

# Mini Project

# Overview

Your client has launched a pop-up cafe in a busy business district. They are offering delivered home-made lunches and refreshments to the surrounding offices. As such, they require a software application which helps them to log and track orders.

# Requirements

As a business:

- I want to maintain a collection of `products` and `couriers`.
- When a customer makes a new `order`, I need to create this on the system.
- I need to be able to update the status of an `order` i.e: `preparing`, `out-for-delivery`, `delivered`.
- When I exit my app, I need all data to be persisted and not lost.
- When I start my app, I need to load all persisted data.
- I need to be sure my app has been tested and proven to work well.
- I need to receive regular software updates.

# Technical Specifications

# User Interface

You will be building a program that runs on the command line (CLI).

- UI should be logical, clear, and simple to navigate.
- It should display a menu of options, some may be nested.
- There should be the option to exit / return to main menu.
- It should handle invalid input.

# Data Persistence

You will incrementally adopt three methods to persist data between user sessions:

- `txt`: Initially we'll store our data in plaintext files.
- `csv`: As our data changes shape, we'll need to switch to the CSV format.
- `SQL`: Ultimately, we'll finish up using a database.

# Testing

Python has some basic testing functionality built-in which we'll use to test the quality of our code. This will allow us to be confident that our app works as we intended it to.

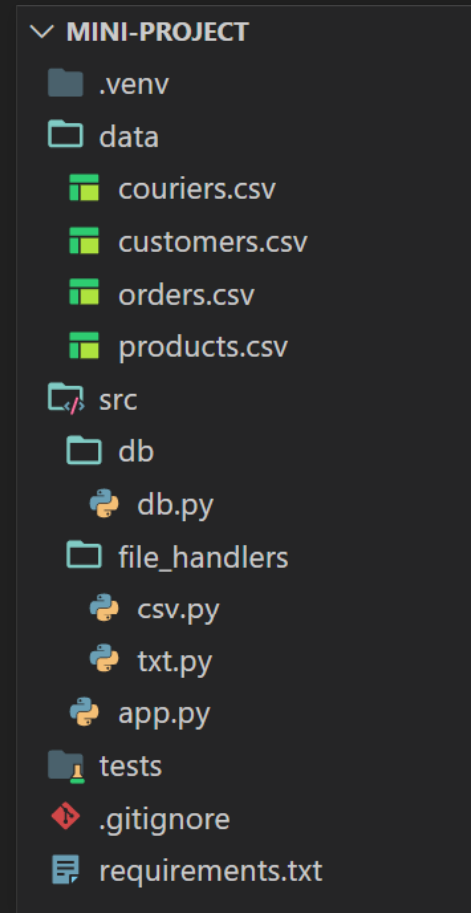
# Data Visualisation

We'll want to build some bar charts to help our client better understand the business. For this we'll use:

Jupyter Notebooks + Matplotlib



# Suggested Project Structure



# Method

- You will be allotted enough time each week to work on your project.
- Your instructor will brief you on the requirements and goals each week.
- You should each produce your own app, however pairing up with a colleague is encouraged.
- At the end you will be expected to present your finished app to a friendly panel for review.

Available Time: 6 Weeks

# Stuck?

Getting stuck on something is not fun for anyone. Due to the size of the group please follow this recommended workflow:

When stuck:

- Don't panic: Relax, take a break, think about the problem.
- Google it: There is a wealth of information on the web and you'll likely find the solution to your problem quite quickly.
- Syntax: If it's a syntax problem try referring to earlier slides, or the official Python Docs / W3 Schools.
- Ask a colleague: Ask someone in the group. It's likely they too have come accross and solved your problem.
- Ask your instructor: If all else fails, reach out to your instructor for some guidance.

# Glossary

- `CLI` Command Line Interface
- `CRUD` Create, Read, Update, Delete
- `Refactor` The process of rewriting code to *improve* it or to fix code smells
- `Code Smell` Code that doesn't adhere to best practices
- `CSV` Comma Separated Value
- `SQL` Structured Query Language
- `Data Layer` The part of your code that handles all data interaction
- `Storage Layer` The part of your code that handles all file storage
- `UI/UX` User Interface / User Experience

# Resources

- <https://realpython.com/>
- <https://blog.finxter.com/python-cheat-sheet/>
- <https://www.mysqltutorial.org/>
- <https://websitesetup.org/mysql-cheat-sheet/>

# Libraries

- pylint
- black (code-formatter)
- pymysql
- matplotlib

