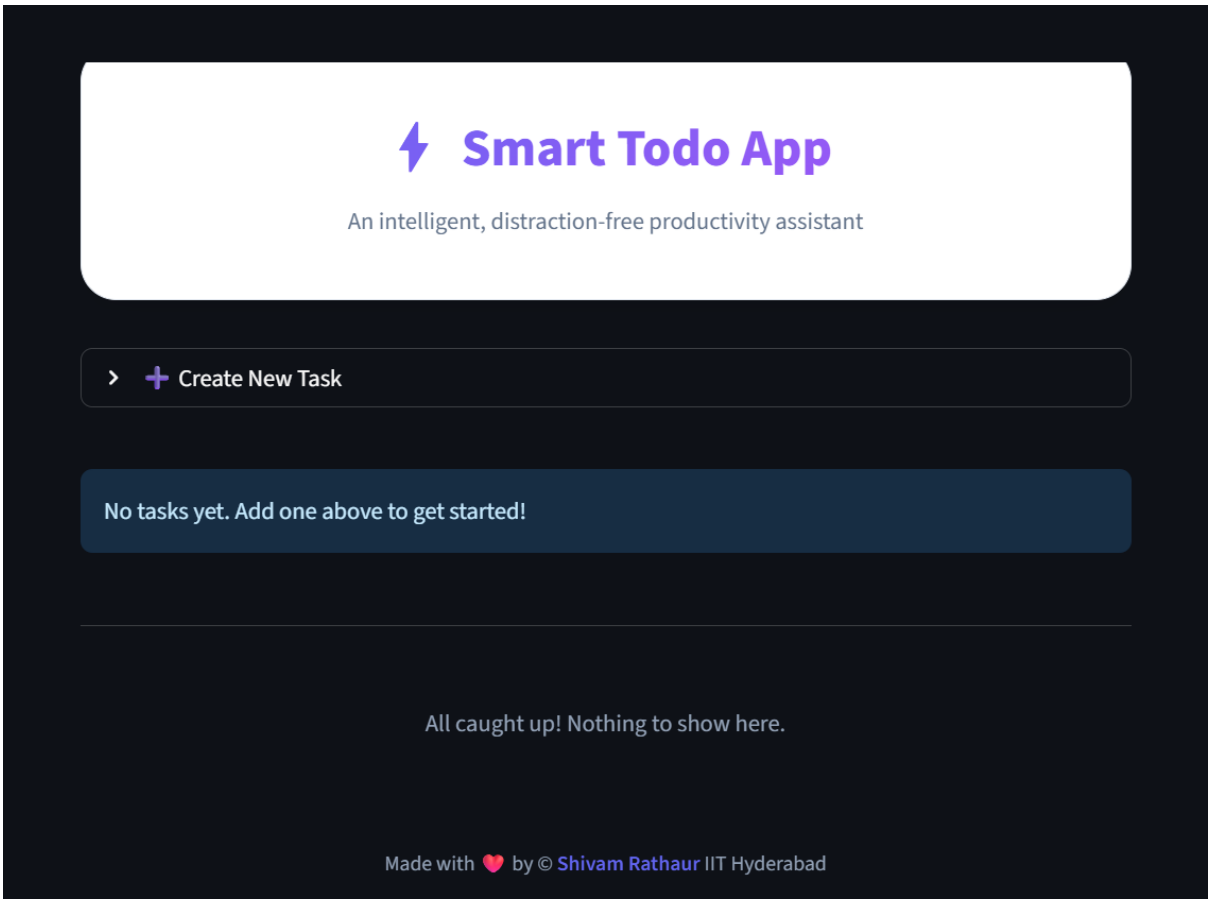
Smart Todo App

Application Explanation Document



Name: Shivam Rathaur
Institute: Indian Institute of Technology Hyderabad
Technology Stack: Python, Streamlit, SQLite, Generative AI (Gemini API)

Introduction

This document explains the design, functionality, and implementation of the **Smart Todo App**.

The objective of this application is to provide a clean, user-friendly task management system enhanced with **Generative AI** to assist users in overcoming task-related procrastination. The app emphasizes robustness, usability, and secure handling of external APIs.

