Sphere Coding Club – Project 2

TASK TRACKER APPLICATION

(Project Report)

Created By:

Madhura

Isha

Anish

Date: 23/11/2025

ABSTRACT

The Task Tracker Application is a simple and efficient desktop-based project management tool designed to help users organize their projects and tasks in a structured manner. The system allows users to create multiple projects, add tasks under each project, assign due dates, set priorities, and track task completion. It uses a Tkinter-based GUI for an intuitive interface and JSON file storage for permanent data persistence. This application aims to improve task organization, enhance productivity, and offer users a clean and user-friendly method to monitor progress in their daily work. The system is lightweight, platform-independent, and ideal for students, freelancers, and individuals managing multiple tasks.

INTRODUCTION

In today’s fast-paced world, individuals often juggle multiple responsibilities, personal commitments, academic work, and creative pursuits simultaneously. As tasks increase and priorities shift, it becomes increasingly challenging to maintain a clear and organized structure for daily activities. Traditional methods of task tracking—such as notebooks, scattered notes, or basic digital lists—frequently fall short when managing multiple ongoing projects. Users struggle with forgetting important deadlines, losing track of progress, or failing to maintain consistency due to the absence of a structured system.

The **Task Tracker Application** is designed to address these challenges by providing a simple, intuitive, and efficient solution for project-based task management. This application enables users to create multiple projects, break them down into manageable tasks, and track the overall progress in a visually clear manner. Each task can be assigned specific attributes such as a due date, status (completed/not completed), and descriptive information, allowing users to prioritize and plan their workflow effectively.

Developed using Python and the Tkinter GUI framework, the application offers a clean and user-friendly interface suitable for users of all experience levels. Unlike web-based tools that require heavy installations or online connectivity, this application is lightweight and operates entirely offline. Data is stored securely using JSON, ensuring that all user information, projects, and tasks remain accessible even after closing the program.

The purpose of the Task Tracker Application is not only to improve productivity but also to help users maintain clarity, reduce stress, and gain better control over their activities. Whether the user is a student managing assignments, a developer organizing modules, or a creator balancing multiple projects, the Task Tracker Application acts as a reliable companion for structured and clear task management. This documentation provides an in-depth explanation of the system's architecture, features, modules, design flow, and implementation details.

PROBLEM STATEMENT

Managing tasks and projects efficiently has become a significant challenge for individuals in academic, professional, and personal environments. With increasing responsibilities and overlapping deadlines, users often struggle to maintain clarity in their workflow. Traditional methods such as handwritten notes, simple checklists, or scattered digital reminders are insufficient when dealing with multiple projects and complex task structures. These methods do not provide features such as due dates, task status updates, automatic progress tracking, or a centralized view of ongoing work.

Furthermore, the absence of a structured system leads to problems such as forgotten deadlines, improperly prioritized work, disorganized project flow, and reduced productivity. Individuals frequently find themselves overwhelmed due to the lack of a reliable tool that can categorize tasks under different projects and maintain a persistent record of progress.

To address these issues, there is a clear need for a lightweight, intuitive, and accessible task management application that simplifies project tracking, ensures proper task organization, and enhances user productivity without requiring advanced technical knowledge. The Task Tracker Application aims to fulfill this need by providing a clean and user-friendly interface along with robust features for project and task management.

OBJECTIVES

The primary objective of the **Task Tracker Application** is to create a structured, efficient, and user-friendly system that simplifies project and task management. The system aims to address the shortcomings of traditional manual methods by providing a digital platform where users can organize their work with clarity and control.

The detailed objectives of the application are as follows:

1. To provide a clean and intuitive user interface

Create a simple Tkinter-based GUI that allows users to navigate, manage, and view their tasks without confusion, making the application accessible for all users regardless of technical background.

2. To allow users to create and manage multiple projects

Enable users to generate as many projects as needed, each serving as a separate workspace for organizing related tasks.

3. To facilitate the addition and removal of tasks

Provide functionality to add detailed tasks under any project and remove them when no longer needed, ensuring a flexible and organized workflow.

4. To allow tasks to have important attributes

Each task should support essential properties such as:

- **Due Date**
- **Priority**
- **Completion Status**
- **Task Description**

This helps users plan effectively and track deadlines.

5. To track task completion and update project status

Automatically update the project's overall status to "Completed" when all its tasks are finished, and to "Not Completed" when pending tasks remain.

6. To ensure data persistence using JSON

Store all data—including users, projects, and tasks—in a JSON file so that information remains saved even after closing the application.

7. To support efficient project navigation

Allow users to switch between different projects quickly, view their details, and modify tasks without losing workflow continuity.

8. To enhance user productivity and organization

Help users stay consistent, meet deadlines, and gain control over their schedules by offering a structured environment to monitor progress.

EXISTING SYSTEM

In the existing method of task and project management, most users rely on a mixture of manual and unstructured digital tools. Common practices include using notebooks, loose paper notes, basic phone reminders, or simple list-making applications that lack the ability to handle multiple projects effectively. These methods may work for small or occasional tasks, but they quickly become inefficient when users are managing several responsibilities at the same time.

Traditional systems do not provide features such as automatic task tracking, due date reminders, or categorized project structures. As a result, users often face the following limitations:

1. No Centralized Organization

Tasks belonging to different categories—such as academics, personal work, or creative projects—are scattered, making it difficult for users to maintain a clear overview.

2. No Automatic Progress Tracking

Existing tools do not automatically update task status or project completion. Users must manually track progress, which is prone to errors and inconsistencies.

3. Lack of Due Date and Priority Management

Manual systems rarely support due dates or prioritization, causing users to miss deadlines or mismanage time.

