Coursework 1 Report

Alexander Spencer
40323886@napier.ac.uk
Edinburgh Napier University - Module Title (SET09103)

Keywords – Coursework, 1, Spencer, Alexander, Web, Tech, Design, Advanced, report,

1 Title

Famous Painters Python Web App

2 Introduction

The famous painters web app is a python-flask based web application that show cases various painters from various periods of time or national origin, as specified by the coursework requirements.

3 Design

The hierarchy of the site can be summed up in a the idea of a tree, with the python app being the base spliting into various sections such as the static files, the config files, the templates, and the testing files.

From there once again, it splits off into the individual files of each folder of HTML files, CSS files, Images, JavaScript and Config files.

This type of architecture is rather common with most high-level web applications.

4 Enhancements

Due to limited time and experience whilst developing this web app, there are many things i would of like to improved upon for both functionality and UX design features of the app.

4.1 Functional Improvements

Some of the functional improvements i as a developer would include is a database to improve the ability to add new information about painters, e.g. a database which would be able to separate by national origin on a single web page. Of course this would take a bit more extra time to do and would be a struggle to set up considering it would be running on a python-flask based web app.

Another functional improvement i would include is an ability to create accounts so you can save lists of favourite artist, the reason i would to like to include a feature as such is because of the continuing modern art being produced these days.

A third possible functional improvement i would include is setting up a fully working search bar on the artist page. This would also be connected to the previously mentioned database functional feature which allow for custom query's through out the database.

4.2 Design Improvements

Due to time constrictions i was not able to create a more user friendly navigation menu bar, if given the chance i do believe i would be able to create a better and more user-centered navigation bar for a all pages.

Another design improvement i would hope to include in the feature would be a in page URL bar to show where the user is in the hierarchy of the site. An example of this could be a bar showing "Artist¿Van Gogh" This would give the user a better understanding of where they are in the site, and how far back they would have to go to get to a specific section of the site.

5 Critical Evaluation

In the site there are many different features that have been implemented as part of this coursework, this section is here so i may present what parts i feel work well, and which don't work well to the point of wish to improve.

5.1 Features that work well

On of the major functions of the site i feel work well is the use of redirects, since most of the artist names are not English based, people can produce spelling mistakes while searching through the artist section of the site, an example being Picasso being spelled Picasso - a spelling mistake that i myself would occasionally make. As a result the development and inclusion of 301 redirects i feel are an important and well made part of the site.

Another major function that can be included is the proper set-up of linking from one set of pages to another, i include this because of the fact in the workbook teaching of python flask it was never explained how to link multiple pages together as such, as a result i was able to learn and implement this feature on my own.

5.2 Features that don't work so well

One of the features i feel would need an improvement is the bootstrap based thumbnail which is widely used through out the site, these are used to host images with caption's, although used successfully, i feel as though the could be improved by making all the images in the thumbnails all the same size, sadly i was not able to have the time to sort this out.

6 Personal Evaluation

6.1 What i learned

During this coursework i learned so much, what i consider the most useful was the ability to setup the web server protocols. I consider these the most important skills i learned because up until this course i had no idea how to to this. Since learning this i feel as though my skill as a developer has jumped dramatically by leaps and bounds.

In addition to the setting up of web server protocols, learning the basics of python, vim and flask. Thanks to learning these skills i feel as though, like learning the web protocols, these newly acquired skills have helped me grow as a developer in a better full stack developer.

6.2 Example of Python Flask Coding

Listing 1: Python Protocols from coursework1.py

```

    import ConfigParser
from flask import Flask, render_template, url_for, redirect
app = Flask(__name__)

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

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

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

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

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

@app.route('/artist/DaVinci')
def vinci():
   return render_template('vinci.html'), 200

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

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

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

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

@app.route('/century/19th')
def ninth():
  return render_template('19th.html'), 200

@app.route('/century/15th')
def fifth():
  return render_template('15th.html'), 200

@app.route('/artist/vincent')
def redv():
  return redirect(url_for('gogh')), 301

@app.route('/artist/pablo')
def redpa():
  return redirect(url_for('picasso')), 301

@app.route('/artist/da vinci')
def redda():
  return redirect(url_for('DaVinci')), 301

@app.route('/country/artist/gogh')
def redcgogh():
  return redirect(url_for('gogh')), 301

@app.route('/country/artist/DaVinci')
def redcvinvi():
  return redirect(url_for('DaVinci')), 301

@app.route('/country/artist/picasso')
def redcpi():
  return redirect(url_for('picasso')), 301

@app.route('/century/artist/picasso')
def redcepi():
  return redirect(url_for('picasso')), 301

@app.route('/century/artist/gogh')
def redcego():
  return redirect(url_for('gogh')), 301

@app.route('/century/artist/vinci')
def redcevin():
  return redirect(url_for('DaVinci')), 301


@app.route('/config/')
def config():
  str = []
  str.append('Debug:'+app.config['DEBUG'])
  str.append('port:'+app.config['port'])
  str.append('url:'+app.config['url'])
  str.append('ip_address:'+app.config['ip_address'])
  return '/t'.join(str)

def init(app):
  config = ConfigParser.ConfigParser()
  try:
    config.location = "etc/defaults.cfg"
    config.read(config.location)

    app.config['DEBUG'] = config.get("config", "debug")
    app.config['ip_address'] = config.get("config", "ip_address")
    app.config['port'] = config.get("config", "port")
    app.config['url'] = config.get("config", "url")
  except:
    print "Could not get config details: ", config.location

@app.errorhandler(404)
def page_not_found(error):
  return render_template('404.html'), 404


if __name__ == "__main__":
  init(app)
  app.run(
    host=app.config['ip_address'],
    port=init(app.config['port']))


```

6.3 Challenges face during the development

One of the major challenges i faced while developing was having constantly having connection issues to Google API's as i was attempting to connect to the site in order to use Jquery, after a while however, this connection was given up and instead a local version of jquery was used in order to elicit a more stable connection while using bootstrap css and bootstrap js.

Another issue i had during the development of the site was the issue i faced setting up the new style of linking images, javascript files and CSS files using the flask style web application. After a while however, this became more a natural process for me to develop with repeated muscle movements.

A third issue i encountered while developing this site, was a download issue regarding bootstrap, however, after a while, i was able to gain a work around via downloading and uploading the files manually rather than via github.

6.4 How i feel i performed

Overall, i feel as though i performed adequately given my knowledge of python flask, the time i had to develop the project, the difficultly of the coursework and how fast i was able to set it all up.