Smart Task Manager & Productivity Dashboard

Platform: Desktop Application

Tech Stack: Python 3.x, SQLite3, PyQt5, APScheduler/Schedule, Matplotlib/Plotly

**1. Project Description**

The Smart Task Manager & Productivity Dashboard is a desktop-based productivity application designed to help users manage their tasks effectively. It combines intuitive task organization with intelligent prioritization and productivity tracking features. The goal is to create a standalone, portable, and intelligent task manager.

**2. Key Objectives**

* Provide an easy-to-use task CRUD system
* Offer smart prioritization and recurring task management
* Deliver a visual dashboard of productivity and performance
* Integrate notifications, reminders, and Pomodoro tracking
* Maintain logs and task history for performance evaluation

**3. Core Functional Modules**

1. a. Task Management (CRUD)

Fields: title, description, due date/time, priority, status, category, recurrence, attachment\_path  
Operations: Create, Read, Update, Delete

1. b. Task Sorting & Filtering

Filters by category, priority, date, status

1. c. Productivity Dashboard

Displays: Total tasks, Completed vs pending ratio, Completion percentage, Daily/weekly charts using Matplotlib or Plotly

1. d. Reminders & Notifications

Scheduled alerts for approaching deadlines using schedule or APScheduler

1. e. Search Functionality

Search tasks by keyword/title

1. f. Task History & Logs

Track completed tasks and their timestamps

1. g. Recurring Task Handling

Supports automatic re-creation based on recurrence

1. h. Notes & Attachments

Allows attaching files and adding notes per task

1. i. User Settings & Preferences

Custom themes, default views, notification settings

1. j. Data Backup & Restore

Manual or scheduled backup of the SQLite database

**4. Smart Features**

* Intelligent Task Prioritization: Based on due date proximity and past user behavior
* Daily Summary Emails: (optional extension)
* Pomodoro Timer: Integrated productivity technique
* Progress Streak Tracker: Visual motivator showing consistent task completion days

**5. Database Design (SQLite)**

Tables:

1. 1. Tasks

id INTEGER PRIMARY KEY AUTOINCREMENT,  
title TEXT,  
description TEXT,  
due\_date TEXT,  
priority TEXT,  
status TEXT,  
category TEXT,  
recurrence TEXT,  
attachment\_path TEXT

1. 2. TaskHistory

id INTEGER PRIMARY KEY AUTOINCREMENT,  
task\_id INTEGER,  
completion\_date TEXT

1. 3. UserSettings

id INTEGER PRIMARY KEY AUTOINCREMENT,  
setting\_name TEXT,  
setting\_value TEXT

**6. Why This Project?**

* Demonstrates core Python and GUI skills
* Real-world usability and impact
* Expandable with smart AI features and integrations
* Fully offline and lightweight