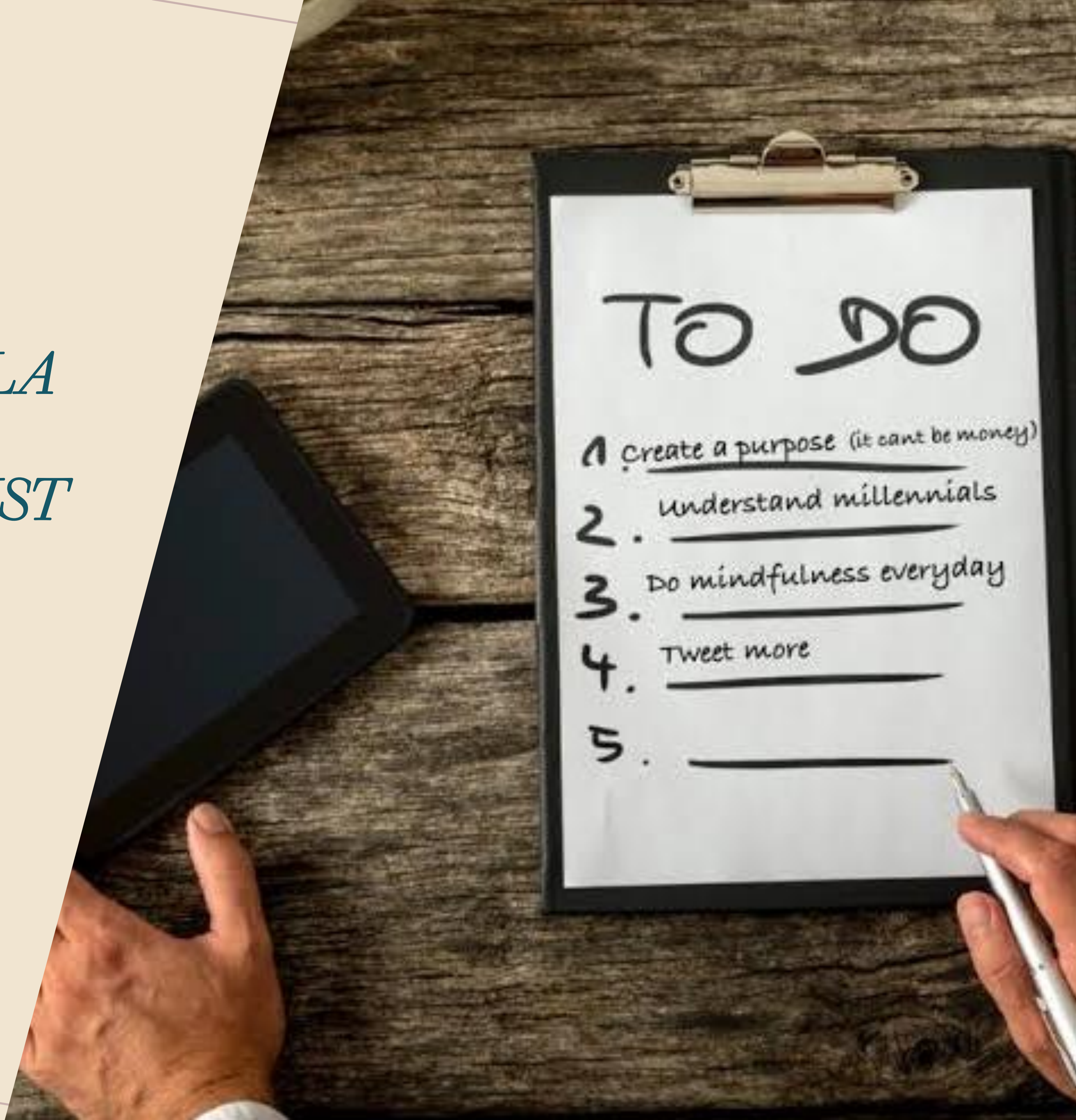
NAME :- SHIRISHA NAMBALA

PROJECT TITLE : TO-DO-LIST

MICRO IT


TO DO
1 Create a purpose (it cant be money)
2. Understand millennials
3. Do mindfulness everyday
4. Tweet more
5.

-: TO – DO – LIST:-

INTRODUCTION:-

The To-Do List Application is a simple and efficient task management tool developed using Python and Tkinter . It enables users to create, edit, categorize, complete, and delete tasks with ease. This application is designed to help improve productivity and time management by providing a clear and organized way to track daily activities. With features like task categorization and persistent data storage, it serves as a practical solution for managing personal and professional responsibilities

OBJECTIVE

The objective of this to-do list application is to help users manage their daily tasks efficiently. It allows users to add, categorize, edit, complete, and delete tasks, making it easier to stay organized and focused. The app is designed to improve productivity and time management through a simple, user-friendly interface with persistent task storage.

FEATURES:-

1) Add new tasks with optional categories.

2) Mark tasks as completed (✓ check mark added).