Technology Stack & Architecture

- **Technology Stack**
  - **Python:** Core application logic
  - **Streamlit:** Frontend UI and interaction handling
  - **SQLite:** Lightweight local database for persistent task storage
  - **Generative AI (Gemini API):** AI-based task analysis feature
  - **Custom CSS:** Used to enhance UI appearance and user experience
- **Architecture Overview**

  The application follows a modular design:

  - `app.py`
    Handles UI rendering, user interaction, session state management, and integration of all components.
  - `database.py`
    Manages all SQLite operations such as creating tables, adding tasks, updating task status, editing tasks, and deleting tasks.
  - `ai_utils.py`
    Encapsulates all Generative AI logic, including secure API access through environment variables and graceful fallback handling.

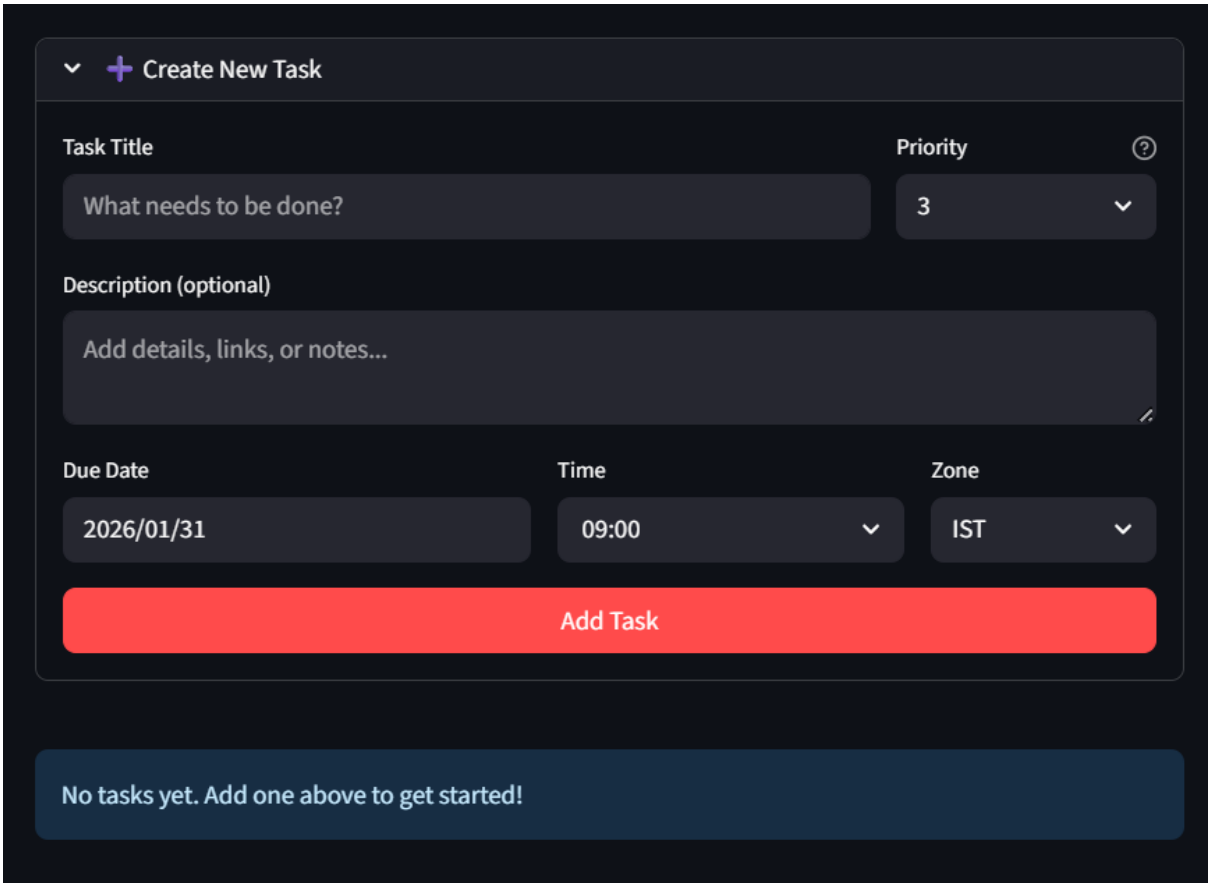
  This separation ensures maintainability, readability, and robustness.

