ToDo -list App (using python and Sql)

**Problem Statement**

In the modern fast-paced world, people often struggle to keep track of their tasks, deadlines, and priorities. Managing time effectively is a challenge when there is no structured system to track to-dos, which can lead to missed deadlines and decreased productivity. There is a need for a simple and efficient tool to help users manage their daily tasks.

**Project Abstract**

This project is a command-line based **To-Do List Manager** that allows users to add, view, update, and delete tasks. It uses **SQLite** as the backend to store tasks persistently. Users can set due dates and priorities and mark tasks as completed once done. The application is lightweight and easy to use, ideal for personal task management.

**Proposed Solution**

We propose a **CLI-based To-Do Application** that:

* Stores tasks in a local SQLite database
* Allows users to:
  + Add new tasks with due dates and priorities
  + View all tasks sorted by completion status and priority
  + Mark tasks as completed
  + Delete tasks when no longer needed

This provides an offline, easy-to-use personal task tracker without any setup overhead or external dependencies.

**Database Design**

**Database Name**: todo. dB  
**Table Name**: task

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| id | INTEGER | Primary key, auto-incremented |
| task | TEXT | Task description (not null) |
| due\_date | TEXT | Due date in YYYY-MM-DD format |
| priority | TEXT | Priority of the task (High/Med/Low) |
| completed | BOOLEAN | 0 for pending, 1 for completed |

**Flow Chart**

+-------------------------+

| Start Program |

+-----------+-------------+

|

v

+-----------+-------------+

| Display Menu |

+-------------------------+

|

+----------------+-----------------------------+

| | |

v v v

Add Task Display Tasks Mark Task as Completed

| | |

v v v

Insert into DB Fetch & Print Update 'completed'

| Task Details Flag in DB

| | |

+----------------+-----------------------------+

|

v

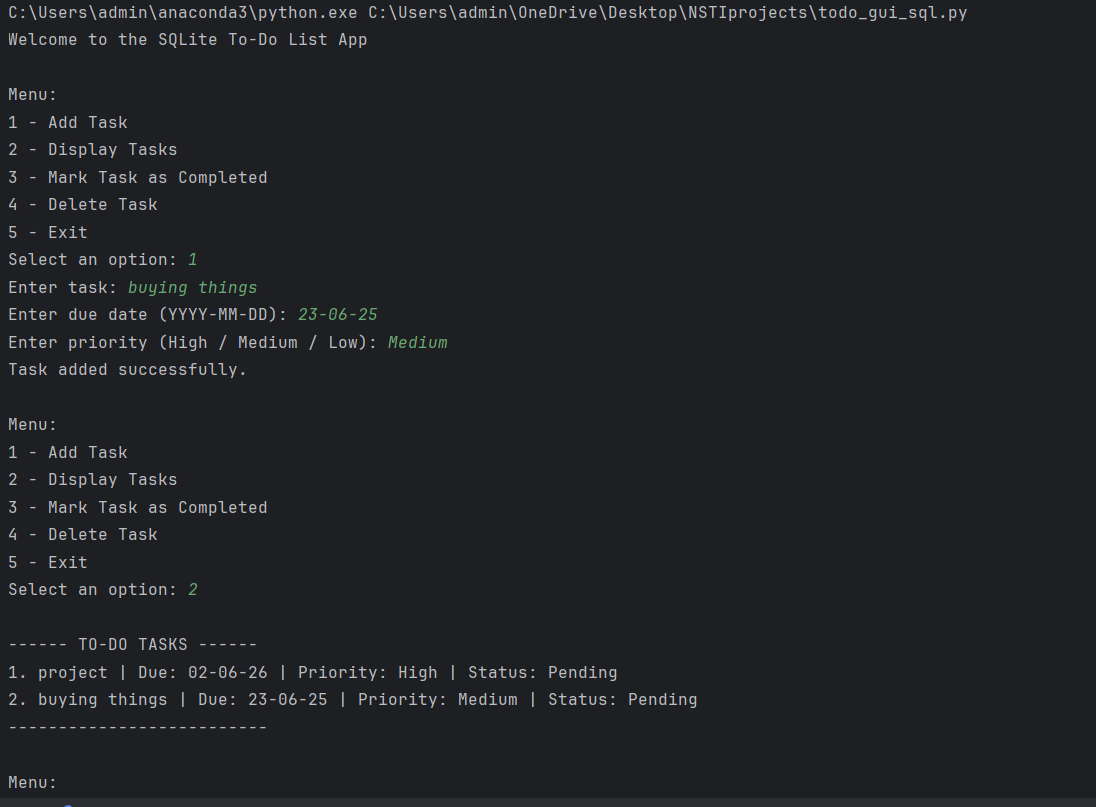
Delete Task / Exit

|

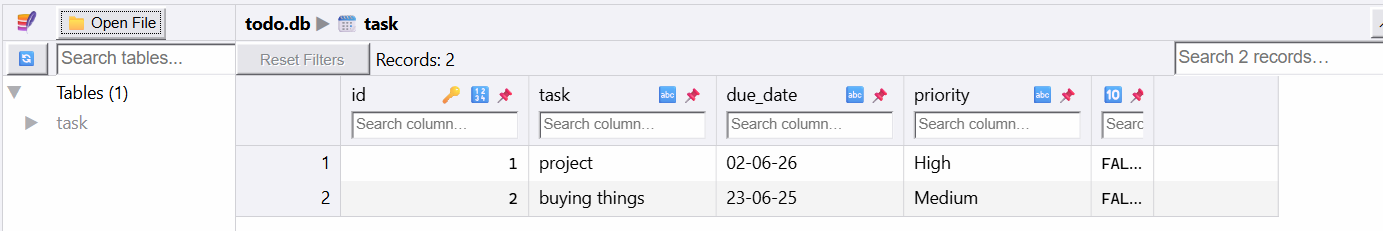
Delete from DB / End Program

Screenshots

Output:



Database:



**Future scope**

* **GUI Version**: Develop a graphical interface using Tkinter or PyQt.
* **Web Version**: Extend the app to run in a browser with Flask/Django backend.
* **Notification Support**: Remind users of due tasks via system notifications or email.
* **Task Categories**: Add support for grouping tasks (Work, Personal, etc.).
* **Recurring Tasks**: Support for daily/weekly repeating tasks.
* **Search/Filter**: Allow filtering tasks by date, priority, or status.