3) Edit existing tasks

4) Delete tasks

5) Save tasks locally using JSON

6) Load tasks automatically when the app starts

7) Simple and intuitive GUI using Tkinter

TOOLS AND TECHNOLOGIES USED:

- **Technologies:-**

1. Python – Core programming language for logic and application flow
2. Tkinter – GUI library for creating the desktop interface.
3. JSON – Used for data storage (saving and loading tasks)

- **Tools:-**

1. Visual Studio Code / PyCharm / IDLE – Code editor/IDE used for writing and debugging
2. Git – Version control tool to manage code history
3. GitHub – Platform for hosting and sharing the code repository
4. Windows/macOS/Linux – Operating system for running the application
5. Command Line / Terminal – For running scripts and Git commands

SYSTEM DESIGN AND ARCHITECTURE:-

TASK FLOW:-

Users can enter a new task using the input field. Tasks are displayed in a list box immediately upon addition

Users can mark tasks as done, which visually updates the task

Tasks can be deleted anytime from the list.

The flow ensures a seamless and user-friendly task management experience.

IMPLEMENTATION DETAILS:-

LOGIC: Task data is added dynamically using .insert() ,Empty inputs are handled with warning dialogs. Completed tasks are shown with a checkmark prefix

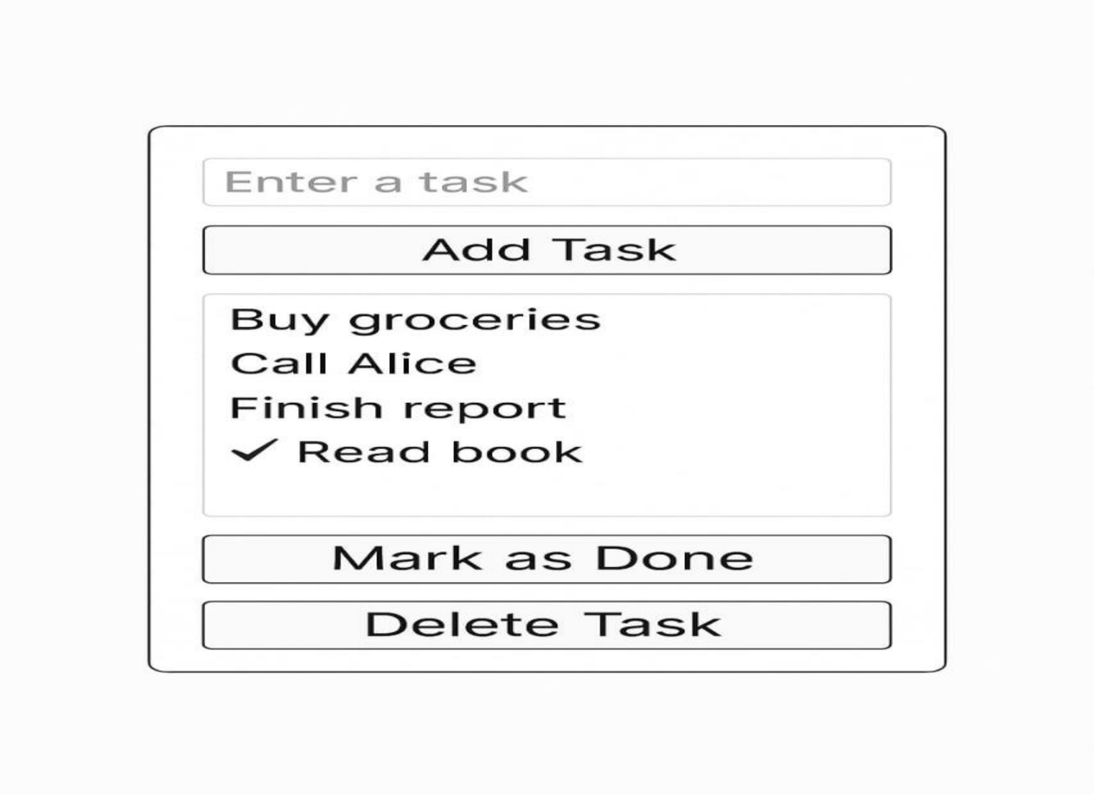
KEY FUNCTIONS :

1. add_task() – Adds a new task to the list_
2. done() – Marks a selected task as complete
3. _task() – Deletes the selected task

CODE STRUCTURE:

Main File – Contains GUI, logic, and event handling using Tkinter .Modular Functions – Separate reusable functions for add, done, and delete .UI Layout – Built using pack() with padding for spacing

-:USER INTERFACE:-



Buy groceries
Call Alice
Finish report
✓ Read book

1. Simple and intuitive design
2. Input field to enter tasks
3. Buttons: Add Task, Mark as Done, Delete Task
4. Task list display with real-time updates
5. Warning alerts for empty input or no selection

-:CHALLENGES FACED:-

->Ensuring users don’t add empty or meaningless tasks was important to maintain the quality and purpose of the list.

->Managing situations where users try to delete or mark a task without selecting it required careful error handling.

->Creating a simple and intuitive interface took time to balance functionality with ease of use for all types of users

->Clearly marking completed tasks without making the list look cluttered was a challenge in keeping the UI clean.

