CAPSTONE PROJECT: File-Based To-Do App

You are building a simple command-line To-Do application.

Unlike normal programs, tasks must not disappear when the program closes.

All tasks must be saved in a file and loaded again when the app restarts.

This is how real apps maintain persistent data.

What Students Will Use:

- Reading from files
- Writing to files
- Using functions for reusability
- Lists + loops + if-else
- Basic app design logic
- Real-world persistence concept

File Design

All tasks will be stored in a text file: tasks.txt

Each line represents one task.

Functional Requirements

You must implement the following functions:

1 load_tasks()

* Reads tasks from `tasks.txt`
* Returns a list of tasks
* If file does not exist, return an empty list

---

2 save_tasks(tasks)

* Takes a list of tasks
* Writes them back to `tasks.txt`
* Each task on a new line

---

3 add_task(tasks)

* Ask user for a new task
* Add it to the list
* Save updated list to file

---

4 view_tasks(tasks)

* Display all tasks with numbering
* Handle empty list gracefully

---

5 delete_task(tasks)

* Ask user which task number to delete
* Remove it from list
* Save updated list to file

Menu System (Loop Required)

The app should repeatedly show:

```text
1. Add Task
2. View Tasks
3. Delete Task
4. Exit
```

```

The program should run until user chooses Exit.

---

## Example Interaction

```text
1. Add Task
2. View Tasks
3. Delete Task
4. Exit

Enter choice: 1
Enter task: Complete Python assignment
Task added successfully!

Enter choice: 2
1. Complete Python assignment

Enter choice: 4
Exiting application...
```

```

Rules for Students

* Do NOT use external libraries
* Do NOT use classes (functions only)
* Use file handling (`open`, `read`, `write`)
* Use loops and if-else

Student Should Understand:

* Why files are needed
* Difference between memory vs disk
* This is similar to real apps (notes, reminders, logs)