4. Data Loss or Inconsistency

Notes can be misplaced, deleted accidentally, or forgotten. There is no reliable system to store and retrieve task information over time.

5. Poor User Experience

Multiple tools used inconsistently lead to confusion, reduced productivity, and difficulty maintaining discipline in task tracking.

Due to these limitations, there is a strong need for a digital system that is structured, reliable, user-friendly, and capable of storing tasks and projects efficiently.

PROPOSED SYSTEM

The **Task Tracker Application** is proposed as an efficient and user-friendly solution to address the shortcomings of traditional task-management methods. This system provides a structured digital environment where users can create, view, and manage multiple projects and tasks with ease. The application is designed to support organized work habits, improve productivity, and reduce the confusion associated with manual task tracking.

The proposed system integrates key functionalities such as task creation, due date assignment, completion tracking, and data persistence through JSON storage. It also offers a Tkinter-based graphical interface that eliminates the complexity of command-line operation, making the tool accessible to both technical and non-technical users.

The major characteristics of the proposed system are as follows:

1. User Authentication System

The application supports **Login and Signup**, enabling multiple users to maintain separate task data securely.

2. Project Management

Users can:

- Create new projects
- View project details
- Update project information
- Delete projects permanently

Each project acts as a container for related tasks.

3. Task Management

Under each project, users can:

- Add new tasks
- Assign due dates
- Provide task descriptions
- Mark tasks as completed or pending
- Remove tasks whenever required

This ensures clarity and flexibility in scheduling.

4. Automatic Project Status Update

Based on the completion of tasks, the system automatically updates the project status:

- **Completed** — when all tasks are done
- **Not Completed** — when one or more tasks remain pending

This provides real-time progress monitoring.

5. JSON-Based Data Persistence

The application uses JSON files to store all:

- User details
- Projects
- Tasks
- Status

This ensures data remains stored permanently and can be retrieved anytime, even after the application is closed.

6. Tkinter Graphical User Interface

The system provides an intuitive GUI with clearly separated screens:

- Login screen
- Home page
- Project view page
- Task management section

This improves accessibility and enhances the user experience.

7. Lightweight and Offline

The system requires no internet connection, server, or heavy database installation. It runs smoothly on any basic computer, making it ideal for students and individuals.

SYSTEM REQUIREMENTS

To ensure smooth installation and execution of the Task Tracker Application, the following hardware and software requirements must be met. The system is lightweight and does not demand high-end resources, making it suitable for most personal computers.

Hardware Requirements

Requirement	Description
Processor	Minimum 1 GHz processor (Intel/AMD)
RAM	Minimum 2 GB RAM (4 GB recommended for smooth multitasking)
Storage	At least 100 MB free disk space
Display	Standard VGA display or higher

9.2 Software Requirements

Software	Description
Operating System	Windows / Linux / macOS
Python Version	Python 3.8 or higher
Libraries Needed	Tkinter (built-in), JSON (built-in)
Text Editor or IDE	VS Code, PyCharm, Sublime Text, or any Python-compatible IDE

9.3 Additional Tools (Optional)

1. **Git** for version control
2. **Virtual Environment** (venv) for dedicated project environment
3. **Python package manager (pip)**

9. 4 Installation Instructions

Before running the Task Tracker Application, the following libraries and tools must be installed.
All installation commands **must be executed inside the Terminal of the PyCharm IDE**.

Required IDE

1. **PyCharm IDE**
   Download and install PyCharm Community Edition from:
   https://www.jetbrains.com/pycharm/

Required Python Libraries

2. Install Tkinter

```
pip install tkinter
```

3. Install CustomTkinter

```
pip install customtkinter
```

4. Install TkCalendar

```
pip install tkcalendar
```

Note:

All installation commands **must be executed in the Terminal of PyCharm IDE** to ensure modules install correctly within the project environment.

SYSTEM ARCHITECTURE

The **System Architecture** of the Task Tracker Application provides a clear overview of how different components of the system interact with each other. It shows the flow of data, the relationship between modules, and the internal structure of the application. The architecture ensures that the application is modular, maintainable, and easy to understand.

This application follows a **modular architecture**, where each major function—such as Authentication, Project Management, Task Management, and Data Storage—is handled separately but connects seamlessly through the application logic.

10.1 Architectural Overview

The system consists of the following major components:

1. User Interface Layer (Tkinter GUI)

This is the visual layer that interacts directly with the user.
It contains screens such as:

- Login/Signup Screen
- Home Dashboard
- Project View Screen
- Task Management Screen

Each screen is designed using Tkinter frames and widgets.

---

2. Application Logic Layer

This layer handles:

- Login validation
- Project creation and deletion
- Task addition, deletion, and updates
- Automatic status updates

It acts as a bridge between the GUI and the data storage.

---

3. Data Storage Layer (JSON File System)

All user data is stored in a **JSON file**.
This includes:

- User information
- Projects
- Tasks
- Completion status

Data is read from and written to the JSON file using Python's built-in json module.

10.2 Data Flow Explanation

1. **User interacts** with the application through the GUI.
2. GUI sends commands to the **Application Logic Layer**.
3. Logic layer processes the request (add, delete, update tasks or projects).
4. The processed information is **saved to JSON** to ensure data persistence.
5. Whenever the app starts, **data is loaded from JSON** and displayed in the GUI.

FEATURES OF THE SYSTEM

The **Task Tracker Application** includes a wide range of features designed to make project and task management simple, organized, and efficient. Each feature aims to improve usability, enhance productivity, and ensure that users have a smooth experience while managing their daily responsibilities.

Below are the detailed features of the system:

---

1. User Login & Signup

The application supports user authentication through a simple login/signup system.

