Task Manager — Project Report

Student Name: Deeksha Parmar
Registration Number: 25BCY10211
Course: Introduction to Problem Solving & Programming
Project Title: RSVP Task Manager – Task Management System using Python

---

1. Introduction

Time management and task organization are essential for productivity in both academic and personal contexts. Managing multiple tasks with varying urgency, difficulty, and deadlines can be challenging, especially for students and professionals who juggle numerous responsibilities.

This project, **RSVP Task Manager**, is a Python-based task management application developed using Jupyter Notebook. It allows users to **add, view, update, and delete tasks** in a structured way, storing task information in a CSV file for persistence. The project demonstrates how **basic programming concepts** and problem-solving methodologies can be applied to **real-world scenarios** for effective task organization.

---

2. Problem Definition

Many people face difficulty in **tracking their daily tasks**, especially when tasks vary in urgency and complexity. Manual methods, such as notebooks or sticky notes, are error-prone and inefficient. There was a need for a **simple, interactive, and persistent task manager** that can:

1. Store tasks with essential attributes.

2. Allow users to update the task status as they complete tasks.
3. Provide the ability to delete tasks that are no longer relevant.
4. Present all tasks in a readable tabular format.

The goal was to **implement a practical solution** using Python programming and CSV file handling.

---

3. Project Evolution

3.1 Initial Format

At the start of development, the task manager was designed with a **simplified format**, containing only the following attributes:

- **Task Name**
- **Task Urgency**
- **Task Difficulty**

This format allowed users to **quickly add tasks** but lacked essential information such as **category, deadline, and status**. While functional for basic task tracking, it did not provide sufficient context for real-world applications, such as academic assignments or personal goals with specific deadlines.

3.2 Final Format

After analyzing requirements and applying problem-solving principles, the format was **expanded to include more comprehensive details**:

- **Task Name** — Description of the task.
- **Category** — Context of the task (e.g., Academics, Personal, Work).
- **Deadline** — Date by which the task must be completed.
- **Status** — Pending or Done.
- **Task Urgency** — Low, Medium, High.
- **Task Difficulty** — Easy, Medium, Hard.

This evolution allowed the system to:

1. Better organize tasks by **category**.
2. Track **deadlines** for timely completion.
3. Maintain the **urgency and difficulty levels** for prioritization.
4. Provide a more **realistic simulation of task management**, closer to practical applications.

---

4. Objectives

The main objectives of this project are:

- To **apply problem-solving principles** in designing a functional task manager.

- To **demonstrate Python programming skills**, including file handling, data manipulation, and user interaction.
- To develop a **persistent storage mechanism** for tasks using CSV files.
- To integrate task attributes such as **urgency and difficulty**, improving prioritization.
- To create an **interactive menu-driven interface** to enhance usability.

---

5. Tools and Technologies Used

Component	Purpose
Python (via Jupyter Notebook)	Programming language for development
Pandas Library	For reading, writing, and manipulating CSV data
CSV File	Persistent storage of tasks
Jupyter Notebook	Interactive environment for development and demonstration

---

6. Methodology

The project was developed using **structured problem-solving techniques** learned in the course, including:

6.1 Problem Analysis

- Identified the real-world problem of task tracking.
- Defined the attributes for each task, initially simple, then expanded.

6.2 Requirement Analysis

- User should be able to add, view, update, and delete tasks.
- Task information must persist across sessions.
- Task list should be presented in a readable tabular format.

6.3 Top-Down Design / Modularization

The application was divided into modules:

1. **CSV Initialization:** Checks if `tasks.csv` exists; creates it if not.
2. **Add Task Function:** Adds a new task with all attributes and automatically assigns a unique Task ID.
3. **View Task Function:** Reads and displays all tasks from the CSV file.
4. **Update Task Function:** Updates the status of a selected task.
5. **Delete Task Function:** Removes a task from the CSV based on Task ID.
6. **Interactive Menu:** Provides a simple interface to access all functionalities.

6.4 Algorithm Development

- **Step 1:** Initialize the CSV file with headers if not present.
- **Step 2:** To add a task, read the CSV, determine the next Task ID, append the task, and save.
- **Step 3:** To view tasks, read the CSV and display the data frame.
- **Step 4:** To update a task, locate it by Task ID, modify the status, and save.
- **Step 5:** To delete a task, filter out the task by Task ID and save.
- **Step 6:** Use a loop-driven menu to allow repeated operations until the user exits.

---

7. Implementation Details

7.1 CSV Structure (Final Format)

Task_ID	Task_Name	Category	Deadline	Status	Task_Urgency	Task_Difficulty

- **Task_ID:** Number of the entered task unique key factor
- **Task_Name:** Description of the task.
- **Category:** Context of the task.
- **Deadline:** Date by which the task must be completed.
- **Status:** Pending or Done.

- **Task_Urgency:** Low, Medium, High.
- **Task_Difficulty:** Easy, Medium, Hard.

7.2 Python Functions

1. `add_task()` — Adds a new task with all attributes.
2. `view_tasks()` — Displays all tasks.
3. `update_status()` — Updates the status of a task.
4. `delete_task()` — Deletes a task based on Task ID.
5. **Interactive menu** — Combines all functionalities in a loop with options 1–5 (Add, View, Update, Delete, Exit).

---

8. Testing and Results

- **Initial testing:** Verified CSV creation and integrity.
- **Add Task:** Tested multiple tasks with varying urgency, difficulty, and deadlines.
- **View Task:** Confirmed all tasks are displayed correctly.
- **Update Status:** Successfully changed statuses from Pending → Done.
- **Delete Task:** Verified tasks were removed correctly.

- **Interactive Menu:** Functioned smoothly with repeated operations, demonstrating a usable application.

The program **performed as expected**, allowing efficient task tracking with persistent storage.

---

9. Application of Problem-Solving Concepts

This project applied several **core problem-solving concepts** learned in the course:

1. **Breaking down the problem** into smaller modules (CSV handling, adding, viewing, updating, deleting tasks).

2. **Algorithm design** for sequential operations like task addition and update.

3. **Use of conditions and loops** to handle user input and dynamic updates.

4. **Error handling and validation**, e.g., checking Task ID existence before updates/deletion.

5. **Iterative development:** Started with a **simple format** (Name, Urgency, Difficulty) and improved it to a **full-featured format** (Name, Category, Deadline, Status, Urgency, Difficulty) based on real-world requirements.

6. **Connecting theory with real-world application**, demonstrating how computational thinking simplifies task management.

---

10. Conclusion

The **RSVP Task Manager** project demonstrates how Python programming can be applied to **solve real-world problems** in task management. By implementing **modular code, persistent storage, and user-friendly interaction**, this project bridges the gap between **theory and practical application**.

Through this project, the following skills were reinforced:

- Structured problem-solving
- Algorithm design
- Python programming and file handling
- Data management using CSV and Pandas
- Modular code design and user interaction

The project evolved from a **basic task tracker** to a **full-featured task management system**, highlighting how iterative problem-solving and user-focused design can enhance practical utility.

Deeksha Parmar
25BCY10211