CS551P Advance Programming (2024-25)
Assessment 3: Part I – Web Development

A programming assignment to build a database driven web application with flask - details below- submit via MyAberdeen, and Codio. This is an individual work. This assessment is worth 20% of the overall marks for the module.

You need to submit a database-driven website built with Flask, which has between 2000-7000 records from an open data source. If you find a larger set, then feel free to create a smaller subset to meet the requirements. You can use this page to find an open data set for use in your application.

The aim of this assignment is to assess your knowledge and advanced skills in developing a web application in Python with Flask. You must use open data for your content.
http://theodi.org/guides/what-open-data
As this is an assessment in python, there should be no Javascript in your submission.

You should also be aware of item 3.2-3.4 in the University's Code of Practice for Student Discipline on plagiarism, collusion, and contract cheating.

You must develop an application that meets the following criteria:
1. to create a database driven Flask application with at least two linked tables. [10 marks]
2. to base your application on a suitable open data source, and to display data by appropriate means. [10 marks]
3. to load your open data of 2000-7000 records into the app in a suitable manner. [10 marks]
4. to use appropriate templates for the pages in your app. [10 marks]
5. to use appropriate error handling in your application. [10 marks]
6. to provide a good code base, making use of different application components. [10 marks]
7. to use Git as part of your development process for source control. [10 marks]
8. to include suitable tests for your codebase. [10 marks]
9. to publish the application on Render. [10 marks]
10. to document the design, development, implementation, installation and use of the application in the form of a one-page final report, and in the readme.md file of the repository. [10 marks]

Category points
3 points, attempted but broken; 5 points, working basics; 7 points, advanced working; 10 points, extras
1. 3 for flask app started and not working with one table, 5 points for working with two or more linked tables, 7, (list plus item pages), 10 points for working using wider range of components
2. 3 for displaying open data, 5 for comparing open data, 7 detailed comparison, 10 analysis doing something more
3. Attempt to load data, 3; 5 for loading some data; 7 points for loading linked tables; 10 for more
4. 3 for attempted templates, 5 with some templates, 7 points with composite templates, 10 more
5. 3 for attempted error handling, 5 for basic error handling, 7 for more extensive error handling, 10 for more
6. 3 for basic flask app, 5 points for some repetition, 7 for code in appropriate files, 10 for more
7. 3 git used, 5 points git log, 7 git log showing branches and merges, 10 for more
8. 3 for attempted tests, 5 points for working tests, 7 for good coverage with tests, 10 for more

9. 3 for attempted render deploy, 5 deployed to render, 7 deployed to render with sqlite3, 10 for more
10. 3 points for document plus readme file, 5 why & how app developed, 7 maintenance details and render url, 10 for more

Deliverables

- A one-page design and development report as PDF with student name, and Render deployment URL
- Application code, including the readme file covering maintenance, testing, running of app.
- Git log using this command:
  ```
  git log --pretty=format:'%h : %s' --graph > git-log.txt
  ```

Ideas for Applications

Go look at the list of open sources at:

https://homepages.abdn.ac.uk/b.scharlau/pages/open-data-sources/

and use the data in an interesting way. Consider how you can use the 'show' page to put your 'thing' in context with the rest of the data you have. It's one of x, in category y - how big are these other sets, how many things share similar attributes? What else can it connect with? Can you compare it to other 'things' to show them side-by-side? Aim for completeness of concept to pull data together not breadth of coverage - deeper is better than wider for this assessment.

Submission Details

Please submit the report, git log, code, and other files, submitted via Codio.