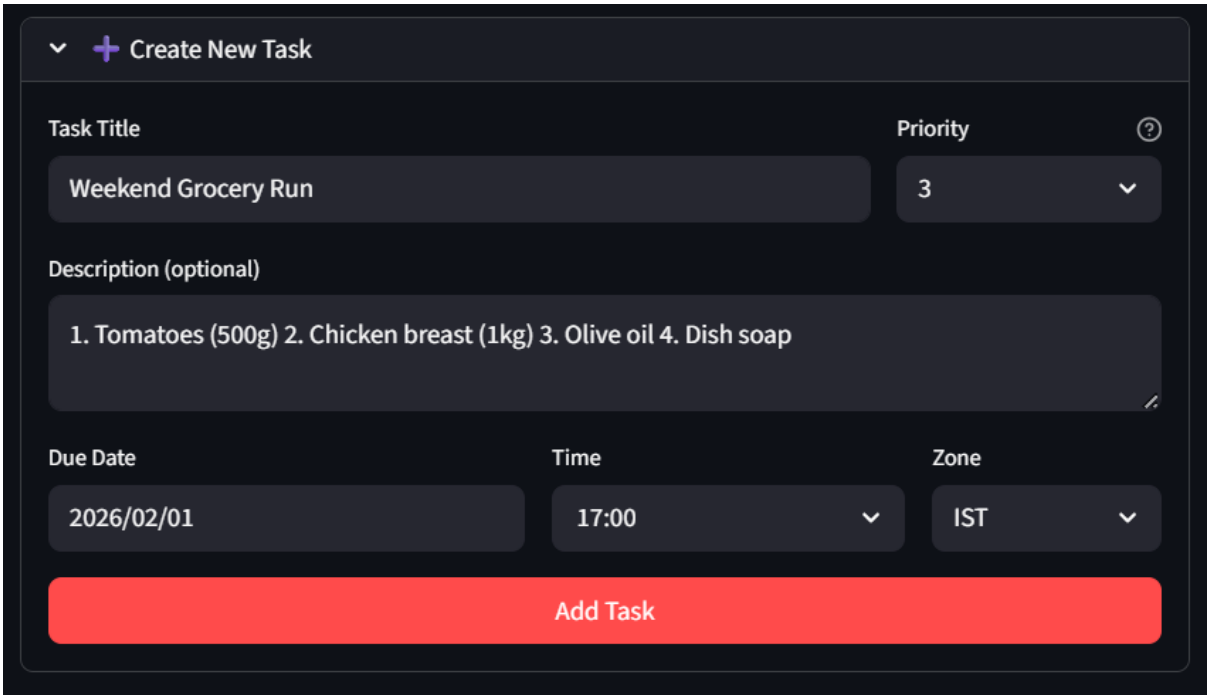
Core Features

- **Task Management**

  The application allows users to create detailed tasks with the following attributes:

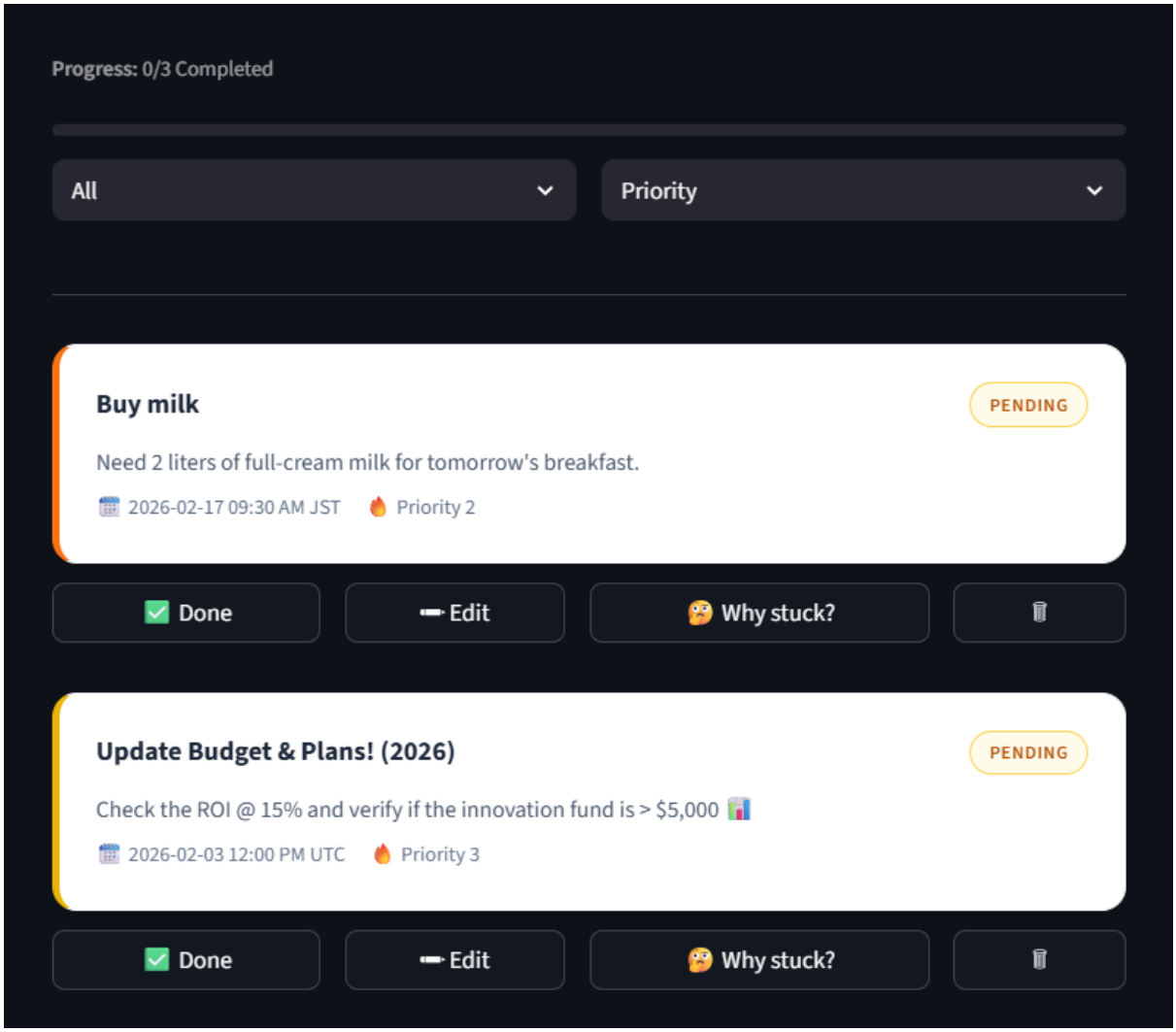
  - Task Title (required)
  - Description (optional)
  - Priority level (1 = highest, 5 = lowest)
  - Due date, time and time zone support


Create New Task
Task Title
What needs to be done?
Priority
3
Description (optional)
Add details, links, or notes...
Due Date
2026/01/31
Time
09:00
Zone
IST
Add Task
No tasks yet. Add one above to get started!


Create New Task
Task Title
Weekend Grocery Run
Priority
3
Description (optional)
1. Tomatoes (500g) 2. Chicken breast (1kg) 3. Olive oil 4. Dish soap
Due Date
2026/02/01
Time
17:00
Zone
IST
Add Task

Users can:

- Edit tasks inline without leaving the main view
- Mark tasks as completed or undo completion
- Permanently delete tasks

