A

Project Report

on

Build a To-Do List App

Master of Science in (Computer Science)

दक्षिण बिहार केंद्रीय विश्वविद्यालय

Central University of South Bihar

Under The Supervision of
Dr. Nemi Chandra Rathore
Assistant Professor
Department of Computer Science

Submitted By:

Asmita Raj (CUSB2202312004)

Medha Madhvi (CUSB2202312016)

Pratyush Pritam (CUSB2202312021)

Department of Computer Science

School Of Mathematics, Statistics & Computer Science
Central University of South Bihar, Gaya

December, 2023

दक्षिण बिहार केंद्रीय विश्वविद्यालय
Central University of South Bihar

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **"Build a To-Do List App"** in partial fulfillment of the requirements for the completion of the Mini-Project of the **Master of Science in Computer Science** submitted in the Department of Computer Science of Central University of South Bihar, Gaya, is an original work carried out during the period of November, 2023, under the supervision of **Dr. Nemi Chandra Rathore (Assistant Professor),** Department of Computer Science, Central University of South Bihar, Gaya.

The matter presented in the project has not been submitted by us for the award of any other project of this or any other places.

Asmita Raj (CUSB2202312004)

Medha Madhvi (CUSB2202312016)

Pratyush Pritam (CUSB2202312021)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Nemi Chandra Rathore

(Assistant Professor)

ACKNOWLEDGEMENT

We take this occasion to thank God, almighty for blessing us with his grace and takingour endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide,

Dr. **Nemi Chandra Rathore** for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We extend our sincere thanks to our respected guide **Mr. Purnendu Prabhat,** for Guidance and support to do the project work.

We would like to thank the other faculty members also, at this occasion. Last but not the least; we would like to thank to our team members & friends for the support and encouragement they have given us during the Project Report of our work.

Thank You,
Sincerely,

Asmita Raj (M.Sc. CS)

Medha Madhvi (M.Sc. CS)

Pratyush Pritam (M.Sc. CS)

ABSTRACT

Task management is a ubiquitous challenge faced by individuals, often complicated by the absence of accessible and user-friendly solutions. Traditional methods for organizing tasks are often cumbersome and lack essential features, leading to inefficiencies and disorganization in personal and professional settings. The absence of a streamlined task management tool creates a need for a solution that is both simple and effective. The To-Do List App in Python, developed with Tkinter, addresses the existing problem by offering a straightforward and intuitive task management platform. The application allows users to effortlessly add, view, edit, and delete tasks through a user-friendly graphical interface. The proposed solution aims to enhance the user experience by providing a tool that is not only functional but also adaptable for future feature enhancements. The To-Do List App leverages Python as the primary programming language and Tkinter as the GUI toolkit. These technologies are chosen for their simplicity, readability, and the ability to create a visually appealing interface. The application development process also involves an integrated development environment (IDE), version control with Git, and potential documentation tools for clarity and collaboration. The To-Do List App successfully achieves its objectives, offering users a simple yet effective platform for managing tasks. Through the implementation of core functionalities, including error handling and scalability, the application provides a seamless experience. User feedback and potential future testing will contribute to refining the app, ensuring its usability and relevance. In conclusion, the To-Do List App presents a practical solution to the existing problem of inefficient task management. Its user-friendly design and adaptability lay the groundwork for future enhancements, such as task prioritization and due dates. The conclusion reflects the application's current utility and highlights its potential for growth and improvement based on evolving user needs and technological advancements.

Keyword: *To-Do List App, User-friendly, Task management, Reminders*

Table of Contents

Title **Page No.**

Candidates Declaration……..**I**

Acknowledgement ……..**II**

Abstract.....**III**

Contents ……..**IV**

List of Figures……..**V**

1. Introduction..... .**1**

1.1 Formulation of Problem…….

1.2 Tool and Technology Used……

2. Features & Funtionality……..**3**

3. Working of Project……..**5**

4. Result & Output……..**7**

5. Conclusions……..**8**

References: ……..**16**

List of Figures

S.No.	Title	Page No.
1.	App Interface	7
2.	Database Snapshot	7

Chapter 1
Introduction

1. Introduction

The To-Do List App is a Python-based application developed using the Tkinter library. This report provides an overview of the application's features, design, and functionality. The app allows users to manage tasks through a user-friendly graphical interface, incorporating essential task management operations such as adding, viewing, editing, and deleting tasks. The purpose of this project was to design and implement a To-Do List App, a tool that helps users organize and manage their tasks efficiently. The app allows users to create, edit, and delete tasks. The project aimed to provide a simple and user-friendly interface for task management. The objectives of the project are clearly outlined, indicating the specific goals the team aimed to achieve. These objectives serve as a roadmap for the reader, detailing the intended outcomes of the To-Do List App. The introduction highlights the key features and functionalities of the To-Do List App. This section gives the reader a glimpse into the capabilities of the application. The overarching purpose of the report is communicated, emphasizing that it serves as a documentation of the project's development process, challenges faced, and the resulting To-Do List App. The report is positioned as a valuable resource for understanding the project's evolution.