- New users can create accounts.
- Existing users can log in using their credentials.
- All user data is stored securely in a JSON file.
  This ensures personalized access and prevents data mixing between different users.

---

2. Create Project

Users can create new projects by entering a project title.
Each project acts as a separate workspace under which multiple tasks can be added.
This helps users divide their responsibilities into well-organized categories.

---

3. View Project

Users can select any project from the list and view its associated tasks.
This includes:

- Task descriptions
- Task statuses
- Due dates
- Total number of tasks

This feature gives users a clear overview of their workload.

---

4. Update Project

Projects can be modified by:

- Adding new tasks
- Removing tasks
- Marking tasks as completed
- Editing descriptions
- Updating due dates

This ensures flexibility and helps users adapt their tasks as plans change.

---

5. Delete Project

Users can permanently delete any project.
Once deleted, all tasks under that project are removed from the system.
This feature helps in removing completed or unnecessary projects.

---

6. Add Task

Under each project, users can add tasks with the following details:

- Task description
- Due date
- Optional priority level

Adding tasks allows users to break down their work into smaller, manageable steps.

---

7. Due Date Assignment

Each task can have a due date, helping users stay aware of deadlines and improve time management.
Tasks without due dates may lose priority, so this feature ensures effective planning.

---

8. Task Status (Completed / Not Completed)

Users can:

- Mark tasks as completed
- Toggle them back to incomplete
- Identify completed tasks through visual indicators

This gives users real-time progress visibility.

---

9. Remove Task

Unwanted or completed tasks can be removed at any time.
This keeps the workspace clean and prevents unnecessary clutter.

---

10. Automatic Project Completion Status

The system automatically updates the project's overall status based on task completion:

- When **all tasks** are completed → Project becomes **Completed**

- If **any task** is pending → Project remains **Not Completed**

This automation saves time and provides instant feedback on project progress.

---

11. JSON-Based Data Storage

All data (users, projects, tasks, status) is saved in a JSON file.
Benefits:

- Permanent storage
- No data loss when the application closes
- No heavy database installation needed
- Simple and fast data retrieval

---

12. Tkinter Graphical User Interface

The entire system is built using Python's Tkinter library.
The GUI includes:

- Buttons
- Labels
- Listboxes
- Input fields
- Separate frames for login, home, and project view

This makes the application visually clean and easy to use for all age groups.

---

13. Lightweight & Offline

The application does not require:

- Internet connection
- Server hosting
- Additional installations

It runs locally on any basic computer, making it portable and efficient.

MODULE EXPLANATION

The Task Tracker Application is divided into several interconnected modules, each responsible for a specific functionality. This modular approach makes the application easier to understand, maintain, and extend in the future. Below is a detailed explanation of each module in the system:

---

1. Authentication Module (Login & Signup)

Purpose:

To verify user identity and provide secure access to the application.

Functions:

- **Signup:** Allows new users to register by creating an account.
- **Login:** Validates existing user credentials.
- **Error Handling:** Displays error messages for incorrect input.

Process:

User data is stored in JSON format. During login, the system loads user information from the JSON file and checks if the credentials match.

---

2. Home Module (Project Dashboard)

Purpose:

To display all projects created by the user in one place.

Functions:

- Show list of existing projects
- Navigation to project details
- Provide buttons for:
    - **Create Project**
    - **Delete Project**
    - **Logout**

Process:

When the user logs in, the system loads all projects related to that user from the JSON file and displays them on the dashboard.

---

3. Project Management Module

Purpose:

To allow users to create and manage multiple projects efficiently.

Functions:

- **Add Project**
- **View Project Tasks**
- **Rename/Edit Project (if needed)**
- **Delete Project**

Process:

Each project is stored as a separate object in JSON with its own list of tasks. The module ensures smooth transitions between different project operations.

---

4. Task Management Module

Purpose:

To manage all tasks belonging to a specific project.

Functions:

- Add a new task
- Remove an existing task
- Mark a task as completed or pending
- Add due dates
- View all tasks
- Update an existing task

Process:

When a project is selected, all tasks are loaded from JSON and displayed. The user can then modify task details, and the updated information is saved back to the JSON file.

5. Task Status Module

Purpose:

To track the completion status of each task and update project progress accordingly.

Functions:

- Mark task as completed
- Mark task as not completed
- Automatically update project's overall status

Process:

If all tasks under a project are marked completed → Project status becomes **Completed**.
If even one task is pending → Status stays **Not Completed**.

6. JSON Storage Module

Purpose:

To provide permanent data storage and retrieval using JSON files.

Functions:

- Save updated projects and tasks
- Load existing user data
- Insert new entries
- Maintain data consistency

Process:

Every operation — login, add project, delete task, mark status — reflects immediately in the JSON file.

This keeps the application lightweight and avoids using heavy databases.

7. GUI Module (Tkinter)

Purpose:

To provide a user-friendly interface for interacting with the application.

