

Enterprise Software Development

Solo Assignment

Deadline: 23.59 Friday, 28 April, 2023

The aim of this assignment is to assess your knowledge and advanced skills in developing a web application in Python with Django which 'sells' something real or imaginary. You must use open data for your content, and you must use a different data set from the one used in the team assessment.

As this is an assessment in python, the only Javascript in your submission should deal with maps and charts.

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 Django application with at least two linked tables. [10 marks]
2. to enable users to 'purchase/order something' as from most e-commerce sites. [10 marks]
3. to use open data records of 2000-7000 items for your 'products' and display them. [10 marks]
4. to provide different levels of access to the app such as 'admin, user, guest', where 'admin' can see orders and appropriate charts. [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 include a search facility. [10 marks]
8. to include suitable tests using Behave for your codebase. [10 marks]
9. to publish the application with a cloud provider. [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 Django 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 items for users, 5 for ordering facility, 7 for ordering multiple items, 10 for more
3. 3 for displaying open data, 5 for comparing items, 7 using graphs & charts, 10 using maps
4. 3 for attempted authentication, 5 working authentication, 7 'dashboard' features for admin, 10 for more
5. 3 for attempted error handling, 5 for some error handling, 7 for effective use, 10 for more
6. 3 for basic django app, 5 points for some repetition, 7 for code in appropriate files, 10 for more
7. 3 for attempted search, 5 for basic search, 7 for appropriate advanced search, 10 for more
8. 3 for basic behave tests, 5 points for wider use of tests, 7 for good coverage with tests, 10 for more
9. 3 for attempted cloud deploy, 5 deployed to cloud, 7 deploy with mysql or postgresql, 10 for more
10. 3 points for document plus readme file, 5 why & how app developed, 7 user details and url, 10 for more

Deliverables

- A one-page design and development report as PDF with student name and cloud deployment URL
- Application code, including the readme file.
- All contained in your Codio box. It is YOUR responsibility to ensure that it runs in Codio.

Ideas for Applications

There is more than you can do in the time available. You need to 'sell' something real or imaginary. You can sell 'experiences', as well as 'naming privileges' for example, to comets, or volcanos, etc. Think beyond the usual 'things' that people buy. Use Faker to generate random customers, and orders of the things you're selling.

Consider how you can use the 'details' page to put your 'content' 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, or where is this 'thing'? Aim for completeness of concept to pull data together not breadth of coverage - deeper is better than wider for this assessment.

Note: there might be cloud database limits. Reduce your dataset as appropriate, and explain your reasoning in the readme.md file.