1.1 Formulation of Problem

Task management is a fundamental aspect of personal and professional productivity. In the absence of efficient tools, individuals often struggle to organize and keep track of their tasks. The To-Do List App aims to address this challenge by providing a user-friendly platform for managing tasks, offering features such as task addition, viewing, editing, and deletion.

The problem at hand is the lack of a streamlined and accessible solution for individuals to manage their tasks effectively. Existing methods may be cumbersome or lack the necessary features for efficient task organization. The To-Do List App seeks to fill this gap by providing a Python-based application that simplifies task management and enhances user productivity.

1.2 Tools and Technology Used

The following technologies and tools were used in the development of the To-Do List App:

I. **Python:**

Python serves as the primary programming language for developing the application. Its simplicity, readability, and extensive standard libraries make it an ideal choice for rapid application development.

II. **Tkinter:**

Tkinter is the standard GUI (Graphical User Interface) toolkit that comes with Python. It provides the necessary components for creating the application's graphical interface, including frames, labels, buttons, entry fields, and list boxes.

III. **Integrated Development Environment (IDE):**

An integrated development environment, such as PyCharm, Visual Studio Code, or IDLE, is used for coding, debugging, and testing the Python scripts that constitute the To-Do List App.

IV. **MySQL Connector:**

MySQL Connector is a Python driver for MySQL databases. It enables Python programs to communicate with MySQL servers.

The script uses MySQL Connector to connect to a MySQL database. It performs operations like inserting, updating, and retrieving tasks from the MySQL database.

V. **MySQL Database:**

MySQL is a popular open-source relational database management system (RDBMS).

The script connects to a MySQL database named "To_Do_List" and interacts with a table named "Task." It performs operations like inserting, updating, and retrieving tasks from this database.

VI. **GitHub (Optional):**

GitHub or a similar platform can be utilized for hosting the project's source code, enabling collaborative development and version control. It also facilitates easier sharing and distribution of the application.

Chapter 2

2. Features & Functionality

Graphical User Interface (GUI):

The application provides a graphical interface using Tkinter, making it user-friendly.

Users interact with the application through buttons, an entry widget, and a listbox to manage their to-do list.

Add Task:

Users can seamlessly add tasks by inputting details into the dedicated entry field and clicking the "Add Task" button.

An error message is displayed if a task is not entered, enhancing user guidance.

View Task:

The "View Task" feature enables users to retrieve task details by selecting a task from the list.

Error handling is implemented to notify users when no task is selected.

Edit Task:

Users can efficiently edit tasks by selecting a task, triggering a pop-up dialog for inputting edited task details.

Error messages are displayed when no task is selected or when the edit operation is canceled.

Delete Task:

The "Delete Task" functionality allows users to remove a selected task, with a confirmation dialog ensuring intentional deletions.

Error messages are employed for cases where no task is selected.

Delete All Tasks:

Users can choose to delete all tasks, with a confirmation dialog preventing accidental deletions.

The system communicates effectively when attempting to delete all tasks with none present.

Retrieve from Database:

Tasks are retrieved from the database on application startup.

The application retrieves tasks stored in the MySQL database and populates the in-memory list and the listbox with these tasks when the program starts.

Database Integration:

The application uses a MySQL database to store tasks.

Tasks are inserted, updated, and retrieved from the MySQL database using SQL queries and the MySQL Connector library.

Error Handling:

The application handles errors gracefully.

The script uses try-except blocks to catch exceptions, such as TclError, providing informative error messages to the user.

Exit:

The "Exit" button gracefully closes the application.

Icon:

The application window has a custom icon.

An icon image ("icon.png") is loaded and set for the application window.

Chapter 3

3. Working of Project

3.1 Initialization:

The script starts by importing the required libraries: tkinter for GUI, simpledialog and messagebox for dialog boxes, and mysql.connector for MySQL database connectivity.

