**Assignment: Employee Management System in Python**

**🎯 Objective**

Create a Python-based **Employee Management System** that allows users to add, view, update, and delete employee records. The system should store data persistently using a file or database. This project will help you practice Python fundamentals, modular design, and optionally explore GUI development.

**📌 Requirements**

**1. Core Features**

* **Add Employee**: Input employee details (ID, Name, Email, Department, Salary).
* **View Employees**: Display all employee records.
* **Update Employee**: Modify existing employee details by ID.
* **Delete Employee**: Remove an employee by ID.

**2. Data Storage**

Choose one of the following:

* **Option A**: Store data in a **JSON** or **CSV** file.
* **Option B**: Use **SQLite** for persistent storage.

**3. Required Enhancements**

These enhancements are **mandatory**:

* ✅ **Search/Filter Functionality** (by name, department, or status)
* ✅ **Input Validation and Error Handling**
* ✅ **Mark Employee as Active/Inactive**
* ✅ **Export to CSV**
* ✅ **User-Friendly CLI** (use argparse, colorama, or rich)

**🧱 Code Structure**

employee\_management/

├── main.py # Entry point of the application

├── employee.py # Employee class (data model)

├── manager.py # Business logic (add, update, delete, search)

├── storage.py # File or DB operations (JSON, CSV, or SQLite)

├── gui/

│ ├── \_\_init\_\_.py

│ ├── app.py # Main GUI window

│ ├── add\_employee.py # Add employee form

│ ├── view\_employees.py # View/search/update/delete UI

│ └── styles.py # Optional: styling for GUI widgets

└── utils.py # Helper functions (e.g., validation, formatting)

**▶️ How to Run the Application**

**🧩 Prerequisites**

Make sure you have the following installed:

* Python 3.8 or later
* Required libraries (install via pip)

**🖥️ Option 1: Run the CLI Version**

Follow the on-screen menu to:

* Add a new employee
* View all employees
* Update or delete employee records
* Export data to CSV

**🪟 Option 2: Run the GUI Version (Tkinter)**

The application window will open, allowing you to:

* Add new employees via a form
* View and manage employee records in a table
* Search/filter employees
* Export data

**✅ Deliverables**

* A GitHub repository with:
  + Complete source code
  + A README.md file (this one)
  + Screenshots (if GUI is implemented)
* A short demo video (optional but encouraged)

**🧠 Tips for Using GitHub Copilot**

* Use comments like # function to update employee by ID to guide Copilot.
* Let Copilot assist with class definitions, file I/O, and CLI enhancements.
* Use Copilot to refactor or suggest improvements.