**Python Mini Project – To-Do List Manager**

**NAME:** T. Dharani

**🎯 Objective**

The objective of this mini project is to develop a **Command-Line and Web-Based To-Do List Manager using Python** that allows users to efficiently manage their daily tasks. The system provides functionality to add, edit, delete, and mark tasks as completed, along with features like setting due dates and assigning priority levels. The project emphasizes the use of Python programming concepts such as data structures, file handling using JSON, modular programming, and Flask for building a user-friendly web interface. It aims to demonstrate practical problem-solving and application development skills in a real-world scenario.

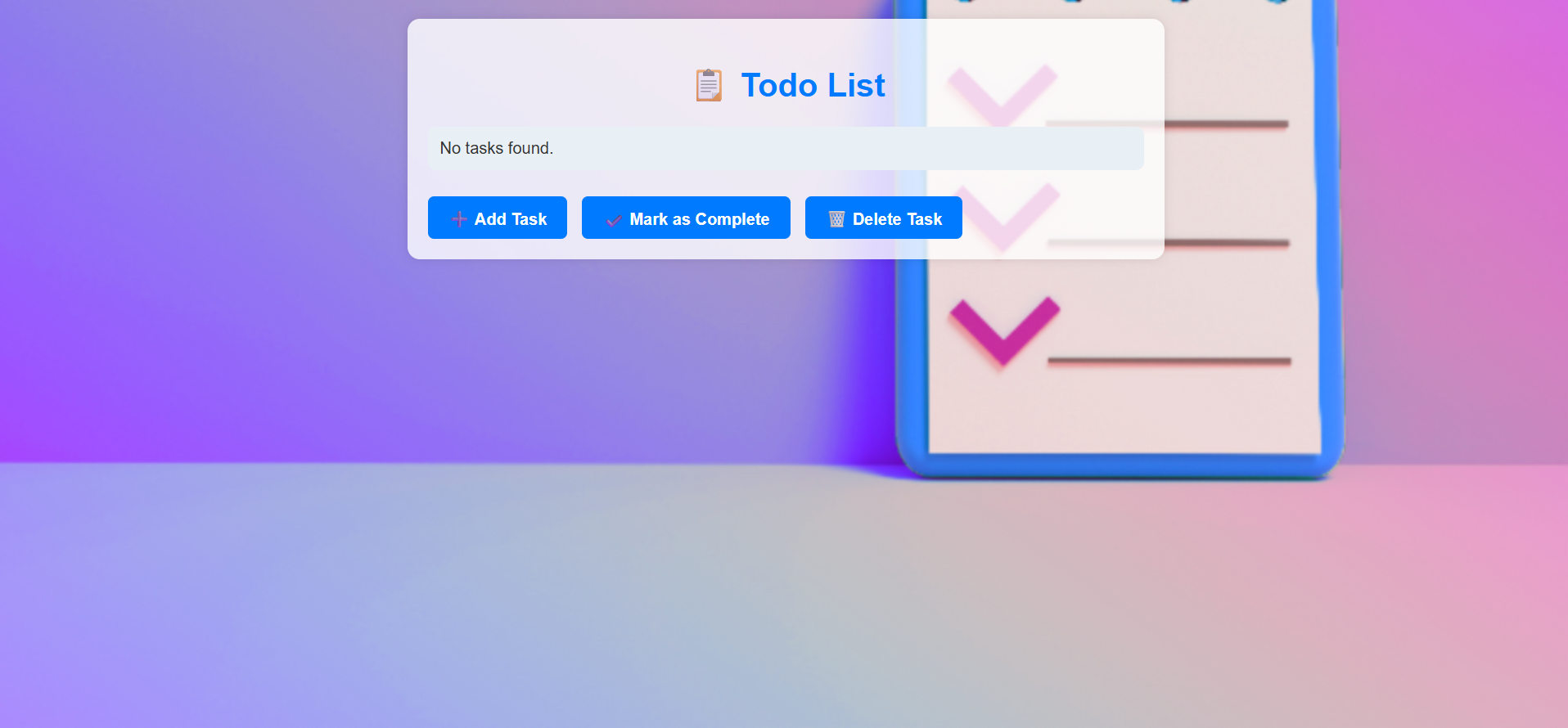
🧠 **Features Implemented**

* Add, delete, edit tasks
* Mark complete
* Save/load using JSON
* Priority, due date