Screens Included:

- Login Screen
- Signup Screen
- Home Screen
- Project Management Screen
- Task Management Screen

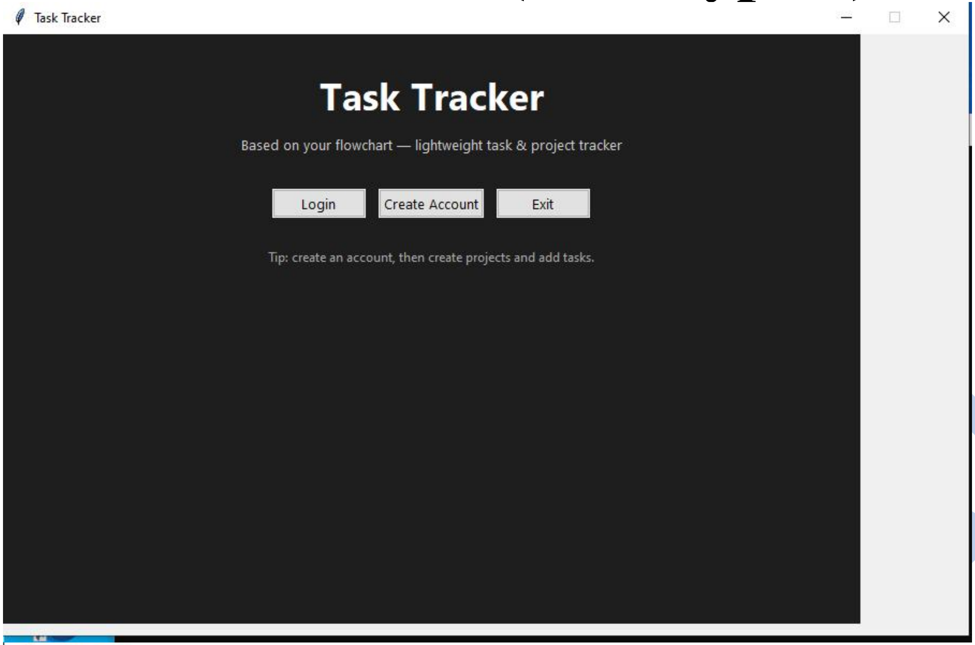
Widgets Used:

- Buttons
- Labels
- Entry fields
- Frames
- Listboxes

Process:

Tkinter manages transitions by switching between different frames, giving a smooth navigation experience.

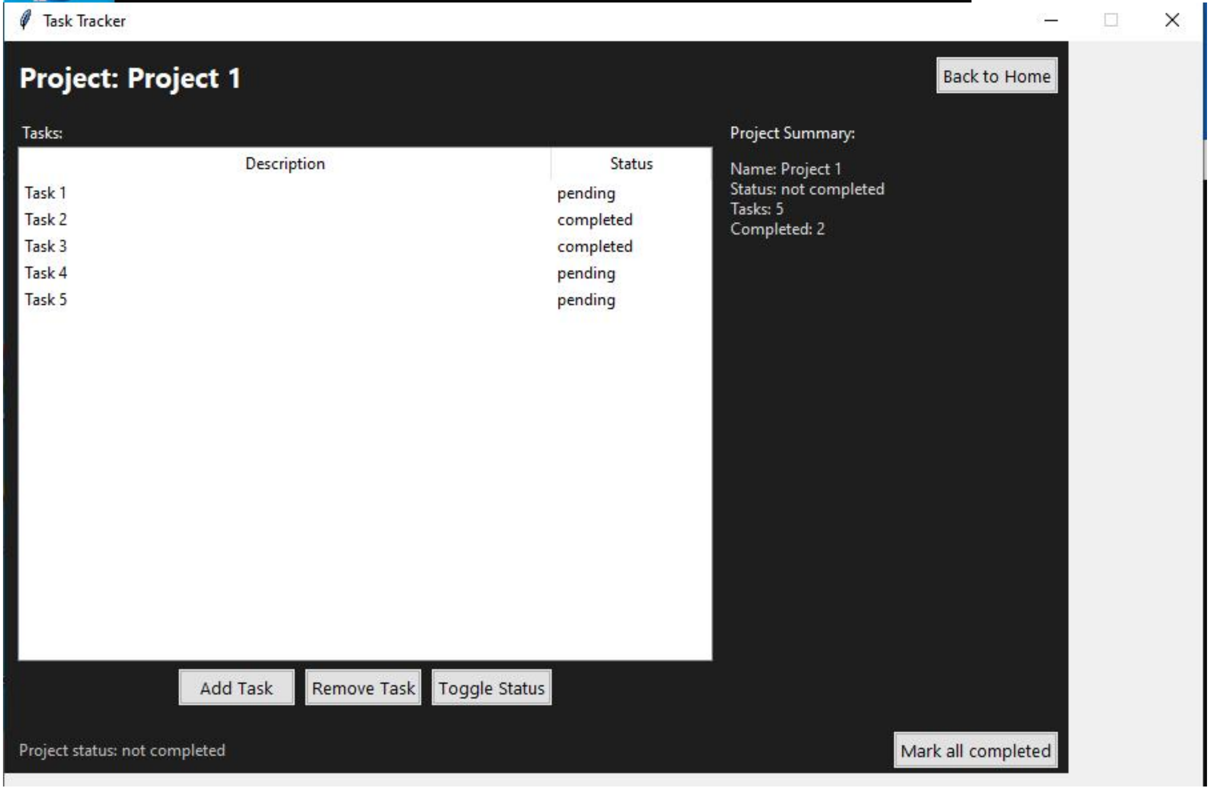
SCREENSHOTS(Prototype 1)



Task Tracker

Based on your flowchart — lightweight task & project tracker

Tip: create an account, then create projects and add tasks.



