Task Management System Using Python and Tkinter

Course: Data Structures || EE432

Instructor: Dr. Nouri Bin Baraka

Date: 20/02/2025

Team Members:

- Riham Al-Hakim - 22180495
- Iman Mohamed Saeed Kazouz - 2180205725
- Aya Abdel-Aty Al-Naas - 2190203789

1. Project Introduction

This project aims to develop a Task Management System using Python and Tkinter

to create an interactive graphical user interface (GUI) that helps users organize and manage their tasks efficiently.

The application provides the following features:

- Add new tasks with details such as title, description, and priority.
- Edit tasks and reorder their priorities as needed.
- Mark completed tasks and filter them accordingly.
- Search for tasks to quickly locate specific ones.

The project was developed collaboratively, ensuring a well-structured and efficient system.

2. Project Objectives

- Enhance task organization to help users manage their time effectively.
- Design a simple and user-friendly interface using Tkinter.
- Provide persistent storage for tasks using JSON or SQLite.
- Ensure smooth and fast performance even when handling a large number of tasks.

3. Tools & Technologies Used

- Python - for programming the application's functionality.
- Tkinter - for developing the graphical user interface.
- SQLite or JSON - for storing data locally.
- Datetime Library - for handling dates and deadlines.

4. Task Distribution Within the Team

Since teamwork is crucial, we implemented the project together, ensuring that

each member contributed to different aspects of the development process.

We also switched roles when necessary,

which helped in smoothly progressing and improving the project continuously.

5. Program Description

Main Features:

1. Task Addition: Users can add new tasks with details such as title, description, and due date.

2. Task Modification: Existing tasks can be edited easily.

3. Task Completion Marking: Users can mark tasks as completed.

4. Search and Filtering: Users can search for tasks and filter them by priority or status.

5. Data Persistence: Tasks are stored in a database to ensure they remain saved when the application is closed and reopened.

6. Challenges & Solutions

During the development of the project, we encountered several technical challenges,

which we resolved to improve the system. Here are some key issues we faced and how we solved them:

1. Empty Description Field Issue: Ensured all required fields are filled before adding a task.

2. Issues with Task Deletion & Editing: Added validation messages and error handling to prevent operations on unselected tasks.

3. Table Handling Errors: Implemented verification checks before executing

operations.

4. Table Not Updating Correctly: Created an automatic update function to refresh the table after modifications.

5. Slow Performance with Large Task Lists: Optimized code to update only affected data instead of reloading everything.

6. Inefficient GUI Updates: Modified code to refresh only necessary parts of the interface, improving speed.

7. Duplicate Task Entries: Implemented a duplicate-checking function to alert the user before adding a duplicate task.

7. Expected Outcomes

After solving these issues and optimizing the code, we expect the following outcomes:

- A fast and efficient task management application.
- A user-friendly GUI that allows easy task creation, modification, and deletion.
- Optimized performance, ensuring smooth execution even with large task lists.
- Data persistence, allowing users to save and retrieve tasks upon reopening the application.

8. Conclusion

This project was an excellent learning experience in teamwork and problem-solving. Throughout the development process, we encountered and overcame multiple technical challenges, leading to significant improvements in our code. The final result is an effective task management system that can be used for personal productivity or small team collaborations.