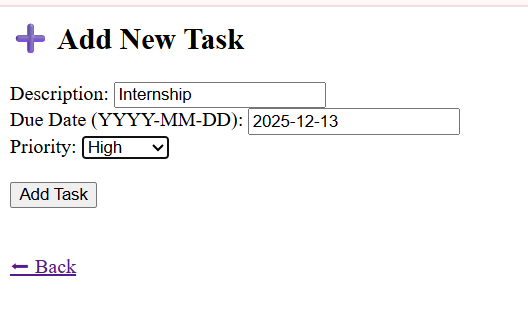
🧩 **Technologies Used:** Python, Flask, JSON

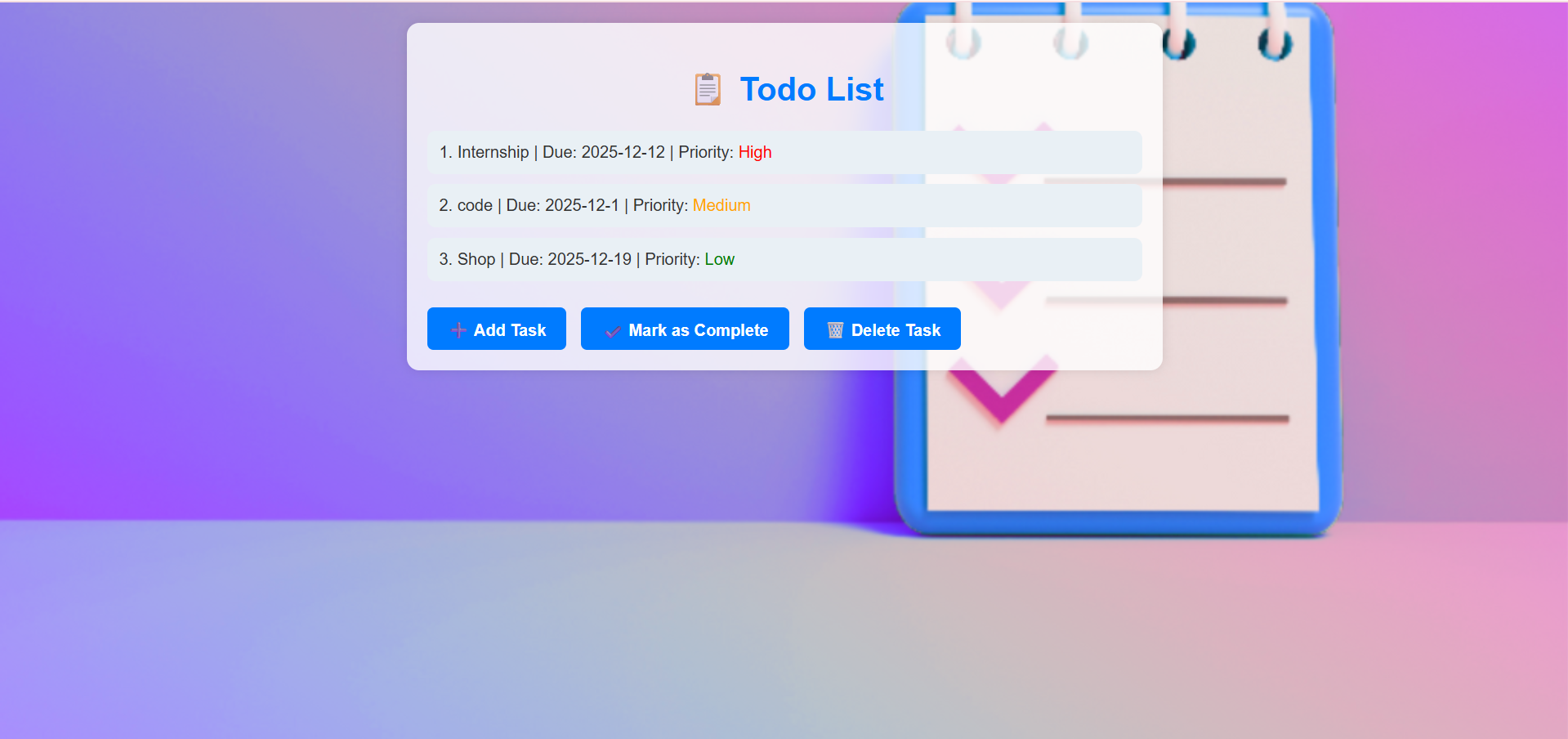
🖼️ **Screenshots:**

**Userinterface(Home screen)**



**AddTasks**

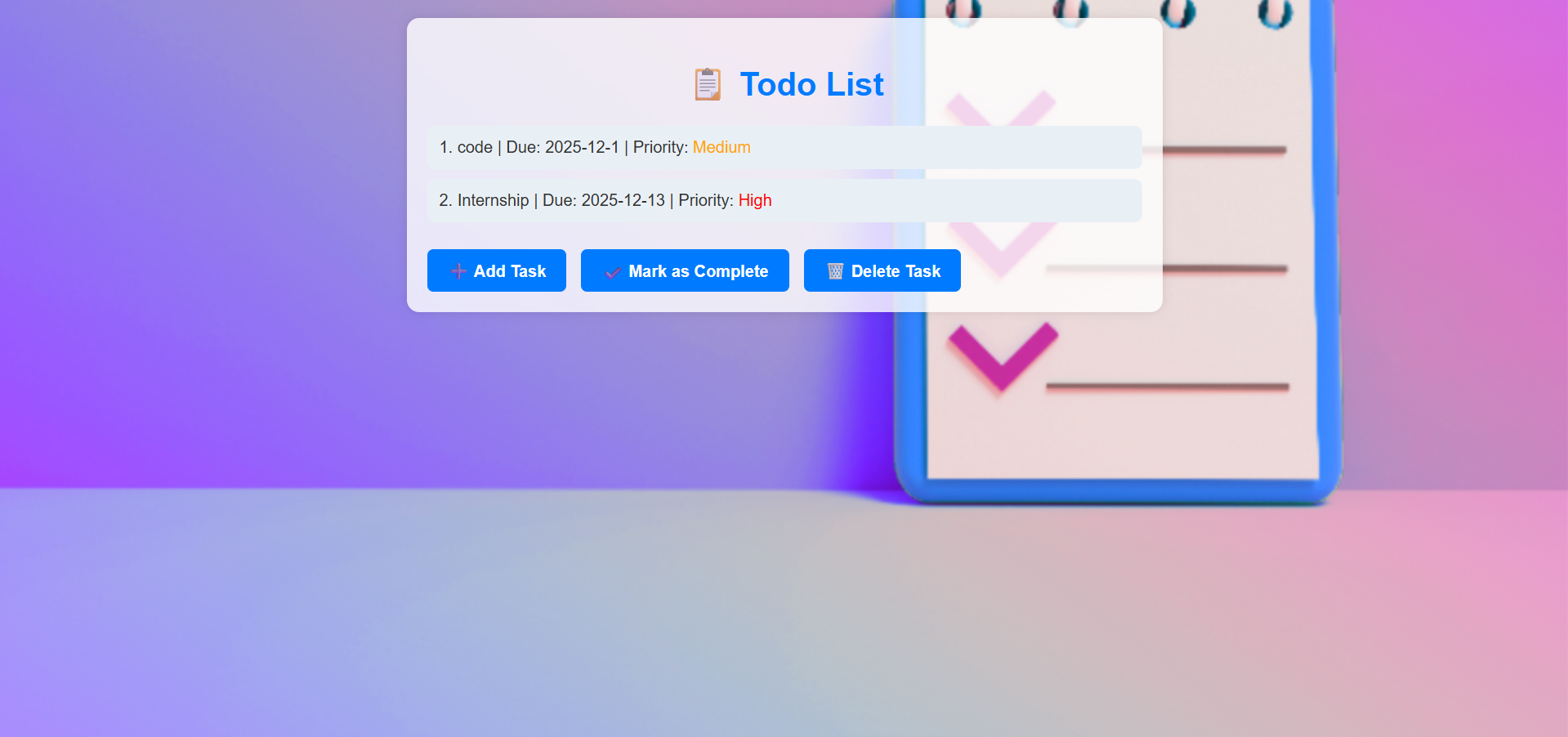
****

**View Tasks page **

**Mark complete page**

****

**Delete complete page**

****

**✅ Conclusion**

**🔍 What I Learned (Python Domain):**

* Strengthened my understanding of core Python concepts such as lists, dictionaries, functions, and conditional logic.
* Gained hands-on experience with file handling using JSON, enabling data persistence between sessions.
* Learned to write modular and maintainable code by organizing logic into separate Python files (app.py and todo\_manager.py).
* Explored how to build a basic web application using the Flask framework, which deepened my skills in backend Python development.
* Understood the importance of user input validation, status tracking, and task filtering using Python logic.

**🚀 What I’d Improve with More Time:**

* Add user authentication for multi-user task management.
* Replace JSON with a database (SQLite) for better performance and data management.
* Enhance the web UI using Flask templates (Jinja2) and CSS styling.
* Integrate notifications or alerts using Python’s scheduling libraries or APIs.
* Add unit tests to ensure code quality and robustness.