It establishes a connection to the MySQL database named "To_Do_List."

3.2 Graphical User Interface (GUI) Setup:

The Tkinter library is utilized to create the graphical user interface.

Three frames (header_frame, list_frame, and button_frame) are created to organize the layout of the application.

Widgets such as labels, entry, listbox, and buttons are placed in these frames to create an intuitive user interface.

3.3 Database Retrieval on Startup:

The retrieve_database() function is called to fetch tasks from the MySQL database during the startup of the application.

The tasks are stored in the task list.

3.4 Button Functions:

The script defines various functions to handle button clicks, including:

add_item(): Adds a task to the in-memory list (task) and inserts it into the MySQL database.

view_task(): Displays details of the selected task from the listbox in a message box.

edit_task(): Allows users to edit an existing task and updates it in both the in-memory list and the database.

delete_item(): Deletes the selected task from both the in-memory list and the database.

clear_item(): Deletes all tasks from both the in-memory list and the database.

retrieve_database(): Retrieves tasks from the MySQL database.

3.5 Main Loop:

The script enters the Tkinter main event loop (root.mainloop()), allowing the GUI to continuously handle user inputs and events.

3.6 Exception Handling:

The script uses try-except blocks to catch exceptions such as TclError and other exceptions during listbox operations and displays informative error messages.

3.7 Resizing Window:

The application window has a fixed minimum size but can be resized within specified limits.

3.8 Visual Feedback:

The GUI provides visual feedback to users. For example, buttons change color on hover, and the selected task in the listbox has distinct background and foreground colors.

3.9 Icon Setup:

The script loads an icon image ("icon.png") and sets it as the icon for the application window.

3.10 Task Management:

Users can add, view, edit, and delete tasks through the corresponding buttons, and the changes are reflected in both the in-memory list and the MySQL database.

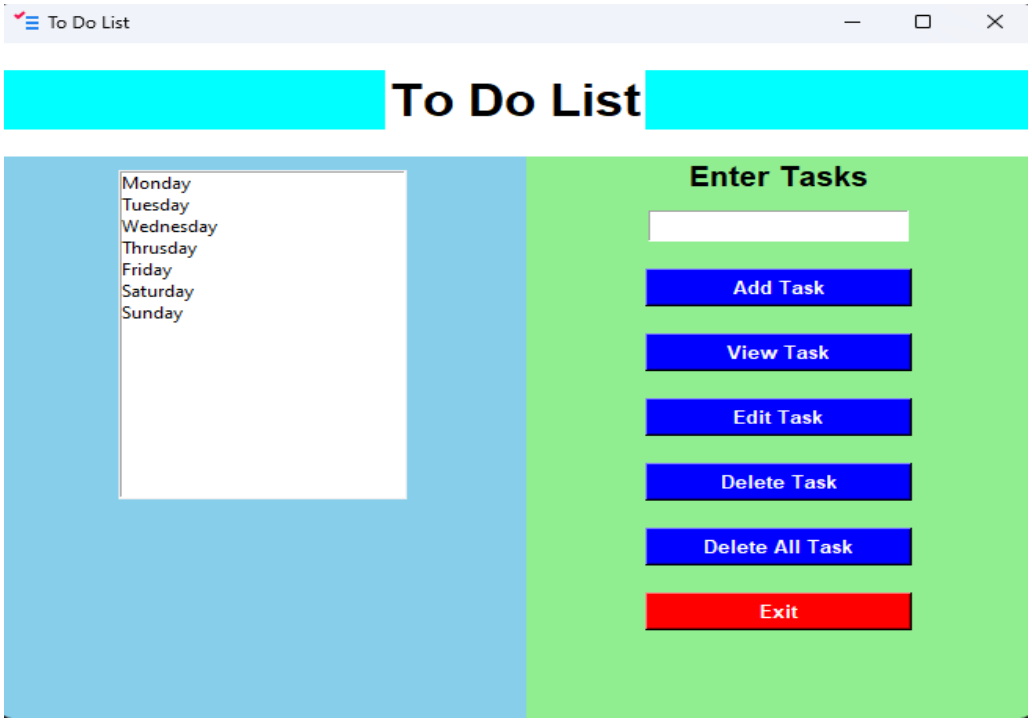
3.11 Database Integration:

MySQL is used for data storage, allowing tasks to persist across different sessions of the application.

