Bepop - Album Directory

Nick Taylor
40203073@live.napier.ac.uk
Edinburgh Napier University - Module Title (SET09103)

1 Introduction

1.1 The Task

The aim of this task was to demonstrate my understanding of the Python Flask micro-framework. To show my knowledge of what I have learned I had to create a web prototype application showing an online catalogue. This could range from a movie catalogue to a clothes catalogue or any other collection that could be of interest to me. One of the main points of this task was to be able to carefully create a URL hierarchy that is appropriate for navigating through the web application. I was excited to be able to explore the Python Flask framework to show off my skills.

1.2 The App

I therefore, created a music album directory called **Bepop**. **Bepop** is a multi-page interactive web application which allows you to go through many different types of music albums ranging from pop to heavy metal and learn more about each of the albums selected. I decided to create this type of online catalogue due to my love for music which made it even more enjoyable working through this coursework. To help me create a well designed website I used python, html, css and bootstrap. Shown in **Figure 1** is the startup page of my website.

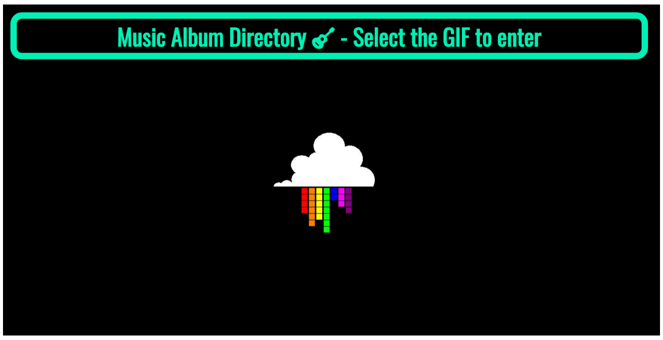


Figure 1: Startup Page

2 Design

For the design, I wanted to keep it as simple as possible so that all kinds of users would be able to interact efficiently and quick with the web app.

Startup Page I created a startup page which would be the users first interaction with my website. I tried to keep it as simple as possible so I added a header which displayed what type of catalogue this is. To add an element of fun to this I added a animated gif in the center of the screen. The main aim of having the startup page was to make the users experience as nice as possible and having this page meant it was very easy to access the main interface of the website.

Main Page When the user has made their way through the startup page of the application they are met with the main interface as shown in **Figure 2**. I wanted to have this setup like a grid as it makes it very easy and effective to show all albums being displayed. This meant I had to work on the margin and padding variables within CSS to make sure each album was displayed the correct and same size as all other albums. Each album shown on the interface had both the name of the band and the name of the album displayed underneath the album cover. This meant that the user had an idea of what they would be reading about if they dived into the album further. The album could be expanded by clicking on the album cover which takes you to a new page.

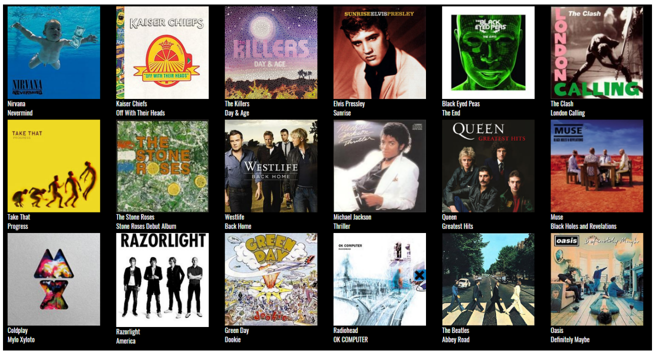
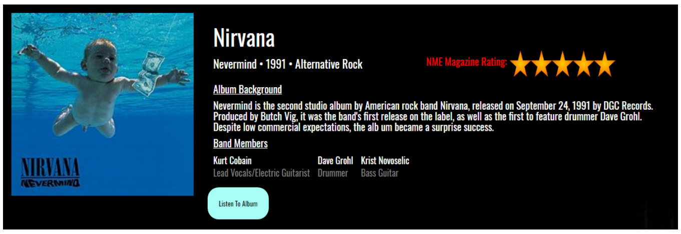


Figure 2: Main Interface Page

Album Page When on the album page for a specific album you are met with the album cover which next to it, has the album background, the genre of music, the year the album was released, the name of the album, a rating which was from a music magazine and I also added a link to listen to the album. The layout of the album page can be shown in **Figure 3**.


