss'
USERNAME = 'zhangboren'
PASSWORD = '123456'
app = Flask(__name__)
app.config.from_object(__name__)
db_location = 'var/zbw.db'

def get_db():
    db = getattr(g, 'db', None)
    if db is None:
        db = sqlite3.connect(db_location)
        g.db = db
    return db

@app.teardown_appcontext
def close_db_connection(exception):
    db = getattr(g, 'db', None)
    if db is not None:
        db.close()

def init_db():
    with app.app_context():
        db = get_db()
        with app.open_resource('schema.sql', mode='r') as f:
            db.cursor().executescript(f.read())

            db.cursor().executescript(f.read())
            db.commit()

@app.before_request
def before_request():
    g.db = get_db()

@app.teardown_request
def teardown_request(exception):
    g.db.close()

```

To connect to database.

```

1 from index import init_db
2 init_db()

```

Create schema to define table in database

```

DROP TABLE if EXISTS entries;

CREATE TABLE entries (
    id integer primary key autoincrement,
    title string not null,
    text string not null
);

```

3.Enhancement

This is a fundamental web, so there are still have a lot of to improvement. For example , this web application is not good enough, it did not contain too much fantastic background to attract people to visit. Therefore, when people visit this website, they may not attract by this website. Because it is not good enough to attract more users.

And another point need to be consider is that there should be have another database for new customer to register, on this new database, the new user can register on, when the new user want to register, and type they information, therefore this database could store this information, for example, the users email, and other information. And also, when the user forget their password, and they can received the information from the information

which they register to get a new password. And then change their password. In addition, they can delete their information from the database.

And also there should add more special skills except message board to attract more visitors. For example, they could add some music to their list, therefore, when they login they can find the music about this singer directly, they do not have to look again.

4. Critical evaluation

In this coursework, it has much more improvement to the first one. For example, there is a session to add in the FLASK. The session is used to remember the user who have login. It is keep in the local, so that when the next login, the server would add the information which matched with the session. And also, in this report there are inherit has been used, it is saved a lot of time to type code, and also reduce the mistake in type coding. And also, a good point is that i add some MV and some music in the album page, therefore, the visitor can watch the MV and listen music directly without logging. And another good point is that i add some interesting message board. From this board, you can find the message that others left. That is really fantastic, maybe you will find some message attract you. In addition, from this message board, it will attract more users. But, if someone want to join the message board, they need to register first, and then they can leave message. Unfortunately, this web application less a special database for people to create account, therefore, only this people who have added in the default username and password could be login for now.

5. Personal evaluation

From this coursework, I have learned a lot of things. It is different from the first coursework. It is much difficult than the first one.

In this coursework i have learned how to use python flask to build a website. And also how use session, how to set a message flashing, how to create a log in and logout function, and also, I start to know the usage of how to use the database, and the usage. The biggest challenge in this part is how use the database and encryption. In the database it is much difficult than the other part. It spent me a lot of time to understanding how to create and how to use the database. And i watch lots of reflect video to understand the database. It is really excited after I understand this. It is worth to spend time on learning when i figured out.

In conclusion, this course give me more chance to improvement my knowledge about flask, from these course i start to interesting with the flask, and want to learn more knowledge about flask. Form this course i have learned what are URL, Redirect, router, redirect, session, database, and encryption.....this is really a wonderful project. Especially, the workbook, it including many details to help me understand the use of flask. And the teacher helped me many times in practice to understand the knowledge.

6. Appendix



Figure1. Main page and introduction

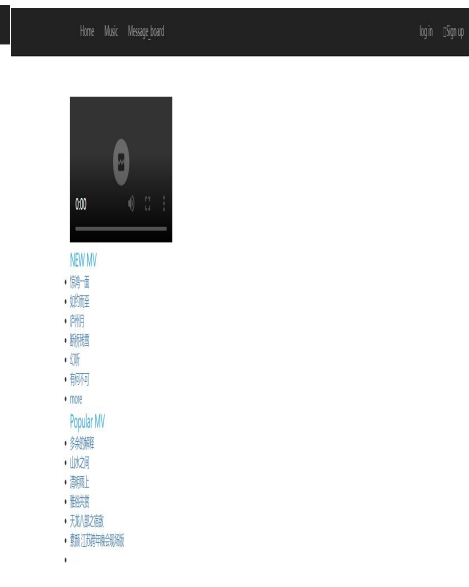


Figure2- music page

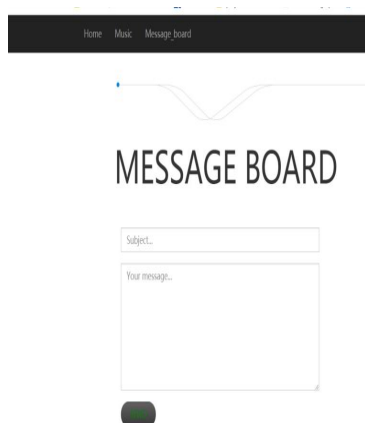


Figure3-message board

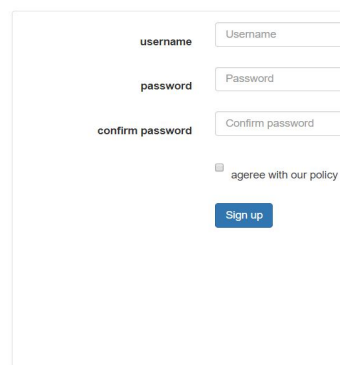


Figure4-login

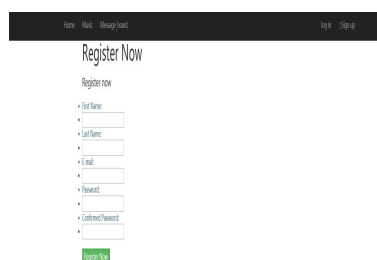


Figure5_sign up