Project: Project 1

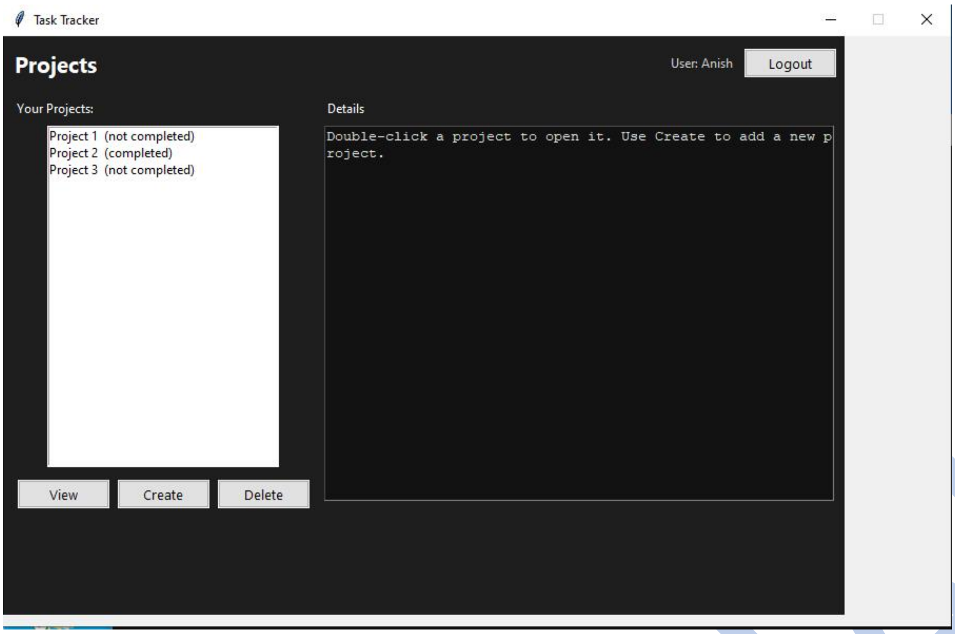
Tasks:

Description	Status
Task 1	pending
Task 2	completed
Task 3	completed
Task 4	pending
Task 5	pending

Project Summary:

Name: Project 1
Status: not completed
Tasks: 5
Completed: 2

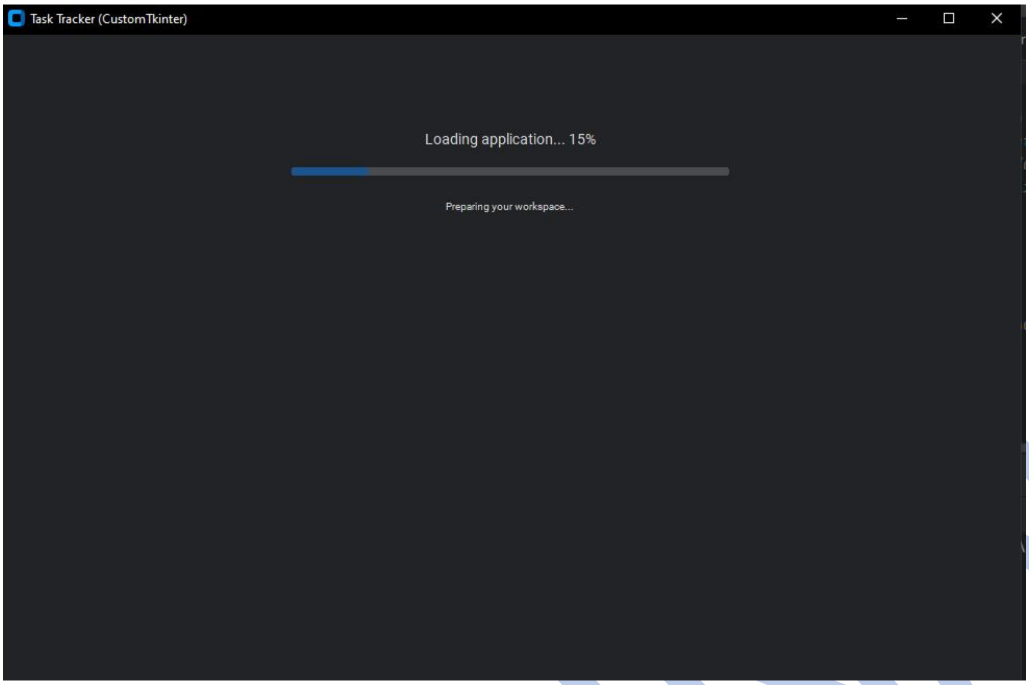
Project status: not completed



Project 1 (not completed)
Project 2 (completed)
Project 3 (not completed)

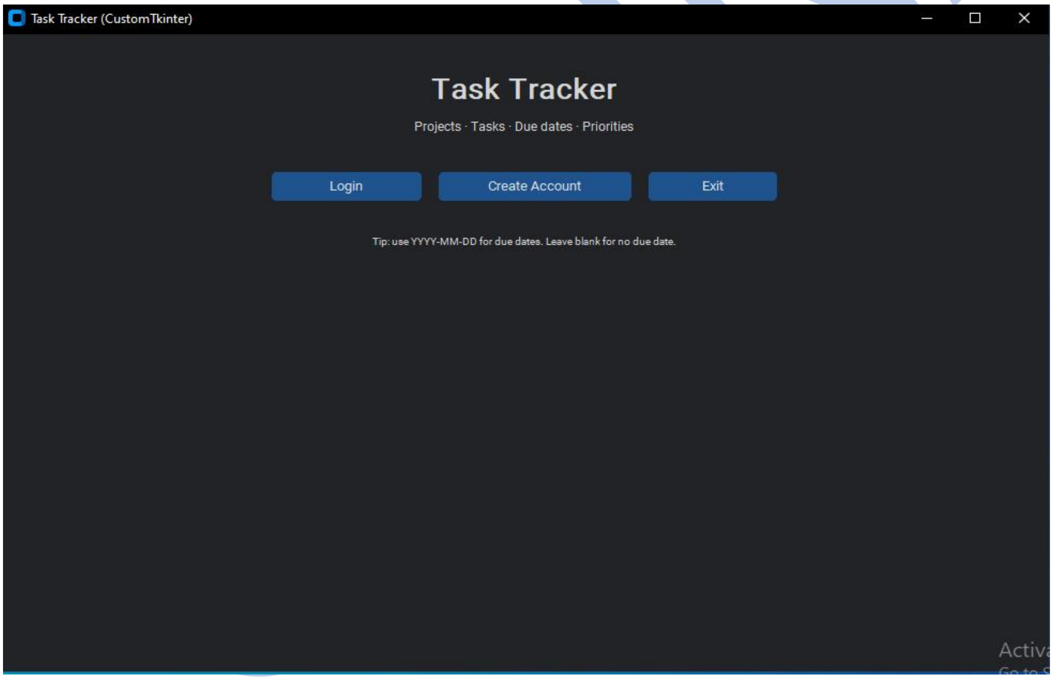
```
Double-click a project to open it. Use Create to add a new p
roject.
```