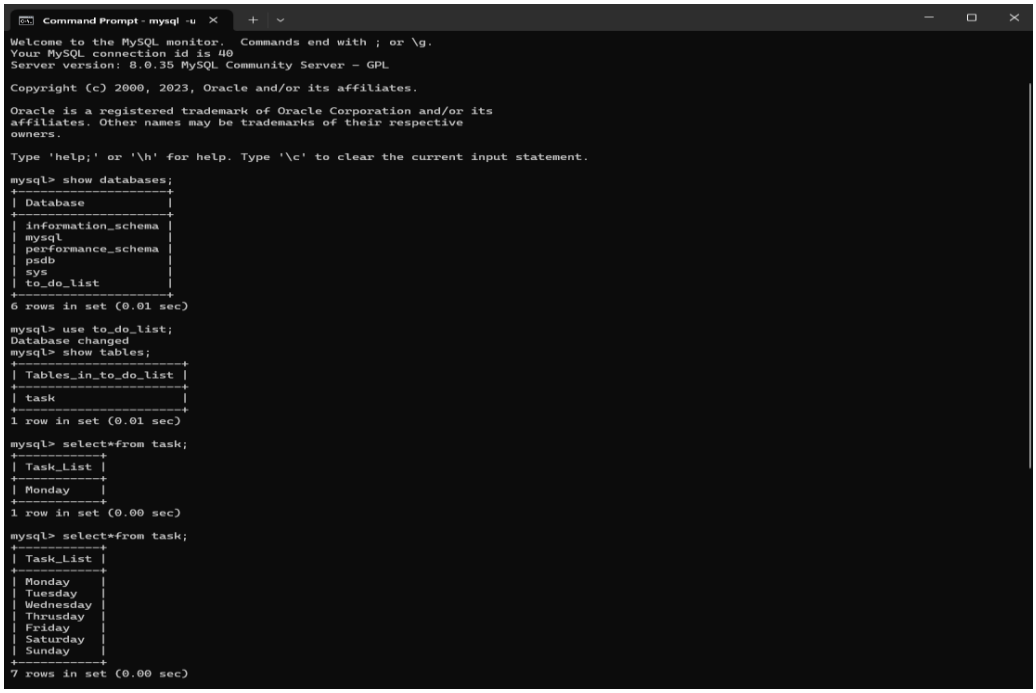
Chapter 4

4. Result/Output

App Interface



Database Snapshot



Chapter 5

5. Conclusion and Future Scope

5.1 Conclusion

The To-Do List App in Python, developed using Tkinter, provides a simple yet effective solution for managing tasks. Through the implementation of core functionalities such as adding, viewing, editing, and deleting tasks, the application offers users a streamlined experience for organizing their responsibilities. The user-friendly interface, error handling mechanisms, and scalability for future enhancements contribute to the overall utility of the application. Ease of Use- The graphical user interface is designed for simplicity, allowing users to manage tasks without unnecessary complexity. Error Handling- Robust error handling ensures that users receive informative messages and prompts, guiding them through the task management process. Scalability-The modular structure of the code and consideration for future development make the application adaptable to additional features and improvements.

5.2 Future Scope

While the To-Do List App provides a solid foundation for task management, there are several areas for potential future development and enhancements:

Task Prioritization- Implementing a priority system for tasks would allow users to categorize and focus on high-priority items.

Due Dates and Reminders- Introducing due dates and reminders would enhance the time-sensitive nature of tasks.

User Authentication- Incorporating user accounts and authentication mechanisms could enable personalized task management for multiple users.

Data Persistence- Adding the ability to save tasks between sessions ensures that users do not lose their task list when restarting the application.

Improved User Interface- Enhancing the visual design and layout based on user feedback can contribute to a more aesthetically pleasing and intuitive interface.

Mobile Compatibility-Adapting the application for mobile devices would extend its accessibility and usability.

Collaborative Features- Enabling collaboration on tasks among multiple users could be beneficial for team-based projects.

Analytics and Insights- Incorporating analytics features to track task completion rates and provide insights into task management patterns.

Reference

1. Tkinter Documentation: https://docs.python.org/3/library/tkinter.html
2. MySQL Connector Documentation: https://dev.mysql.com/doc/connector-python/en/
3. https://edube.org/learn/pcpp1-4-gui-programming/python-professional-course-series-gui-programming
4. https://www.javatpoint.com/simple-to-do-list-gui-application-in-python