Nirvana

Nevermind • 1991 • Alternative Rock

NME Magazine Rating: ★★★★★

Album Background

Nevermind is the second studio album by American rock band Nirvana, released on September 24, 1991 by DGC Records. Produced by Butch Vig, it was the band's first release on the label, as well as the first to feature drummer Dave Grohl. Despite low commercial expectations, the alb um became a surprise success.

Band Members

Kurt Cobain — Lead Vocals/Electric Guitarist
Dave Grohl — Drummer
Krist Novoselic — Bass Guitar

Figure 3: Album Page

3 Enhancements

Startup Page In terms of the startup page, it does what I had intended it to do. It acts as a welcome page for the user to be able to navigate easily into the next page of the app. I believe the simpleness of this is what makes it great but I could have added some more features and make it more aesthetically pleasing. For future, I could have made the title more appealing to the eye and add more of a creative background for the page. I could add a hover feature for when you are hovering over the title it brings up a brief description of the app and what it does.

Album Cover Regarding the album cover, it would have been great if I added animations on the album. For example when you click on the album cover it would flip and you would be able to see an image of the back of the album cover with all the songs listed on the back. It is the slight enhancements like this that I believe could make my app even better.

Genre Although the website is created within the specifications, I think it still lacks some user functionality within the album page. Thinking of ways to improve this, adding a full genre category would be excellent for users wanting to be able to categorise all different genres of music. By doing this, user functionality would be greatly increased as they would be able to filter the music albums by genre.

Code/Responsiveness Throughout this coursework I spent a lot of time and effort working on the code in the back end of the app. As shown in **Code Listing 3.1** this code was reused throughout the coursework. I would have liked to have been able to use a JSON file to save all the descriptions and other attributes for the album. By using a JSON file I could have ran a for loop which would go through each album and replace each attribute with the correct part in the JSON file. This would have stopped the code being reused therefore making it a more effective solution. This would have saved a lot of time as I would not need to go through each album to add this specific code for each. If I had longer to work on this coursework I would have made it more responsive so that the web app could be web compatible and mobile app compatible.

3.1 Code Listing

Listing 1: Sample Code

```
@app.route("/music/the_beatles")
def thebeatles():
  return render_template('thebeatles.html'), 200
```

4 Critical Evaluation

Listen to Album Link Towards the end of app completion I decided that adding a link to listen to the album would be great for users to interact with the app. This means that the user was able to have a preview listen to see if they liked the album and the genre of music being played.

App Usability I believe that if I was to give my app to a user that was not very good with navigating around a website they would be able to go through the app easily with no issues. This was one of the main points I wanted to succeed in when I was planning my app. I wanted to make sure that the app would be suitable for all users regardless of their computational background.

5 Personal Evaluation

Having previously not taken the Web Technologies module, web development was a new thing for me. I had never came across Python before and also Python Flask. I had done some self study on HTML and CSS a few years back but this greatly improved my knowledge into these languages. Specifically, Python Flask was a struggle for me at first. Thanks to the workbook Simon had supplied I was always able to get a certain answer to help me in my progression with the Python language. I used the W3 schools website to help me to design my main interface and make sure everything was laid out the best it could be so it was presentable to the user. I now feel confident enough to branch out into web development to increase my knowledge even further.

6 Conclusion

Overall, I am very pleased with the work I created for this coursework. I created a user friendly album directory which was easy to navigate and also gave users sufficient information regarding each album. Looking forward to coursework 2, I am very excited to advance my skills that we have learned and create a new web application to a much higher standard.

References

S.Wells, ”Coursework Specification 1”
S.Wells, ”Workbook
W3 Schools, ”https://www.w3schools.com/css/”
W3 Schools, ”https://www.w3schools.com/html/”