ASSIGNMENT 4
APIs

(handout for students)

Total: 42 marks

This APIs assignment **is creative**. We will look for and mark:

1. Good use of Git and GitHub
2. Good code practices (e.g. comments, formatting, consistent capitalisation, naming, etc.)
3. Creativity that is taken to implement the answer
4. Using constructs taught in API sessions (Flask, Requests etc.)
5. Effective use of coding constructs (correct use of techniques in a way that makes sense in the picked theme)

Total marks will be awarded for demonstrating concepts fully, there will be partial marks available for attempts.

What to submit:

- Within your *CFG-Assignments* repository, create a branch including the assignment number and title eg (assignment-4-APIs) where you will be adding your assignment work.
- Share a **link** to the GitHub repository for this assignment via Slack, **and** add your assigned instructor as a collaborator to this repository.
- Make a **pull request** for Question 1 that the instructor can review and comment on as it would be in your future workplace code review.

Top tips:

- Start early
- Read all requirements carefully before starting
- Check the mark distributions as provided below

Question 1: Build your own API

Design and implement a simple API as done in class.
Remember to come up with a unique creative problem or scenario and show real-life expected use of your database!

Marks:
A. Design and implementation of an API (33 marks)
B. Code readability, layout, and use of best practices (4 marks)
C. Creativity (5 marks)

You should:

+ Implement 2 API endpoints with appropriate functionality
+ Implement one **additional endpoint** of your choice (can be POST or GET but with a different implementation)
+ Implement client-side for each of the 3 API endpoints you have created.
+ Create a **MySQL database with at least 1 table**
+ Have a **config** file (do not leave your private information here)
+ Have **db_utils file and use exception handling**
+ Use appropriate **SQL queries** to interact with the database in your Flask application, and demonstrate at least **two different queries**.
+ In main.py have a **run() function**/call the functions to simulate the planned interaction with the API, this could include welcome statements, displaying etc., (hairdressers booking example from lesson)
+ Have correct but minimal imports per file (do not import things you do not use in the file)
+ **Document** how to run your API in a **markdown** file including editing the config file, any installation requirements up until how to run the code and what is supposed to happen.
+ Submit in GitHub as a **Pull Request**

If time permits:
Optional : Test your API - this is not part of the markscheme, but testing and Test Driven Development (TDD) are essential skills in programming.

If you cannot come up with a creative idea you can use this scenario, but **5 points out of the total mark are for creativity** and you will receive 0 for creativity:
"Restaurant reservation system API. This API should allow users to make reservations for a restaurant, view available tables, and cancel reservations if needed. Each reservation should include the customer's name, reservation date and time (saved in the DB), and the number of guests in the party. Additionally, the API should keep track of the available tables and their seating capacity."