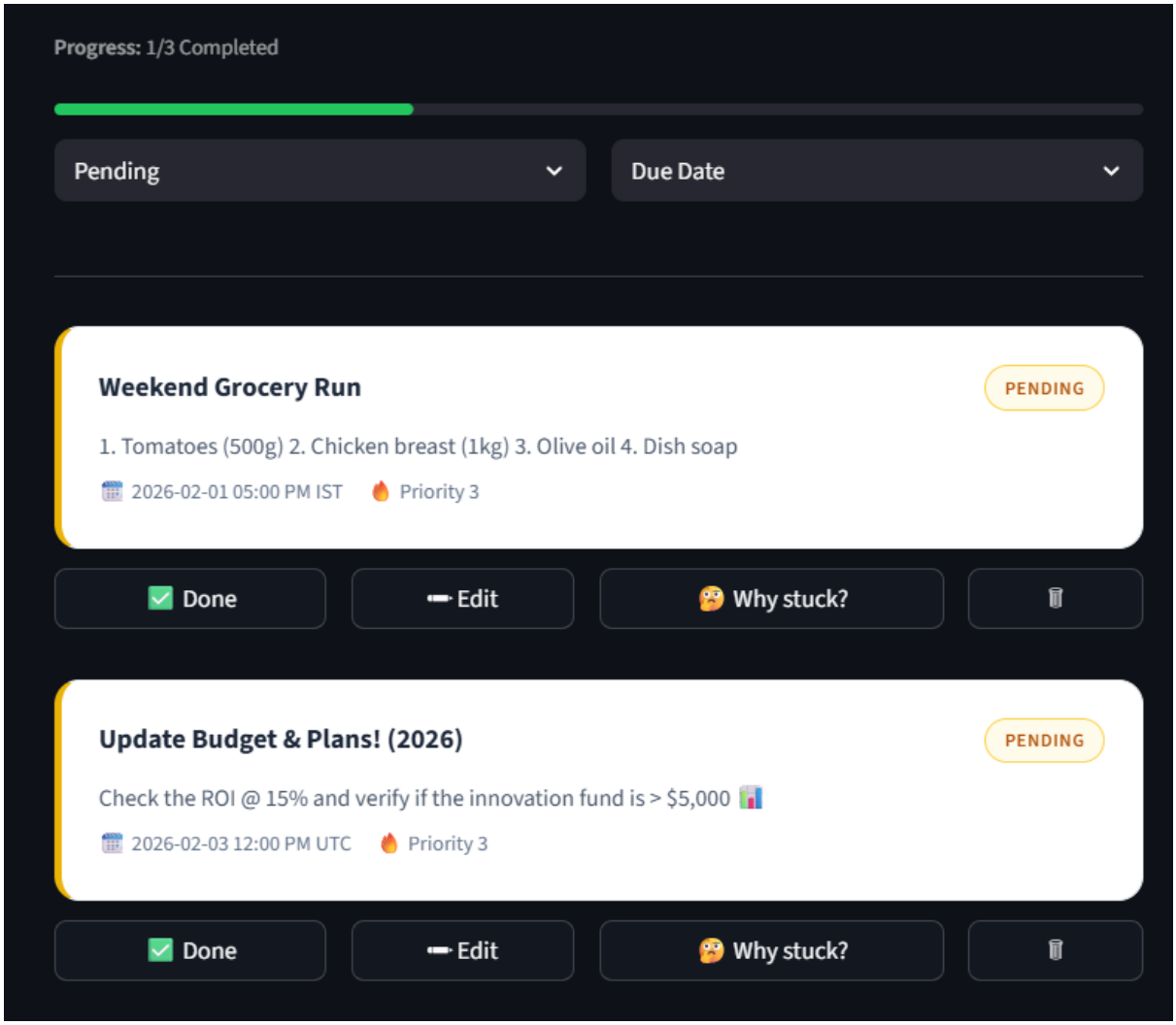


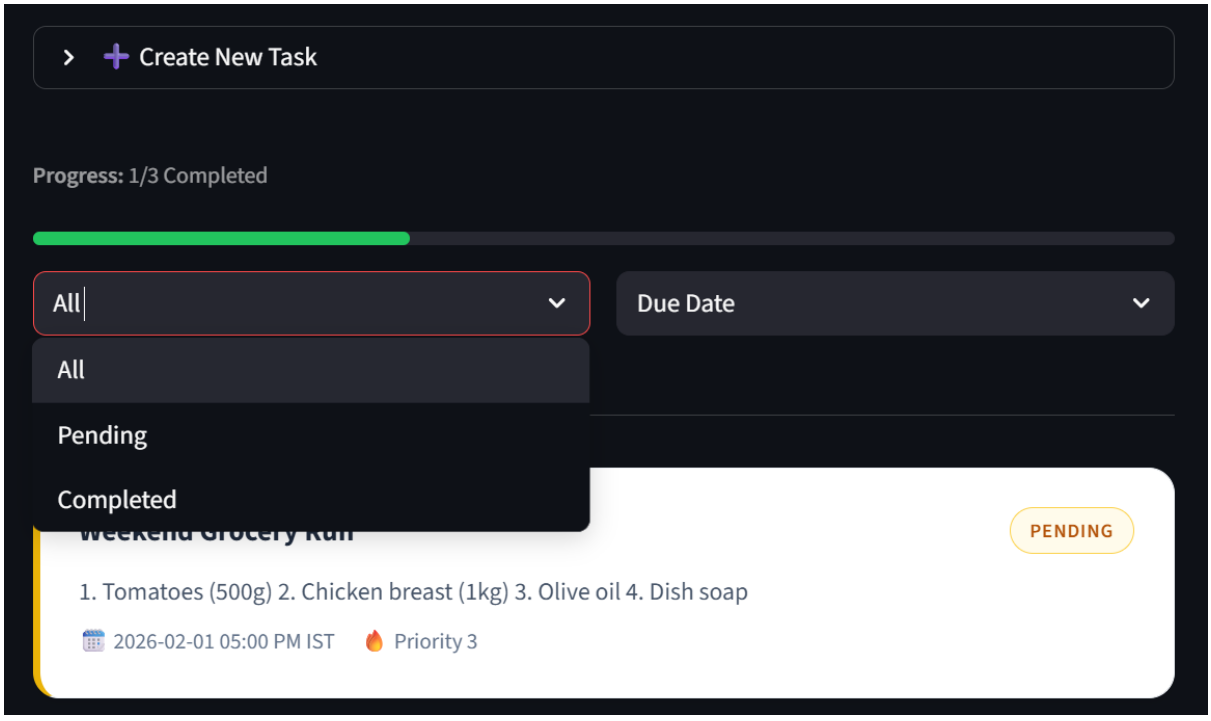
➢ **Task Organization & Tracking**

To help users stay organized, the app provides:

- **Priority-based visual indicators** using color-coded borders
- **Sorting options** by priority or due date
- **Filtering options** to view all, pending, or completed tasks
- **Progress bar** displaying overall task completion percentage

These features make it easy to track progress at a glance.


Progress: 1/3 Completed
Pending
Due Date
Weekend Grocery Run
PENDING
1. Tomatoes (500g) 2. Chicken breast (1kg) 3. Olive oil 4. Dish soap
2026-02-01 05:00 PM IST
Priority 3
Done
Edit
Why stuck?
Update Budget & Plans! (2026)
PENDING
Check the ROI @ 15% and verify if the innovation fund is > $5,000
2026-02-03 12:00 PM UTC
Priority 3
Done
Edit
Why stuck?


Create New Task
Progress: 1/3 Completed
All
Due Date
All
Pending
Completed
PENDING
1. Tomatoes (500g) 2. Chicken breast (1kg) 3. Olive oil 4. Dish soap
2026-02-01 05:00 PM IST
Priority 3

