# Coursework Report

Ryan Hunter
40206280@live.napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

## 1 Introduction

The Python Flask web application is a collection of rock music sorted by sub-genre, artist and album. The website has a uses a hierarchy of URLs to break down each category into smaller ones making it easier to find and retrieve information. These sections are split into sub-genre, artist and album. Each album shows the songs the album along with the length of each song.

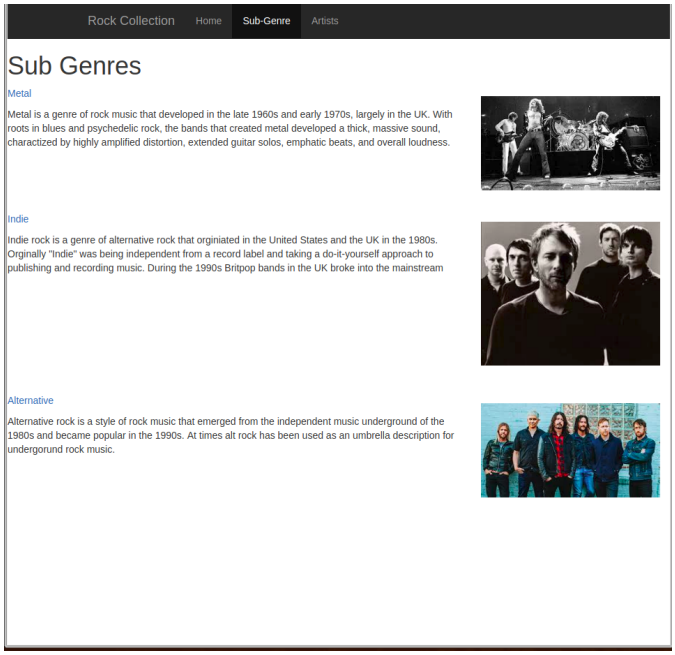


Figure 1: **Sub Genre page** - The navigation bar sub-genre page with links to all artist in the genre

## 2 Design

The most important factor to take into consideration while designing a web application is ease of use. A website should not be difficult or confusing to its users. The site was designed with this in mind. To start the design process wire frames were drawn on paper to give a rough idea of how the site will look and to see how the user will interact will the links to other pages. The colours chosen for the site are high contrast colours. The black text on a white background is simple but very easy to read for the user. The navigation bar is grey with white again for high contrast and readability. The pages all use the same structure and are almost formatted the same, allowing users to simply work journey through the site without getting confused with a drastically different page layout. The navigation bar also allows users to return to the home page at any point.

The URL hierarchy is designed to be uncomplicated and flow. For example clicking on a sub-genre will take the user to a choice of bands, clicking on a band will give the user a choice of albums from that band, clicking on the album will show the tracks and the length of those tracks. This was originally designed using a hierarchy tree giving a visual representation of the way the pages will be linked together making the problem more manageable and easier to translate into code. This approach also allows for easy expansion. It would be very easy to add another genre or more albums into the site.

Bootstrap was used to create the template for each page. The navigation bar gives the user an easy was to return to the home page at any point throughout the site and gives the website a cleaner look. The navigation bar also allowed for the sub-genre page, giving a small description of each genre, and the artists page, which allowed the user to see all the artists in the collection.

## 3 Enhancements

One of the features that could have been implemented into the web application is audio playback. If the site was able to play each track on all of the albums, it would have given more functionality to the site and instead of just being a collection of tracks, it could have been used as a streaming service. This could have been implemented by having a small play button next to the track, similar to how itunes implements it's 'taster'clips of a track, or by having the tracks name it's self with a click event handler to play an audio file.

Another improvement that could have been easily made to the site is adding an image and some information to the pages. Adding some text to each page with a description of each sub-genre, band, album and song. Also pictures of the bands and album art could have been added, much like how the sub-genre navigation bar page is styled. This would have made the website more informing and professional looking.

The website could also have been enhanced by adding more sub-genres, bands, albums and songs. This would have given the site more depth and overall been more useful to its users.

# 4 Critical Evaluation

The website ,even though basic, does achieve the end goal of being easy to navigate and well designed. The URL hierarchy is structured in a way that makes sense, filtering down the information from genres into single tracks on an album. users can go back to home at any time and clicking the previous page button on their browser will allow users to go back one category and select a different item from that category without starting the whole search again.

One of the things that could of been improved would be the python source code. Instead of writing a function for every page to publish it online, a function that automatically reads the URL for the page from the HTML source file using HTTP GET and POST a

```
@app.route('/physical')
def phy():
    return render_template('physical.html'), 200

@app.route('/lz2')
def lz2():
    return render_template('lz2.html'), 200

#black sabbath albums
@app.route('/blacksab')
def black():
    return render_template('blacksab.html'), 200

@app.route('/paranoid')
def para():
    return render_template('paranoid.html'), 200

@app.route('/selftitled')
def self():
    return render_template('selftitled.html'), 200

@app.route('/vol4')
def vol():
    return render_template('vol4.html'), 200

#rage albums
@app.route('/rage')
def rage():
    return render_template('rage.html'), 200

@app.route('/ragest')
def st():
    return render_template('ragest.html'), 200
```

Figure 2: **The Python Flask app** - How the URL hierarchy was written in flask

# 5 Personal Evaluation

During the time spent on this coursework I have learned a lot. Before starting this module I had never written a line of python or python flask and had only experienced HTML and CSS in the previous Web Technologies module. Now i feel confident in creating a URL hierarchy in flask and HTML. My CSS skills have also improved greatly and I am confident in adding some basic style to my web pages.

I now understand how to embed CSS into HTML and where to write the CSS. Previously I would guess where to write

```
<div>
    <style>
        img{
            max-width:100%;
        }
    </style>
    <img src= "{{ url_for('stati
```

Figure 3: **CSS improvements** - My ability to target an image by using style within a div

my CSS and eventually get it right through trial and error. Now that I have done some more reading and practising, I understand how to use divs with CSS and also HTML. Before this coursework I did not know how to style an image, move it and use padding on it. My python flask skills now consist of me understanding how to write and declare functions and also use static files.

The main challenge I faced was initially designing the URL hierarchy. I was struggling to map the URL tree out in my head, It was too large and difficult to comprehend. I over came this challenge by physically drawing the tree diagram on paper. This gave me a visual reference of how the website should function and made implementing my design easier.

Overall I feel i have done relatively well considering my Web Development skills were very basic at the start of this coursework. I was only confident in my ability to write hello world in HTML. Now I have created a website with forty-two pages, used a concurrent style throughout the website with bootstrap and been able to add some CSS styling, all in languages that I had never used before or was not overly confident writing. However I still could have improved this website by adding more functionality such as streaming the music on the album pages. My code could have been written more efficiently but, I am happy with progress I am currently making.