Advanced Web Technologies Course Work Report

Darwon Rashid

40280334@napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

Since the early days of the internet, people would flock to the internet to look up information on anything they would find. Eventually the idea of a directory came about, and suddenly, websites would hold large information on anything. It could be a directory on anything, music, movies, and etc.

This course work's purpose is to showcase an understanding of the Python Flask framework. The task is to create a web application that is a directory for a given topic. The topic I have chosen for this web application is based on the Hip-Hop group called The Wu-Tang Clan.

1 Introduction

The web application allows the user to explore the Clan's artists and albums in multiple user friendly ways. The user either can manually search for whatever he/she wants by navigating through the url or navigate through the web application by navigating between pages and searching through the search bar till they get to where they want. The user can also communicate with other users through a forum page. The web application also supports admin functionality in which the admin can control the forum page.

The web application starts at the home page which displays the three options the user can go to right off the bat. They can choose to go the forum page or go view all the Clan members or go view all the Clan albums. There is also the navigation bar at the top that is there on all the pages. The navigation bar provides a feature for being able to log in and unlock admin functionality. It has two more features, first being a drop down menu that allows the user to pick between choosing to navigate between the three main options and second a search bar that looks for what the user types in.
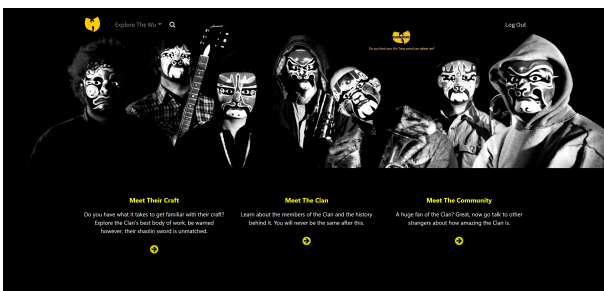

Figure 1: Index Page

2 Design

Before beginning work on the web application, the structure of how the project is set up had to be done first. The structure is pretty straightforward. Static files are saved in a folder called static, the pages themselves are saved in a folder called templates, and the data for the web application is saved in a folder called data. There is also a folder called etc which includes the config files for the web application. The main python file is in the same level as the other folders, and is the point of entry for the app. The file is called app.py and it is responsible for running the web application. This approach was suggested by the module's workbook, and also by the Flask documentation.

As for the structure of the web application, this hierarchy shows the general outline of how it is set up.
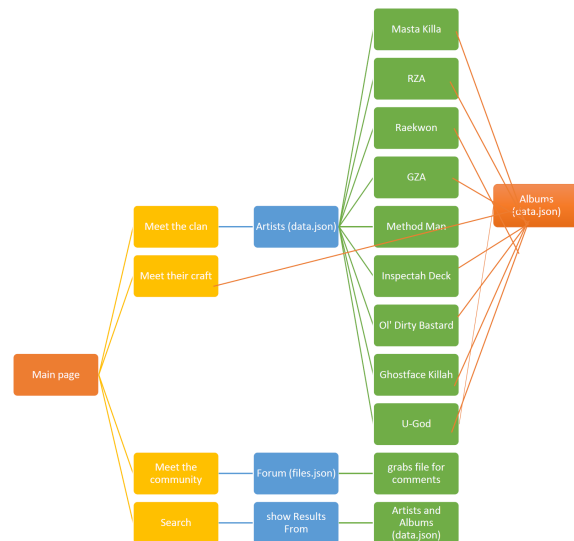

Figure 2: Hierarchy

While users can manually navigate to the information they want to find through the web application, they can also navigate using the url. The structure for how the user uses the url to navigate through the website is very straightforward and is made to be as simple as possible. Users for example, can go straight to the artist page if they wanted by doing /artist/artist-name. Having it setup like this allows users to cut the time it takes for them to get information they want.

All the data in the website comes from json objects. In the data folder, there is a json file called "data.json", this file holds all the information for each artist and their respective albums. The web application goes through this file, and then parses the data to the templates for it to be viewed. On top

of this, there is also a file called "files.json", which is used for picking files in the data folder that hold comments from the forum page. json objects were chosen to hold data due to how simple it is to add, remove, and edit them. They are also supported by Python and Flask, which makes them easily compatible with the web application.

The look and feel of the website is completely inspired by The Wu-Tang Clan. It was designed by using the popular bootstrap framework that allows for easy and responsive designing. The colors for the website are based off the black and yellow colors of Wu-Tang Clan. Bootstrap allows for flexibility and has an extensive set of tools to use for designing the web application. Bootstrap also allows the web application to be mobile responsive, so the look and feel of the web application adjusts to the size of the user's screen. It was very important to make a design that wasn't foreign to the user in terms of usability, so all of the choices taken on how information is displayed and grouped took this into consideration.

The Flask framework has a template engine called Jinja2 which is used in the html pages to add more dynamic rendering. There is a base html page that the others all inherit from. This base includes all the links to the static files, and houses the navigation bar.

3 Enhancements

For future enhancements, I would like to add another option for the admin being able to add, edit, and remove artists and albums. As it stands now, the web application is able to read artists and albums from the "data.json" file. I would like to also redesign the pages for which you see the artist and album information to make it look cleaner and more stylish.

Since I haven't tested the web application with real users, I would like to update how the user navigates with the url after I get feedback from real users. Since we were not allowed to use any database, and had to manage the json objects ourselves, I would like to add a database to hold the information. Json objects weren't made as a database replacement, but as a means to display information, so having a database manage the json objects is a much better alternative than what the web application has.