Generative AI Integration

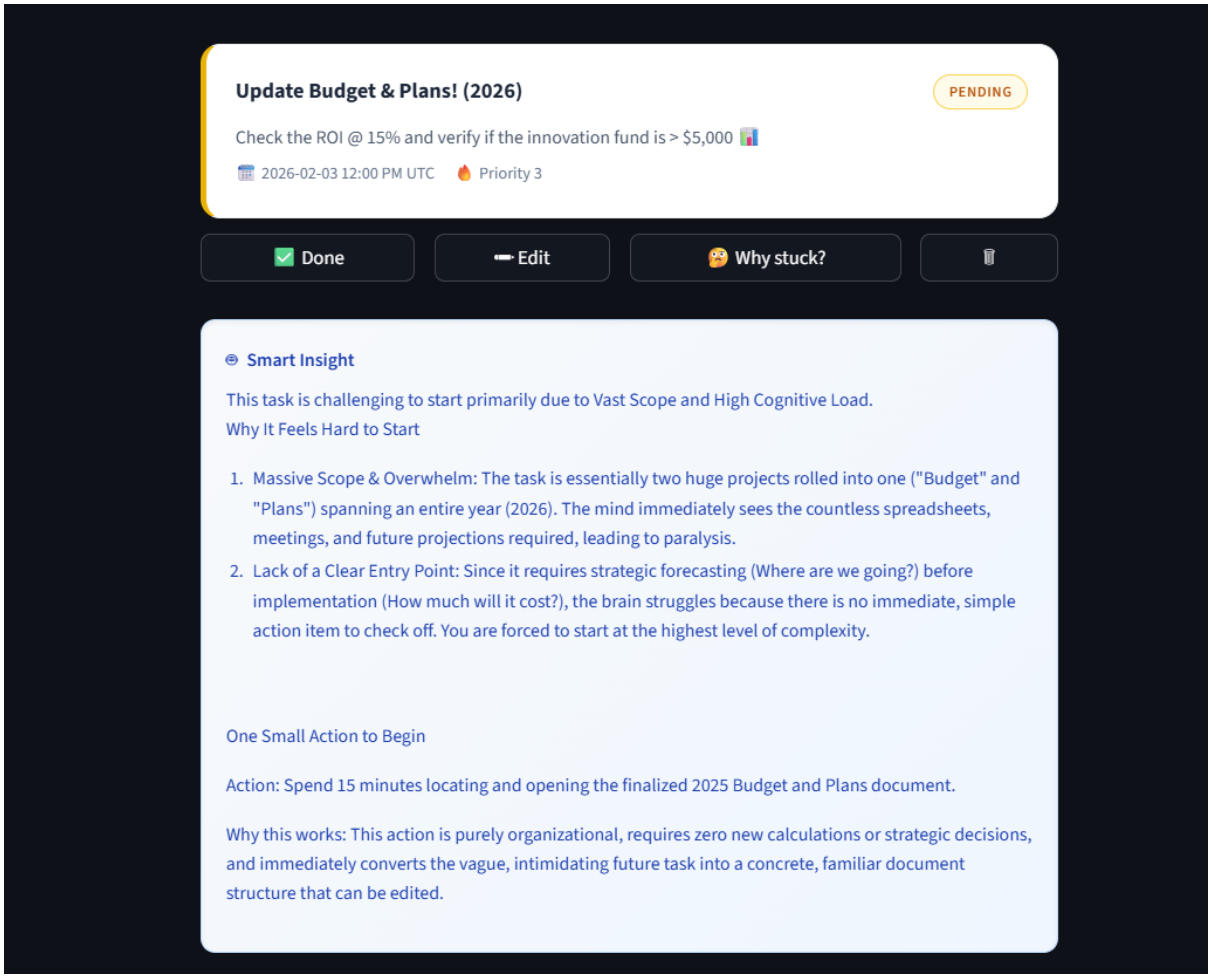
- **"🤔 Why Stuck?" Feature**

  The app includes a Generative AI feature designed to assist users when a task feels difficult to start.

  When the user clicks **"🤔 Why stuck?"**, the AI:

  - Analyzes the task title
  - Provides a short, human-friendly explanation of why the task may feel challenging
  - Suggests a small, actionable step to help the user get started

  This feature is intentionally designed to be supportive rather than intrusive.



- **Robust AI Error Handling**

  To ensure stability:

  - API keys are **never included in the source code**
  - Keys are accessed only via environment variables

- If the AI service fails (e.g., quota exceeded or network issue), the app displays a **graceful fallback message** instead of raw errors

