Interview Assignment - Python & UI

Assignment

Description

You are a Customer Account manager of a manufacturing company. Your customers place orders and you need to plan order execution accordingly.

You have visibility to future supply for a few days. So in order to better serve your customers, you decide to fulfil customer orders as soon as possible, whenever supply becomes available from the factory.

In addition, you would like to execute the orders in a first-come-first-serve manner. That is, you will fulfil the customer order (or partial of an order) based on order date, until supply is exhausted.

In case multiple customers place orders on the same day, some allocation rules should be applied (**please propose your own rules**) so that your customers don't feel being mistreated.

Input

You will read three input files:

1. a demand order table that stores customers, their order dates and order quantities
2. a supply table that stores projected supply from each factory for X number of days
3. a sourcing table that maintains the sourcing rules, i.e. which customer can source from which factory

Please refer to the files 'demand_order.csv', 'supply.csv', and 'sourcing_rule.csv' for examples.

Output

You will generate an allocation result, indicating the factory, customer, and planned execution quantity on each day. You will need to pivot the dates to columns so that your data becomes more readable.
Please refer to the file 'order_execution_plan_horizontal.csv' for example.

Requirements

You need to complete the assignment with below requirements:

- A **Python web application** (Flask, bottle, etc.) allowing user to submit csv files and/or input arguments for processing.

- The application should be able to render the result on the UI and allow users to download/ access the result when the process is done.
- **Design test cases as complete as you can** to cover different scenarios (over supply, over demand, order on the same date, etc.); if possible, automate your test execution.
- Engineering excellence (good documentation practice, configuration, testing, coding convention, etc.).

Good to have

- Styled UI
- Validation of input files & arguments
- Deploy the web application on a cloud
- Any other additions you think will “surprise and delight” your code reviewer.

Hints

- Specify your assumptions to the problem.