

# Coursework 2 Report

Jack Sweeney  
40209035@napier.ac.uk  
Edinburgh Napier University - Module Title (SET09103)

## Abstract

For this assignment the task was to plan and develop a product that made use of everything learned while working through the Advanced Web Technologies module which also met specifications set out at the beginning of the task.

**Keywords** – Python, Flask, coursework, Jack, Sweeney, films, database, napier, report

## 1 Introduction

The task set out by this assignment was to design and develop a catalogue of entries in the form of a *Python Flask* application, this application had to meet specifications with detailed features and tools that the app should make use of, these included; *Routing, requests, responses and redirects, static files, logging, testing, and the use of CSS* which all had to be developed efficiently and with the aid of version control.

## 2 Planning

To meet the specifications detailed above it was decided that an application could be built that contained a detailed list of movies and directors. It was thought that this kind of application could be functional while also including the tools and features required for the assignment.

The app would consist of an index page containing a rolling gallery of the site's 'featured films' and a link to a database detailing these films all paired with a dropdown navbar that would contain the sites 'featured directors', each navbar section would link to a director page and consequently the films they have been involved with within the site, each of these pages would also contain the navbar discussed.

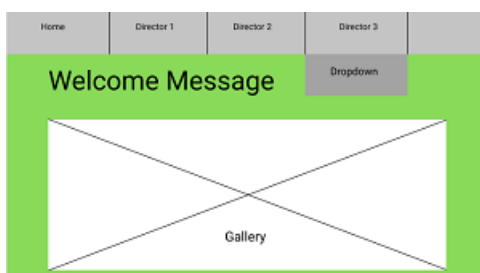


Figure 1: Storyboard

## 3 Development

For the development stage of this project many tools were used, the *Levinux* virtual environment was used for the main bulk of the work paired with *PuTTY*, *Vim* was used for the coding required for the application and *Bootstrap* and *SQLite* were used for the style and database within the application respectively.

Within the application many features were utilised. *HTML* templates were used to add style and ease of navigation to the application, within these templates *CSS*, specifically *Bootstrap* was used to enhance the style and make the application more user friendly, added to this static files were utilised to add more information on the films and directors detailed (i.e. images).

On the back-end of this site there were many features and tools added to improve the functionality of the application, for example routing was utilised to ensure users would never feel overwhelmed or lost within the site's pages, along with this requests, responses and redirects were used to make navigation far easier for users, all of which helped to ensure an easy to read URL hierarchy was constructed.

```
@app.route('/MartinScorsese')
def ms():
    return render_template('MS.html'), 200

@app.route('/Director1')
def rem1():
    return redirect(url_for('ms')), 301

@app.route('/MartinScorsese/TheWolfOfWallStreet')
def wolf():
    return render_template('wolf.html'), 200

@app.route('/WolfOfWallStreet')
def rewolf():
    return redirect(url_for('wolf')), 301

@app.route('/MartinScorsese/Goodfellas')
def good():
    return render_template('good.html'), 200

@app.route('/Goodfellas')
def regood():
    return redirect(url_for('good')), 301
```

Figure 2: Routing

A 404 *HTML* template was also included in the application to ensure users were not left at a 'dead end' if they searched for a page that did not exist

## 4 Testing

Testing was used towards the end of the project's timeline, this was an important part of development, through testing we can ensure that error codes will be correctly identified

and corrected to better understand and improve the app itself.

To test this application a test harness was constructed (Figure 3), this allows a developer to 'trial run' an application with specific error codes to see what happens, for example if a 404 error is found within an application, the test harness will find this and it will fail letting the developer know which error made it fail and where it can be found (Figure 4). If no errors are found within an app the test harness will simply run and close giving the developer a message stating that nothing was found.

```
import unittest
import index

class TestingTest(unittest.TestCase):
    def test_root(self):
        self.app = index.app.test_client()
        out = self.app.get('/')
        assert '200 OK' in out.status
        assert 'charset=utf-8' in out.content_type
        assert 'text/html' in out.content_type

if __name__ == "__main__":
    unittest.main()
```

Figure 3: Testing

```
FAIL: test_root (__main__.TestingTest)
-----
Traceback (most recent call last):
  File "testing.py", line 8, in test_root
    assert '200 OK' in out.status
AssertionError
-----
Ran 1 test in 2.396s
FAILED (failures=1)
```

Figure 4: Testing

## 5 Enhancements

If the project were given a wider time frame to complete there are a few changes that would be made overall, the first of this would be regarding the 'featured films' page and the database implemented within it, as it stands this is not an acceptable page to be included within an application of this nature and with more time to improve this could be a bigger and more attractive feature within the site, however as it stands this page still meets project requirements.

|  |  |
|--|--|
| ← → 🔄 📄 localhost:5000/FeaturedFilms   |  |
| <ul style="list-style-type: none"><li>• (1, u'1', u'The Wolf of Wall Street', u'Martin Scorsese', u'Leonardo DiCaprio, Jonah Hill', u'2013', u'8.2')</li><li>• (2, u'2', u'Goodfellas', u'Martin Scorsese', u'Ray Liotta, Robert De Niro', u'1990', u'8.7')</li><li>• (3, u'3', u'The Hatful 8', u'Quentin Tarantino', u'Samuel L. Jackson, Kurt Russell', u'2016', u'7.8')</li><li>• (4, u'4', u'Pulp Fiction', u'Quentin Tarantino', u'Samuel L. Jackson, Uma Thurman', u'1994', u'8.9')</li><li>• (5, u'5', u'Blade Runner: 2049', u'Denis Villeneuve', u'Ryan Gosling, Harrison Ford', u'2017', u'8.4')</li><li>• (6, u'6', u'Arrival', u'Denis Villeneuve', u'Amy Adams, Jeremy Renner', u'2016', u'8.2')</li></ul> |  |

Figure 5: Database

Along with improvements to databases, the amount of content within the app would also be improved with a wider time frame, right now the site details 3 directors, each having 2 sub-pages detailing films they have directed. With more time

the app could contain many more pages and biographies detailing directors and films making for a more full and content rich application.

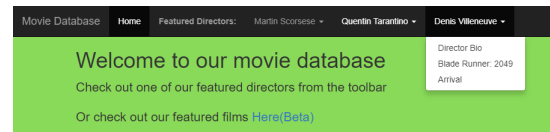


Figure 6: Content

## 6 Conclusion

In conclusion this project was a success, it met project requirements and met them within the given time frame, all while being relatively stylish, professional and user-friendly, however as stated above enhancements would have been made to specific aspects of the product if given a greater time scale. That being said this application is fit for purpose and suitable for the task carried assigned.