Another enhancement I would make is to the forum page, as it stands users can go into the forum page and add a comment. I would like to have users able to create their own profiles and have the ability to respond to someone's comment on the forum. I would also migrate all the persistence for the forum page handled by the web application to the database. The error handler in the website is very basic, and for the future I would like to implement a more dynamic error handler that adjusts to the different types of errors that happen in the web application.

4 Critical Evaluation

4.1 Features

For a simple directory, there are enough features to satisfy the user when using the web application. None of the features of the web application are foreign in concept or how they were implemented. There was careful background reading done on how to implement these features.

4.1.1 Forum Page

When the user gets to the forum page, the web application loads the latest comments that other users have commented. Since you can't predict the number of comments that users will comment, there was need for the web application to split all of the comments into parts. The user can navigate between these parts, each comments page represents a part of all the comments. Each comments page will hold 10 comments, then the user can either go to the next or previous page. To post a comment, the user needs to input a name and message. If there are more than 10 comments, a new json file will be created that will save the comment and then the user can view it by going to the next comments page. There is also verification for the user's input. Each comments page also sorts the comments to be displayed by latest date.
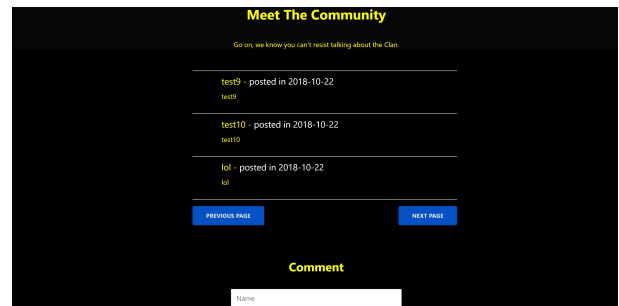

Figure 3: Forum

4.1.2 Admin Functionality

When the choice to add a forum page was made, it was certain that there should be a feature that manages the forum post in the sense that you could manage the comments. If you log-in successfully, you gain admin privileges, and then you could go to the forum page and delete comments. You can either choose to delete each comment, or choose to delete an entire comments page.
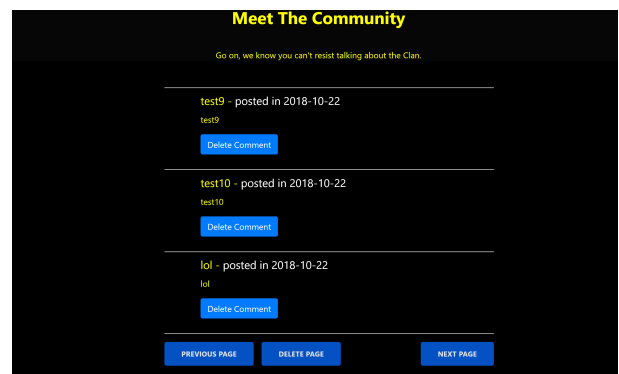

Figure 4: Admin Privileges

4.1.3 URL Search

The user can search through the web application by using the URL. The user can either search through the search bar in the navigation bar or search through the URL. It can filter out artists and see what their albums are and vice versa. You can search the URL by /search/search-phrase, where search-phrase is what the user has typed in to be searched. It will then display any results that it finds, and if it doesn't find anything, it will notify the user as well. This search-phrase can be anything from an artist's date of birth to an album's release date. The user can go straight into any artist or album page by doing /artist/artist-name and /album/album-name respectively. The user can also do another shortcut in which they can type /artist-name/albums, which displays all the albums that artist has.
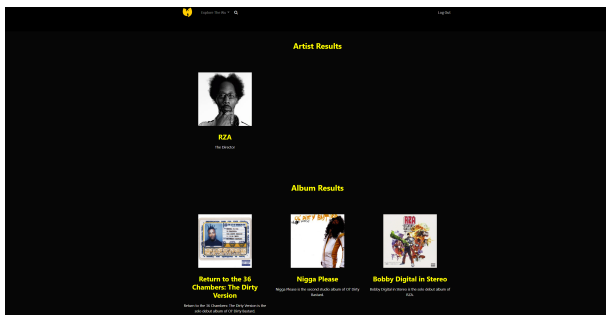

Figure 5: Search Results

5 Personal Evaluation

Through this assignment, I have learned more about the server side of things. I have about the Flask framework and how powerful it is. I learned about how to read and write to json files. I also expanded my knowledge on Bootstrap and how to use it to design pages. The workbook did a terrific job to introduce the framework and help with this assignment.

I still struggle in the front-end side for it takes a long time and gets very tedious to get to where you want with it. Using Bootstrap helped a lot with the designing of the web application which made me spend less time compared to previous course works on design due to this. I wish I took less time in the design area, I feel like I take longer than needed.

This was a good starting point in my opinion for the real challenge comes from the next one. I learned the foundations, and now I am looking forward to learning more. I feel like I performed well for this assignment.

6 References

Information about Wu-Tang Clan and some images came from here:

[1]
https://en.wikipedia.org/wiki/Wu-Tang_Clan
[2]
https://www.wutang-corp.com/artists/bios.php

Information on all things Flask and Jinja2:

[3]
http://flask.pocoo.org/docs/1.0/
[4]
http://jinja.pocoo.org/docs/2.10/

Information on all things design

[5]
https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp
[6]
https://getbootstrap.com/docs/4.1/getting-started/introduction/

Where I got some images from

[7]
https://images.google.com/