Project Title:

Inventory Management System with GUI using Python

Problem Statement:

In small businesses or personal setups, tracking stock items manually becomes inefficient and error-prone. This project aims to build a simple, user-friendly inventory management tool with a graphical interface that allows non-technical users to manage stock efficiently.

Project Overview:

This is a desktop-based **Inventory Management System** built using **Python's Tkinter GUI library**. It enables users to **add, update, delete, search, and export** inventory items. The application also displays all items in a live-updating table and allows exporting the inventory to a .csv file for backup or further analysis.

Key Features:

- **Add Items:** Add new items with name, quantity, and price.
- **Update Items:** Modify existing item details (quantity or price).
- **Search Items:** Retrieve and autofill item details in the form.
- **Delete Items:** Remove items from the inventory.
- **Export to CSV:** Save inventory data to a .csv file with quantity × price totals.
- **Live Table View:** Inventory table updates instantly on every action.
- **Form Validation:** Prevents invalid or incomplete inputs using input checks and pop-up messages.

Technologies Used:

Tool/Tech	Purpose
Python 3	Core programming language
Tkinter	GUI development
CSV module	Exporting inventory data

Tool/Tech	Purpose
ttk.Treeview	Displaying tabular data in the GUI

Logic and Functionality:

- All inventory data is stored temporarily in a Python dictionary (inventory = {}), where each key is an item name and the value is a dictionary containing quantity and price.
- Form inputs are validated before any action is performed (e.g., checking if price is a number, quantity is an integer, and all fields are filled).
- GUI components like Entry, Label, and Button are used to interact with the user.
- A Treeview widget is used to render a live table view that refreshes after every inventory operation.
- The "Export CSV" button writes all inventory data to a inventory_export.csv file.

🧪 **What You Learned:**

- How to build a real-world GUI using Tkinter
- Data structure handling using dictionaries
- Input validation and user interaction with pop-up messages
- How to use Treeview for dynamic table rendering
- File operations for exporting data (csv.writer)
- GUI application design patterns and event-driven programming

---

How to Explain It in Interview (1–2 min):

"I developed a GUI-based Inventory Management System using Python and Tkinter, designed for small businesses to track items, quantities, and prices. The app features real-time table updates, the ability to add, update, delete, search items, and export the entire inventory to a CSV file. I used a dictionary to hold inventory data in memory, performed input validation, and displayed everything dynamically using the Treeview widget. This project helped me improve my understanding of GUI design, file handling, and form validation in Python."