**Advanced Web Tech Coursework 1 – Report**

# Introduction

The web application that I was asked to produce was an application produced by Python programming language within the Levinux environment and using Flask library. The requirements for the application was meant to be a collection of data whether it is music, stamps or games. Apart from that, other web programming languages such as HTML, CSS or JavaScript could have been used to enhance the application.

The application that I produced is an Online music catalogue which is a music collection. My application consists of artists, albums and genre page where data is displayed from JSON into the user browser which is styled by CSS. The user can interact with the application, the same way as with a website by browsing different pages and displaying data based on their request.

# Design

The design of the application is split into three main parts. The first part is the header section which is located at the top of the application in which there are links to other pages like artists, albums and genres.

The second part of the application is the content section which is located just below the header. In this section, the content is displayed by looping through JSON data and pulling the correct information based on the user request.

The last part of the app is the footer which is just below the content page which displays simple copyright information with the year of production.

# Enhancements

There are various features that I would like to add to the application. One of the features I would like to implement is a search bar. With the search bar the users would have an easier time using the application as they would just search what they would like to see and it could save them some time.

Another feature that I would like to add is a sorting feature that would sort the albums by “album length”, “number of tracks”, “year of release” or “name”.

The feature that I would like to improve is to properly encode links. The links in the application have been encoded by using Python replace function in which the “space” has been replaced with “%20” asci code which works very well however the proper way to encode links is to use “urllib” library. The urllib library is much more efficient and effective way of encoding links.

# Critical Evaluation of the app

The online music catalogue web app has met with the requirements set in the coursework assignment. The functionality of the app is very simple as all the operations on website are done by just clicking a mouse and the information shown is clear and readable.

The overall design of the application is pleasing to the eye and simple to follow. There are some structural elements that could be implemented differently. For example, the images on the home and the genres page, could be implemented with the raw text on the background of the artist or the genre rather than the text being inside the image which makes it less readable and it losses quality when it is scaled.

The application code is well organised with proper indentation across all the files. In the HTML, CSS and Python files there are comments to different elements of the application which is useful for future developers or for people trying to understand the code.

The application could use a little bit of more python functions such as “request” or “redirect url” to enhance the code itself to make it more logical and efficient.

# Personal Evaluation

At the start of the application development I only knew some basics in Python that I learned at University in the workbook provided. The most challenging part of writing a Python application was that I have never used Python before and I had to learn basics quickly to produce my application. I spent a lot of time reading tutorials and guides on how to write a Python application. The tutorial that helped me the most was a Python guide on “Codecademy” which was helpful as it included theory connected with practical work with good explanations.

My first version of the application was loading JSON data into HTML with no styling or any complex code. As I learned how to properly pull JSON variables, I started to properly construct my code in the appropriate fashion so it looked like a proper application with styling and structure.

One of the challenges I faced during the development was linking pages one with another. My problem was how to create an “a href” link that when clicked it would link to a new page based on a variable in which my case was the “artist name”. I researched the internet for a couple of days and I couldn’t find any answer that would be a solution to my problem. I solved the problem by asking one of the assistants during a practical session of the class. The assistant managed to explain to me how links in python work and from then on I could move swiftly with my project on to the next steps.

I have also struggled for a big amount of time to encode my links. Different artist and album names had a “space” character in them therefore the URL links had to be encoded to connect to the appropriate page and to pass the variable. I have looked for the solution on the internet and there were many solutions such as using “urllib” which is an URL library in Python however I couldn’t implement it myself. The solution I came up with was to use a replace function inside the link itself, which changed the “space” character to “%20” asci character. It wasn’t a standard way of doing it however it worked well and therefore I continued working with that. The ULR code before “<a href=/{{ artist\_name }}>Back to {{ artist\_name }}</a>” and the link after the solution “<a href=/{{ artist\_name.replace(" ","%20") }}>Back to {{ artist\_name }}</a>”.

Near the end of the development I wanted to clear up my Python code as I had lists of JSON data under the “@app route” of the application and I wanted to load JSON files from files in the static directory. I researched websites like “StackOverflow” but I didn’t find anything that would solve my problem. However, during one of the practice session, my problem was fixed by our lecturer “Simon” with a very simple and efficient solution that helped me to finish my application.

By completing Python application, I have learned new skills. One of the skills that I am most proud of is learning GIT. Before the coursework, I knew what was GIT about and how companies use web repositories to keep track of the changes to their code and the new versions of their applications. I haven’t used GIT before the coursework because I wasn’t sure how to use it properly. But because on this course GIT is a requirement, I managed to learn the basics of it and I realised how it can be invaluable in production of any project.

Also, due to production of the web app I have learned Python and how to connect it with templates that use HTML and CSS. When I started coursework, I thought that Python would be a hard programming language however as I was working on my application for couple of days I understood that it is a simple language with lots of useful functions and that constructing a web app can be created with small amount of code.

# Summary of resources

Tutorials:

Codecademy. (2016). *Learn to code*. [online] Available at: http://www.codecademy.com.

Images:

Google.co.uk. (2016). *Google*. [online] Available at: http://www.google.co.uk.

Information (albums and artists):

Wikipedia.org. (2016). *Wikipedia*. [online] Available at: http://www.wikipedia.org.