SCREENSHOTS(FINAL)



Loading application... 15%

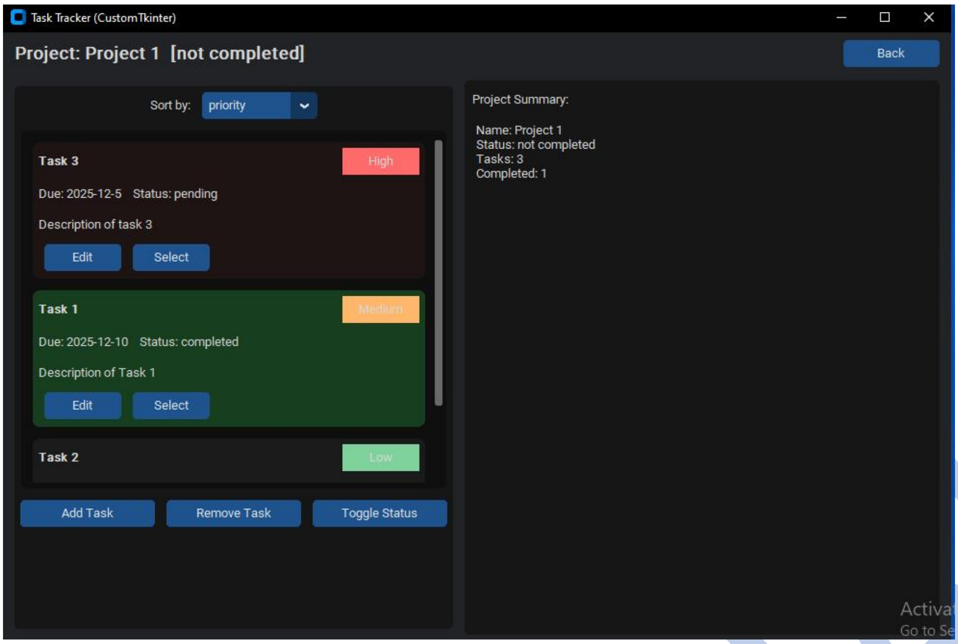
Preparing your workspace...



Task Tracker

Projects · Tasks · Due dates · Priorities

Tip: use YYYY-MM-DD for due dates. Leave blank for no due date.



Project: Project 1 [not completed]

Sort by: priority

Task 3 High

Due: 2025-12-5 Status: pending

Description of task 3

Task 1 Medium

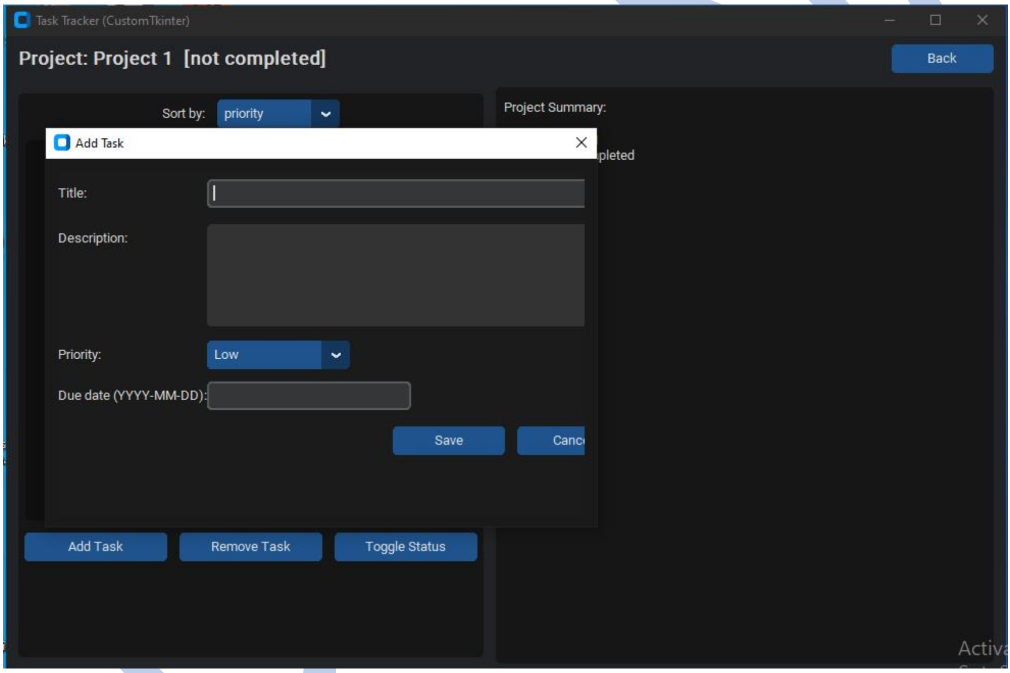
Due: 2025-12-10 Status: completed

Description of Task 1

Task 2 Low

Project Summary:

Name: Project 1
Status: not completed
Tasks: 3
Completed: 1



Project: Project 1 [not completed]

Sort by: priority

Project Summary:

Title:

Description:

Priority: Low

Due date (YYYY-MM-DD):

TESTING

Testing is a crucial phase of software development that ensures the application performs as expected and delivers accurate results. The Task Tracker Application was tested using functional testing methods. Each module was validated with different inputs to verify stability, usability, and data consistency.

