Python – Git – SQL SkillCheck

This exercise is designed to check the knowledge and skills accumulated throughout the module. It will contain multiple processes and techniques from the lessons and is an open brief rather than a lab with steps.

The exercise has a set of requirements and starting brief that should be met, and this exercise should be done across the space of a day.

# Brief

You are hired as a DevOps engineer for a popular high street Coffee chain ‘QA Café’, who are looking for an updated order management system. They require a method of Storing and updating orders that are made in the café and require a User Interface to interact with the store of orders. In order for the order management system to be effective they need to do the following:

* Enter an order
* Read an order by ID
* Read all orders
* Update an order by ID
* Delete an order by ID
* Delete all Orders

The owners of QA Café are requesting the data for orders be stored within some form of SQL database and are open to more functionalities being added.

A template of the file structure and basic setup is hosted [here](https://github.com/Reece-elder/QAA-Python-Git-SQL_SkillCheck) and should be used as a starting point for your skill check. You can either fork this repo and rename it, copy the files over to a repo of your choice or use this as inspiration for your own repo.

# Requirements

You must consider the following basic requirements when designing your app:

* Python based App with Terminal based UI
* SQL Database, either locally hosted, cloud hosted or using SQLite
* SQL Table within your database that contains order\_id, customer\_name, drink, size, extras, price,
* Python code considering OOP when designing and using functions where appropriate
* Modular code, ideally using Controller, Service and DB objects
* Repo pushed to GitHub using proper FBM and merge requests
* Test suite using pytest for unit tests of the basic functionality
* Evidence of a linter being used to test the quality of the code
* Basic Documentation about the project and how to use in form of README.md

These are basic MVP requirements of this app; you are free to add extra functionality in whatever way you deem appropriate once the requirements have been met. Extra functionality could include:

* Graphical User Interface
* More complex table and / or relational tables
* More functionality to interact with the database
* Mock and Integration Testing
* Code analysis with SonarQube and other similar tools
* Hosting the app on a cloud server for access
* More thorough Documentation utilising Markdown language

# Deliverable

You are expected to host your solution on a public GitHub repo under the name of QAA-Python-Git-SQL\_Skillcheck so we can easily find the appropriate repo. Your code must be merged to the main branch, and you should be using proper FBM (with the use of Dev and Feature branches) throughout.

There is a mark scheme associated with this Skill Check, you will be not getting back a grade, but there will be comments associated with different aspects of the skill check you have managed to do.

You should ensure your link to your repo is submitted to the relevant activity within bud by the deadline. You should make sure your repo is public so anyone who accesses the link via bud can see through the work you have done. You should expect feedback submitted via the same message board on Bud.