Coursework Report

Benjamin Brown
40279333@napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

1 Introduction

Within this assignment we are asked to create a Python-Flask app that will deliver a web-page that will be ran on the Universities server allowing users to both access and interact with in some way.

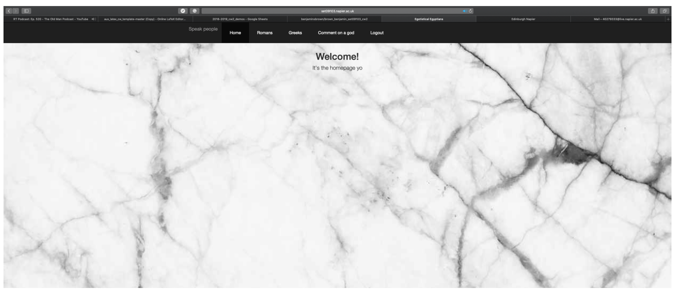

Figure 1: **Home Page** - Home page allowing you to choose Greeks/Romans

2 Design

My web app is a simple design that allows you to choose between 2 cultures(GreeksRomans) and from withing there choose to learn about 3 of the gods from each respective pantheon, there is also a third option allowing you to go to a "make a comment" section allowing you to post a comment about any god (whether they are on the site or not). transferring between the pages is easily done through either the Navbar on the top of every page of by the buttons on each page, some are pictures with href built in and others are simple buttons. My site was designed to be operated as easy as possible so anyone could enter it blindly and use it entirely. There is also an extra layer of requiring the user to login, in its current state the site only has access for one user; Username: ben and Pasword: brown, without being logged in the site will defer you to the login page continuously until you put in the correct details

3 Enhancements

Possible enhancements I would like to add are the ability to have a like/dislike option on each god, also a standard online quiz that will reveal which god you are most like(Log-in system will have to be expanded and allow user-signup for this to work). Another enhancement I would like to add is

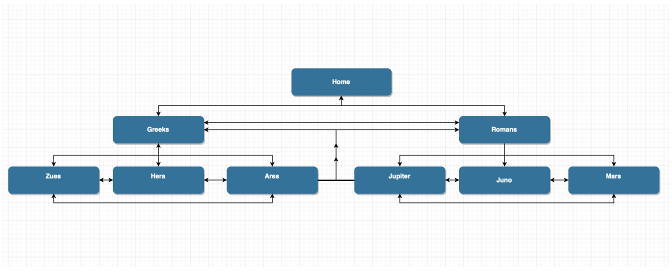

Figure 2: **Nav map** - URL hierarchy of site

a section for someone to add another god in that's currently missing with a built in evaluation to make sure that no addition is a duplicate of one already added to the site, in the same vein as this modification to the pages by select administrative users would be a worthwhile addition while regular users could put in requests for certain changes.

3.1 Multiple Users

Adding the ability to have multiple user login will require an additional layer to the database allowing it to store and intake an additional 3 fields; Username, Password and Email-address. This will be linked to an additional page called "registration" containing text-areas for the 3 required fields.

3.2 Administrators

Administrative Users would be implemented by an additional field called "is admin" which would be assigned to select users directly within the code.

4 Critical Evaluation

My Web-app works well for general browsing within the site, it has reasonably good protection against users going to places that don't exist by having a 404 redirect that mean the site doesn't just become useless should the user go to /nothing for example. However the site is missing the ability to delete posts made by any user, the only way to delete would be accessing the databse outwith the site and deleting individual posts

5 Personal Evaluation

I feel in this task I performed reasonably as I have created a successful python-flask app that does deliver the user a site through which using routing, HTML and a little bit of CSS an easy to use site is shown. By referencing the workbook i learned a baseline of bootstrap to use for the site to allow the navbar to operate properly giving a sense of consistency throughout the entire site, within each of the pages there are also links to other pages as well as a consistent "back" button allowing you to go back unless you are at the home page.