Below are the major test cases conducted:

14.1 Test Case Table

Test Case ID	Feature Tested	Input	Expected Output	Actual Result	Status
TC-01	Signup	New username, new password	Account created successfully	Works as expected	Pass
TC-02	Login	Correct username and password	User logged in and Home screen displayed	Works as expected	Pass
TC-03	Login	Wrong password	Error message displayed	Works as expected	Pass
TC-04	Create Project	Project title: "College Work"	Project added to the list	Works as expected	Pass
TC-05	Delete Project	Select project → Delete	Project removed from list and JSON	Works as expected	Pass
TC-06	Add Task	Description + Due date	Task added under selected project	Works as expected	Pass
TC-07	Remove Task	Select task → Delete	Task removed from list	Works as expected	Pass
TC-08	Mark Task Completed	Click "Mark Completed"	Status changes to Completed	Works as expected	Pass
TC-09	JSON Storage	Exit app and reopen	Data persists and loads correctly	Works as expected	Pass

Test Case ID	Feature Tested	Input	Expected Output	Actual Result	Status
TC-10	Automatic Project Status	All tasks completed	Project marked as Completed	Works as expected	Pass

14.2 Observations

- All core functionalities work correctly.
- Data remains consistent after adding, editing, and deleting tasks.
- JSON file updates instantly with every change.
- GUI transitions between screens smoothly.
- No bugs or crashes observed during normal use.

ADVANTAGES

The Task Tracker Application offers several advantages that make it an efficient and practical tool for managing daily tasks and projects. Its simple design, lightweight structure, and reliable functionality make it accessible to a wide range of users.

1. Simple and User-Friendly Interface

The Tkinter-based GUI provides a clean and intuitive layout that makes navigation effortless. Users of all technical backgrounds can use the system comfortably.

2. Efficient Task Organization

The application allows users to create multiple projects and categorize tasks under them. This structured approach improves clarity and ensures better time management.

3. Lightweight and Fast

Since the system uses Python and JSON, it requires no heavy installations or databases. It works smoothly on almost any standard computer with minimal resources.

---

4. Offline Functionality

The application does not require internet access, making it reliable and accessible anytime and anywhere.

---

5. Data Persistence Using JSON

All data—users, projects, and tasks—is stored permanently in JSON files. Even after closing the program, information is preserved, providing consistency and reliability.

---

6. Easy to Customize and Maintain

The modular architecture allows developers to add new features or modify existing ones without affecting the entire system. This makes future enhancements straightforward.

---

7. Real-Time Task Status Tracking

Users can instantly mark tasks as completed or pending, giving them a clear view of progress at all times. The automatic project status update increases convenience.

---

8. Suitable for a Wide Range of Users

Students, freelancers, office employees, and creators can all use this tool to stay organized, plan schedules, and track work progress efficiently.

LIMITATIONS

While the Task Tracker Application provides several useful features for organizing tasks and managing projects, it also has a few limitations due to its simple, lightweight design. These limitations highlight areas where the system can be improved in the future.

---

1. Local Storage Only

The application uses JSON files stored on the local device.
This means:

- Data cannot be accessed from multiple devices
- Users cannot sync tasks across systems
- No cloud backup is available

---

2. No Multi-User Collaboration

Although users can create individual accounts, the system does not support team-based project management where multiple users can work on the same project simultaneously.

---

3. Limited Notification System

The current version does not provide:

- Alerts
- Reminders
- Notifications for upcoming deadlines

Users must manually check due dates.

---

4. Basic User Interface

The Tkinter GUI, while simple and effective, is limited in:

- Modern design
- Custom animations

- Advanced UI interactions

Compared to modern frameworks, it lacks aesthetic flexibility.

---

5. No Priority Sorting (Optional Feature)

Tasks are not automatically sorted by:

- Priority
- Due date
- Status

All sorting must be done manually by the user.

---

6. Not Suitable for Large-Scale Projects

Because the system uses JSON and Tkinter, it is not designed for:

- Enterprise-level data handling
- Large databases
- Complex workflows

It is ideal for personal or small project management only.

FUTURE ENHANCEMENTS

To improve the efficiency, usability, and overall functionality of the Task Tracker Application, several enhancements can be integrated in future versions. These upgrades would make the system more powerful, modern, and suitable for a wider range of users and use-cases.

---

1. Integration of Database Systems (SQLite/MySQL)

Replacing JSON with a real database would allow:

- Faster data retrieval
- Handling larger amounts of data
- Better multi-user support

- Improved scalability
  This upgrade would make the system suitable for team projects and professional environments.

2. Cloud Sync and Online Access

Adding cloud-based storage would enable:

- Access from multiple devices
- Real-time data sync
- Backup and recovery of tasks

This is especially useful for users who work across laptops, PCs, or mobile devices.

3. Notification and Reminder System

The app can be enhanced with:

- Pop-up reminders
- Due date countdown
- Alerts for overdue tasks
- Daily or weekly reminders

This would significantly improve productivity and deadline management.

4. Advanced Sorting and Filtering Options

Users could sort tasks based on:

- Priority
- Due date
- Status
- Recently added

This makes task management faster and more organized.

