

Web Services and Application Big Project

Worth 60% of overall marks (This is marked out of 100)

Due date: Monday 20th May 2024

Please create and copy a link to your repository as soon as possible, you can continue to modify the project up to the due date.

Description:

Write a program that demonstrates that you understand creating and consuming RESTful APIs. I will allow a lot of flexibility in this project, so that you can use it as an opportunity to do something that is useful for your work.

If you cannot think of a project to do:

Create a Web application in Flask that has a RESTful API, the application should link to one or more database tables.

You should also create the web pages that can consume the API. I.e. performs CRUD operations on the data.

Handup:

A link to the GitHub repository that contains the project (and only this project).

The repository should contain:

1. Your code.
2. A "ReadMe" file if there is anything (complicated) I need to do to run this code.
3. You do not need to host the server on a cloud hosting site (Azure, Pythonanywhere) but if you do, please provide the link on your readme.
4. Any other documentation you feel is appropriate.

Deadline:

The deadline is Monday 20 May 2024, I cannot give any extensions to this, because I have to have all the projects corrected and entered into the system by the end of that week, So pretend that the deadline is earlier (say 13 May and give yourself an extension to the 20 May 😊)

Assessment strategy for Project Type A:

You have flexibility as to what you do for this project. I understand that this can cause confusion as to what you should do, so here schedule of some of the features you may want to implement.

	Description
Minimum project	A rehash of the sample project lab (I will do a complete sample project lab around week 10). i.e. A basic Flask server that has a <ol style="list-style-type: none">1. REST API, (to perform CRUD operations)2. One database table and3. Accompanying web interface, using AJAX calls, to perform these CRUD operations
	Features you can add for more marks
	Make it look nice
	Have more than one database table
	Link to an outside data source and either: <ul style="list-style-type: none">• Save the data into a database table that you view elsewhere in your application.• Manipulate/analyse the data in real time.
	Hosted your application online (e.g. Azure, Pythonanywhere). <ul style="list-style-type: none">▪ Be warned that some free hosting does not allow access to third party systems
	Perform more than CRUD operations on your data

To get over 70% your application should be very well laid out, look good and work efficiently and well.

Project should be well laid out and easy for me to run.

Marks may be deducted for:

- Poorly formatted code, that I find hard to read (Do not over comment your code),
- If I find it hard to run,
- I find it hard to understand your GitHub layout. (a README file is handy).

Project Type B:

This is a lot more open.

I am happy to take work that you have done for your company (or yourself) that demonstrates you understand APIs.

Your project should interact with an API, or create one.

The marking scheme is a lot more open on this.

If the data you are dealing with is sensitive and so you cannot give me access to it, you may make a video of you demonstrating your application working (with the sensitive information blurred out)

Best of Luck.

Email me if you have any questions.