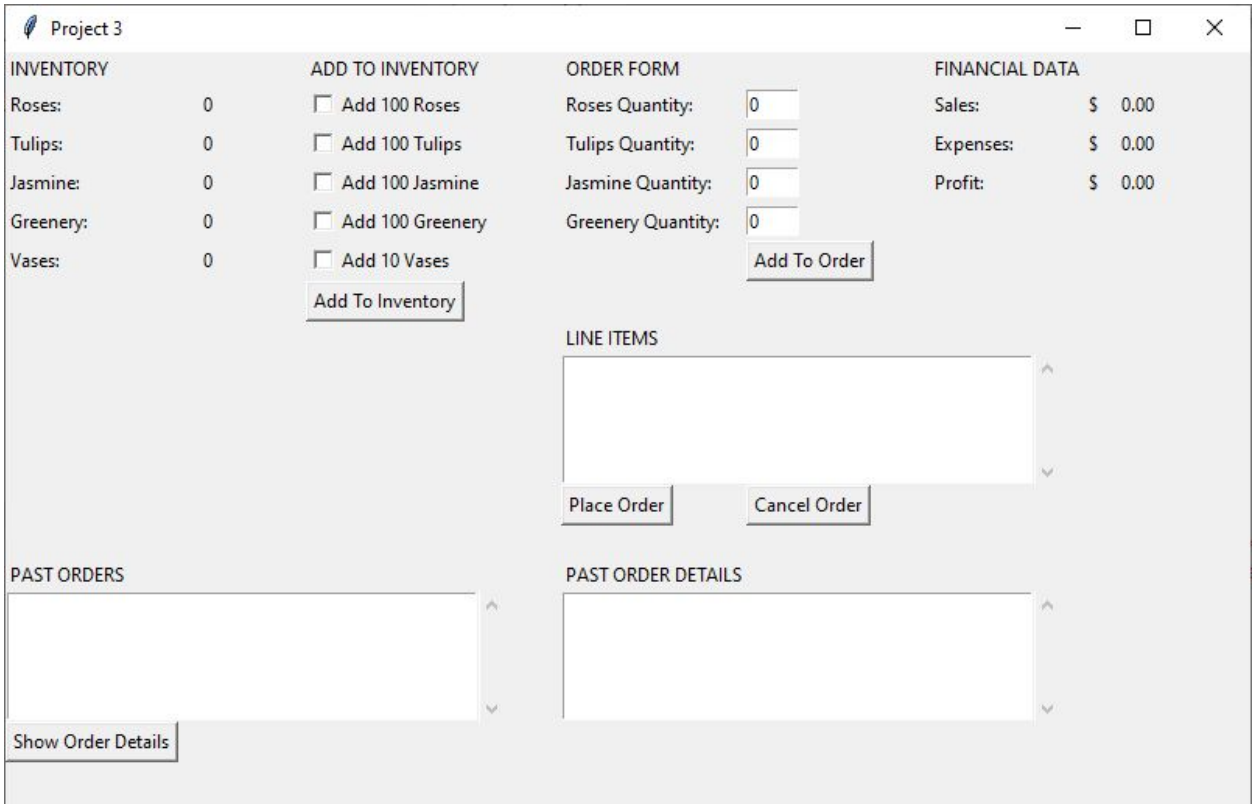
Project Overview

This project is a continuation of Projects 1 and 2. You are to develop a graphical application, using Python, intended to help keep track of inventory, sales, expenses, and profit for a flower shop.

This project will be a team based project.

Project Details

The following is a sample interface for your project. You may use this design or another design provided the interface meets the requirements of the project.



The interface must have the six following sections:

- Inventory/Add to Inventory
- Order Form
- Line Items
- Financial Data
- Past Orders
- Past Order Details

Inventory / Add to Inventory

The Inventory will reflect how many units the flower shop has on hand for the various items.

The user will check the box to indicate which item they want to add to inventory when the Add To Inventory button is clicked. The following table will show how many units are to be added and at what expense:

Item	Units	Cost
Roses	100	$100.00
Tulips	100	$75.00
Jasmine	100	$85.00
Greenery	100	$65.00
Vases	10	$50.00

When the user adds to the inventory, be sure to update the inventory values on the form as well as the expenses and the profit. Profit is calculated by subtracting the expenses from the sales.

The inventory values must be stored within the database to ensure the values are retained after the program is closed.

Order Form

For the order form, the user may select how many of each item they want within the vase for their order. The user can then add the vase configuration to their overall order as a line item. Each vase increases our sales by $30.

Order form data does not need to be retained should the program be closed.

Line Items

When the user clicks Add To Order, show a line item in the list that provides details about what is going to be added to the vase. Setup the output so that it is clear to the user what has been added to the line item.

When the user clicks Place Order, do not allow the order to go through should it result in a negative inventory. A warning must appear to the user and no inventory times or sales should be updated. Give the user the ability to either add additional inventory or cancel the order. Only when there is sufficient inventory should the order go through.

Line Items data does not need to be retained should the program be closed.

Financial Data

Each time the user adds to inventory, expenses will be incurred. Each time the places an order, sales will be increased by $30 per vase. Be sure to update the output, including the profit. The profit is calculated by subtracting the expenses from the sales.

The financial data must be stored within the database to ensure the values are retained after the program is closed.

Past Orders

The past orders will be a list of all unique past orders that have been placed. The user may select a past order to view the details of that order.

The past orders must be stored within the database to ensure the values are retained after the program closes.

Past Order Details

When the user selects a past order to review, the past order details will show the line items from that particular order.

The past orders details must be stored within the database to ensure the values are retained after the program closes.

Submissions

Team Roster

This project will be a team project. Teams must consist of a minimum of three members and a maximum of six members. The Team Roster must be in the form or a Word Document or a PDF. It must include the full name and RIT username of each member on the roster.

Those students not on a roster after the deadline will be placed within existing teams at random.

Only one member of the team needs to submit the group roster.

Project Code

Submit your Python file to the **Project 3 \ Code** assignment folder on myCourses.

In the case of this project, only one member of your team is required to submit their team's code. The instructor will use the last submitted set of code when grading.

There will be no late submissions for this project.

Database Design

Should you use a database design that is different from the one provided by the instructor, you must submit that database as well for testing purposes. If no database is submitted, the instructor will assume your team is using the provided template and grade based on using that database.

Peer Review

Start by downloading the peer review template provided by the instructor.

In the case of the peer review, each team member must submit their own peer review for credit. Failure to submit a peer review may result in a zero for the entire project for that individual member, regardless of the team grade.

The Peer Review must be either a Microsoft Word Document or a PDF and must use the template provided.

Grading Criteria

Your application will be graded based on the following criteria:

- Project must be graphical. Output to the terminal must be limited to debugging only.
- Your application meets the requirements of the project.
- Your application cannot crash or cause errors should the user enter invalid data.
- If you create your own user interface, it must be intuitive.

All graded material must be submitted to the designated Assignment folder. Other methods of submission will not be accepted.

While this is an individual project, you may work with your classmates to assist and learn from each other. Each student is expected to submit their own work. Do not submit someone else's work as it may result in a zero grade for the project.

You may submit to the Assignment folder multiple times. The instructor will simply grade the last submission. If you make a mistake and realize it, just resubmit. Each submission is automatically timestamped so that the instructor will know which one is the last submission.