5. Improved Modern User Interface

Switching to frameworks like:

- PyQt
- Kivy
- CustomTkinter

would provide:

- A cleaner design
- Smoother animations
- Dark mode
- Better aesthetics

This would make the app feel more modern and professional.

---

6. Calendar View for Task Planning

A calendar view could show:

- Upcoming tasks
- Deadlines
- Monthly task overview

This helps users plan long-term goals easily.

---

7. Export and Import Data Features

Users could export their projects in formats like:

- PDF
- CSV
- Excel

This is useful for students presenting work or professionals managing reports.

---

8. Mobile Application Version

A mobile version would allow users to:

- Add tasks on the go
- Receive notifications
- View deadlines anytime

This could be made using Kivy or Flutter in future development.

---

9. Theme Customization

Users could choose:

- Dark mode
- Light mode
- Custom color themes
- Font adjustments

This improves personal comfort and accessibility.

DEVELOPMENT TIMELINE

Date	**Activity / Progress**	**Team Members Involved**
14th November	• Task distribution for faster project completion	Anish, Madhura, Isha (Distribution)
	Rough flowchart discussion with Prajwal	Madhura
15th November	• Modified and refined the flowchart	Anish, Isha, Madhura
16th	• Created basic structure of the	Anish(structure)

Date	Activity / Progress	Team Members Involved
November	application, discussed required changes	
	Performed initial testing of code and app	Isha (Discussion)
	discussed required changes	Madhura (Testing)
17th November	• Modified and improved the GUI design	Anish, Isha, Madhura
19th November	• No work due to practical submissions	Entire Group
20th November	• Modified and updated the application	Anish, Madhura, Isha
21st November	• Finalization of the features in the application	Anish, Madhura.

TEAM CONTRIBUTION SUMMARY

The following section summarizes the contribution of each team member involved in the development of the **Task Tracker Application**. Every member participated in different phases of the project, ensuring successful completion within the given timeline.

1. Anish (Developer – Backend & GUI Logic)

- Designed and developed the basic structure of the application.
- Implemented the core logic for project and task management.
- Worked on integrating JSON storage for data persistence.
- Modified and improved the Tkinter/CustomTkinter GUI.
- Assisted in debugging and refining overall functionality.

2. Madhura (Testing & Module Verification)

- Conducted testing of application modules throughout development.
- Identified bugs, tested GUI flow, and verified task creation features.
- Provided feedback for interface improvements.
- Collaborated during flowchart modification and feature finalization.
- Ensured smooth functioning of the app by validating task workflows.

3. Isha (Documentation, Flow Planning & Coordination)

- Prepared and structured project documentation and reports.
- Modified the flowchart and planned system flow with the team.
- Discussed feature requirements and contributed to GUI adjustments.
- Created the Development Timeline for daily progress tracking.
- Helped finalize the list of features included in the application.

Summary

All members contributed to different aspects of the project, including design, coding, testing, planning, and documentation.
The diverse skill sets and collaborative efforts resulted in a functional and user-friendly Task Tracker Application.

CONCLUSION

The **Task Tracker Application** successfully fulfills its purpose of providing a structured, efficient, and user-friendly system for managing projects and tasks. By integrating essential features such as project creation, task management, due dates, completion tracking, and permanent JSON-based storage, the application offers a reliable solution for individuals seeking better organization in their daily workflow.

The use of Python and Tkinter ensures simplicity, accessibility, and smooth operation across different systems, making the application suitable for students, freelancers, and individuals managing personal or academic work. The modular architecture enhances maintainability and ensures that each component of the system functions independently yet cohesively.

Throughout development and testing, the application demonstrated stability, accuracy, and consistent performance. Although it has certain limitations due to its lightweight design, the proposed future enhancements—such as cloud integration, modern UI upgrades, and advanced sorting features—provide significant opportunities for expansion and real-world usability.

In conclusion, the Task Tracker Application provides a solid foundation for task organization and productivity improvement. It stands as a practical and effective tool that can be enhanced further to meet advanced user needs, making it a valuable addition to personal project management tools.

REFERENCES

Python & Standard Libraries

1. Python Software Foundation. *Python 3 Documentation*. https://docs.python.org/3/

2. Python Software Foundation. *json — JSON encoder and decoder*. https://docs.python.org/3/library/json.html

Tkinter & GUI Development

3. Tkinter Official Reference (Python Docs). https://docs.python.org/3/library/tkinter.html

4. TkDocs – Tkinter Tutorials and Reference Guide. https://tkdocs.com/tutorial/

CustomTkinter (Used for This Project)

5. Tom Schimansky. *CustomTkinter – Official Documentation*. https://customtkinter.tomschimansky.com

6. Tom Schimansky. *CustomTkinter GitHub Repository*. https://github.com/TomSchimansky/CustomTkinter

7. PyPI – CustomTkinter Module Information. https://pypi.org/project/customtkinter/

8. Dev.to. *CustomTkinter – A Complete Tutorial*. https://dev.to/devasservice/customtkinter-a-complete-tutorial-4527

9. GeeksforGeeks. *Build a Basic Form GUI using CustomTkinter*. https://www.geeksforgeeks.org/python/build-a-basic-form-gui-using-customtkinter-module-in-python/

Flowchart, System Design & DFD Resources

10. GeeksforGeeks. *Flowchart – Definition, Symbols, Rules*. https://www.geeksforgeeks.org/flowchart-design-flowchart-symbols-rules/

11. Visual Paradigm. *What is a Data Flow Diagram (DFD)?* https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/

Software Engineering Concepts

12. Tutorialspoint. *Software Engineering Concepts*. https://www.tutorialspoint.com/software_engineering/index.htm