->Making sure the application runs smoothly even with many tasks added was essential to maintain performance.

FUTURE SCOPE:-

->Add reminders and due dates to help users manage deadlines more efficiently.

->Introduce task priority levels to organize tasks based on urgency and importance.

->Enable cloud storage or database integration for saving tasks across multiple devices securely.

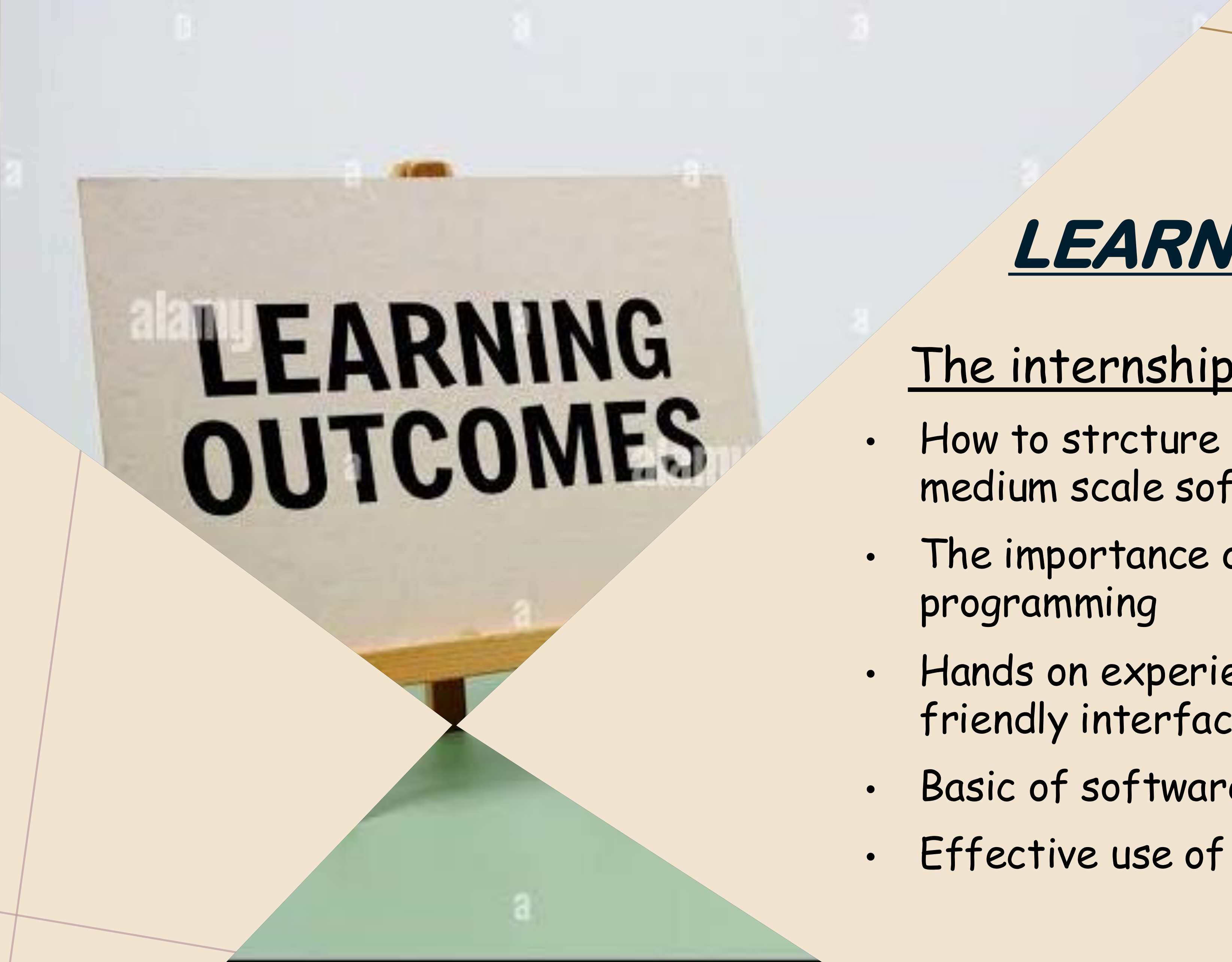
SYSTEM REQUIREMENTS:-

Python 3.x installed

Tkinter library (comes pre-installed with Python)

OS: Windows, macOS, or Linux

Any text/code editor for reviewing code



LEARNING OUTCOMES:-

The internship project taught me:

- How to strcture and manage a small to medium scale software project.
- The importance of clean code and modular programming
- Hands on experience in creating user friendly interface
- Basic of software testing and debugging
- Effective use of version control systems

CONCLUSION:-

This To-Do List project helped me strengthen both my technical and organizational skills. Though it appears simple, the process involved thoughtful planning, clean UI design, logic building, and attention to user experience. It provided a strong foundation in frontend development and task management principles.By working on this project, I learned how to build scalable interfaces, handle task data effectively, and deliver a smooth user workflow — giving me valuable insight into real-world software practices