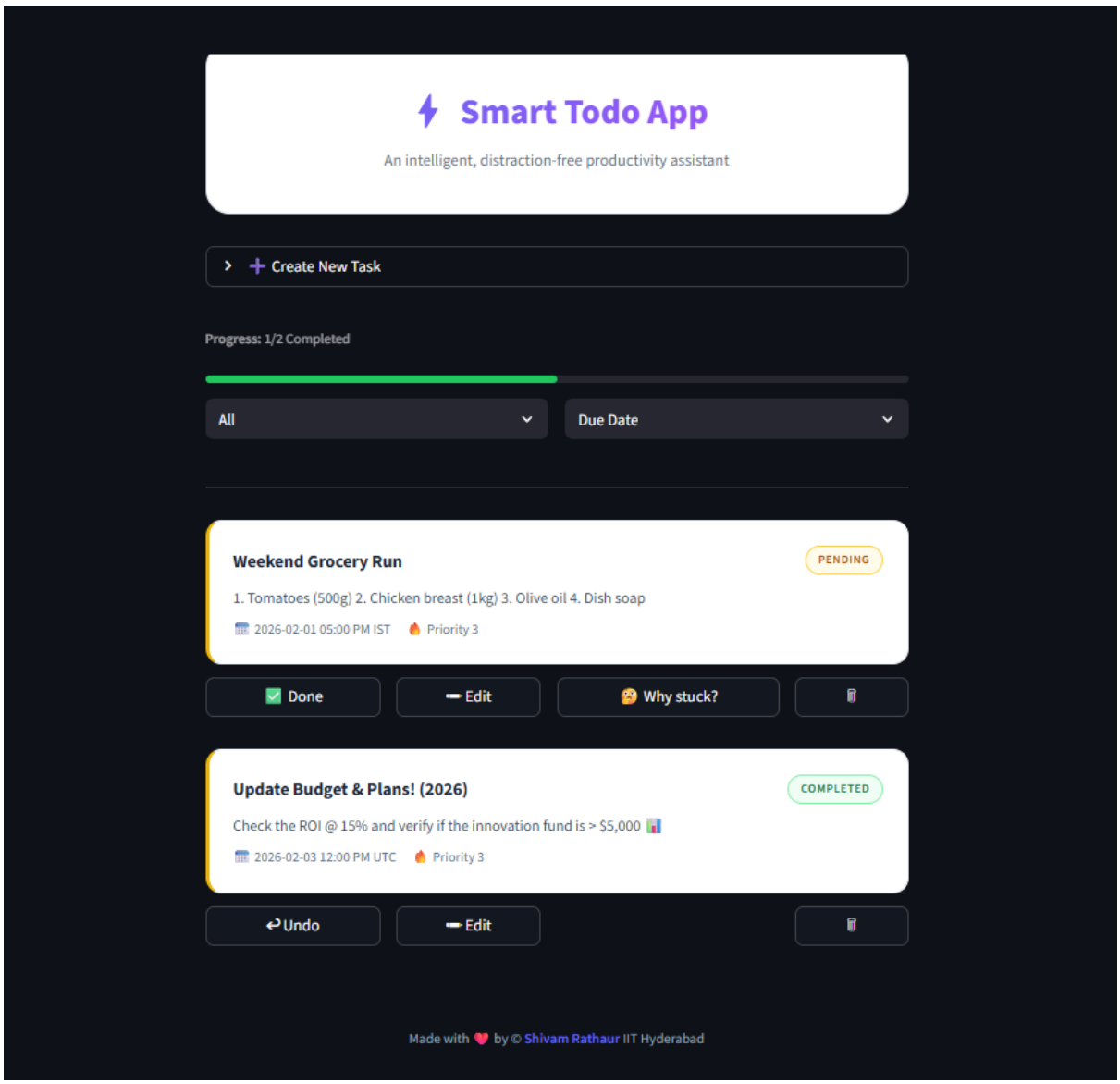
This ensures the app never crashes due to external API issues.

UI / UX Design Considerations

The UI was designed with simplicity and clarity in mind:

- Modern card-based layout for tasks
- Clean typography and consistent spacing
- Hover effects for better visual feedback
- Explicit action buttons to avoid accidental submissions
- Prevention of unintended task creation via keyboard input

The design aims to balance aesthetics with usability, making the app intuitive for first-time users.



Data Persistence & Reliability

All task data is stored locally using SQLite:

- Tasks persist across application restarts
- Status, priority, due date, and edits are reliably saved
- No external database setup is required

This approach ensures both simplicity and reliability.

Security & Best Practices

- No API keys are hardcoded or committed
- Environment variables are used for sensitive credentials
- Graceful handling of API failures
- Clean session state management to avoid unintended behavior

These practices align with industry standards for secure application development.

How to Run the Application

- Install dependencies:
    - `pip install -r requirements.txt`
- Set the environment variable for the AI API key:
    - Windows: `setx GEMINI_API_KEY "api_key"`
    - macOS / Linux: `export GEMINI_API_KEY="api_key"`
- Run the app:
    - `streamlit run app.py`

Conclusion

The Smart Todo App demonstrates practical use of Python, SQLite, Streamlit, and Generative AI to build a robust, user-friendly application. The project emphasizes clean architecture, thoughtful UI design, secure API usage, and meaningful AI integration.

This application reflects both technical proficiency and attention to user experience.

GitHub Repo: 👉 Click Here