IMAGE VIEWER USING TKINTER

NAME : DILLI GANESH

CYBERNAUT PROJECT INTERN

MONTH 3 (PROJECT 2)

TEAM : 8

1. INTRODUCTION

The **Image Viewer Using Tkinter** is a Python-based desktop application that allows users to load, display, and navigate images stored on their local computer. The project is developed using the **Tkinter GUI framework** and the **Pillow (PIL)** library for image processing.

This application demonstrates the practical usage of **Graphical User Interface (GUI) programming**, **file handling**, and **image resizing techniques** in Python. The main goal of the project is to provide a **simple, lightweight, and easy-to-use image viewing tool**.

2. PROJECT MOTIVATION

Modern image viewers often include complex features that consume more system resources. Beginners and educational projects require a **simple and understandable application** that focuses on core functionality.

This project was motivated by the need to:

- Understand GUI development using Tkinter
- Learn image handling using Python
- Build a clean and minimal application for academic use

3. OBJECTIVES OF THE PROJECT

The key objectives of the Image Viewer Using Tkinter project are:

- To design a **user-friendly graphical interface**
- To load and display images from a selected folder
- To support common image formats such as **JPEG, PNG, and BMP**
- To provide navigation buttons for browsing images
- To handle errors gracefully
- To create a resizable window that adapts to image size

4. SCOPE OF THE PROJECT

The scope of this project includes:

- Desktop application development
- Local image file viewing
- GUI-based user interaction

The project does not include:

- Image editing features
- Online image loading
- Database integration

5. TECHNOLOGIES USED

5.1 PYTHON PROGRAMMING LANGUAGE

Python is used as the core language due to its simplicity, readability, and strong support for GUI and image processing libraries.

5.2 TKINTER

Tkinter is Python’s standard GUI library. It is lightweight, easy to learn, and suitable for small to medium desktop applications.

5.3 PILLOW (PIL)

Pillow is used to load images, resize them, and support multiple image formats.

6. SYSTEM REQUIREMENTS

HARDWARE

- Minimum 4 GB RAM
- Keyboard and Mouse
- Desktop or Laptop

SOFTWARE

- Python 3.x
- Pillow library
- Operating System: Windows / Linux / macOS
- VS Code (IDE)

7. FUNCTIONAL REQUIREMENTS

- The system shall allow the user to load an image folder
- The system shall display images in supported formats
- The system shall provide Next and Previous navigation
- The system shall resize images dynamically
- The system shall display error messages when required
- The system shall provide an Exit option

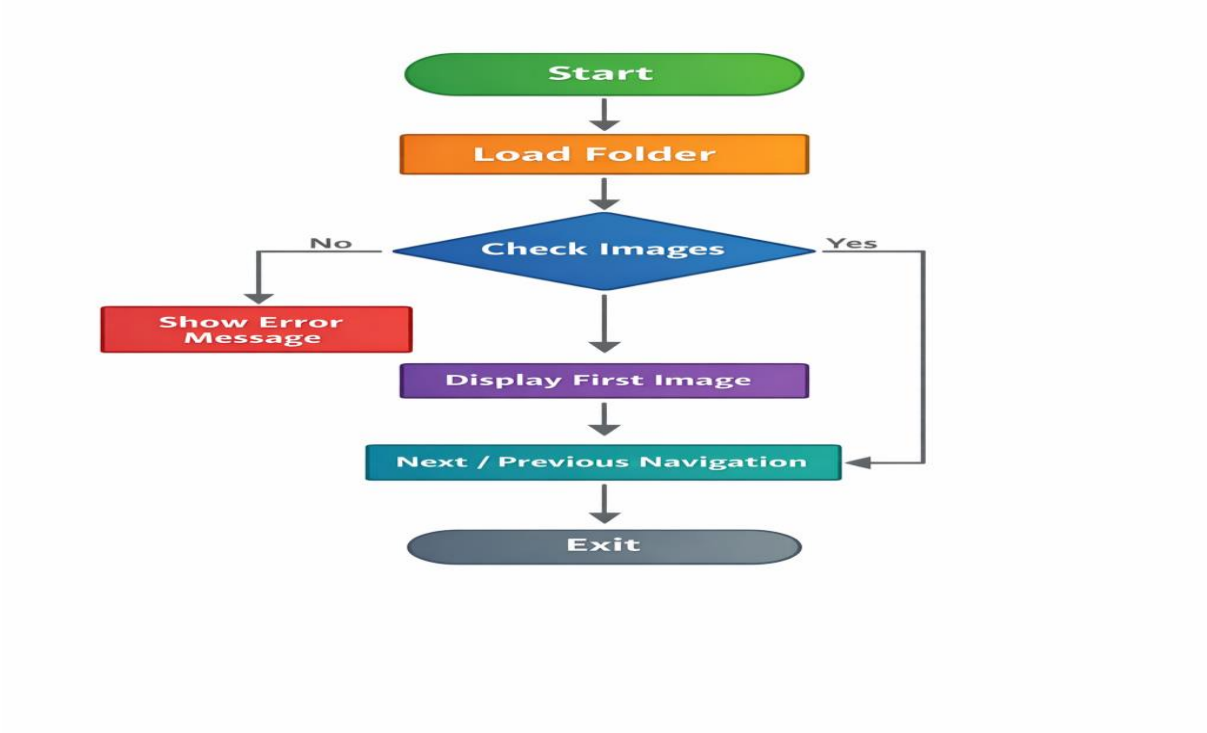
8. NON – FUNCTIONAL REQUIREMENTS

- Easy to use interface
- Fast image loading
- Low memory consumption
- Simple and clean design

9. SYSTEM DESIGN

FLOWCHART EXPLANIATION

The system begins by launching the GUI window. The user selects a folder using the Load Folder button. The system checks for supported images. If images are found, the first image is displayed. Users can navigate using Next and Previous buttons. If no images are found, an error message is displayed. The application terminates when the Exit button is pressed.



10. MODULE DESCRIPTION

1. FOLDER SELECTION MODULE

This module allows users to select a directory containing image files. It validates the selected folder and filters supported image formats.

2. IMAGE DISPLAY MODULE

This module loads images using the Pillow library and resizes them dynamically while preserving the aspect ratio.

3. NAVIGATION MODULE

This module manages the Next and Previous operations to browse through images sequentially.

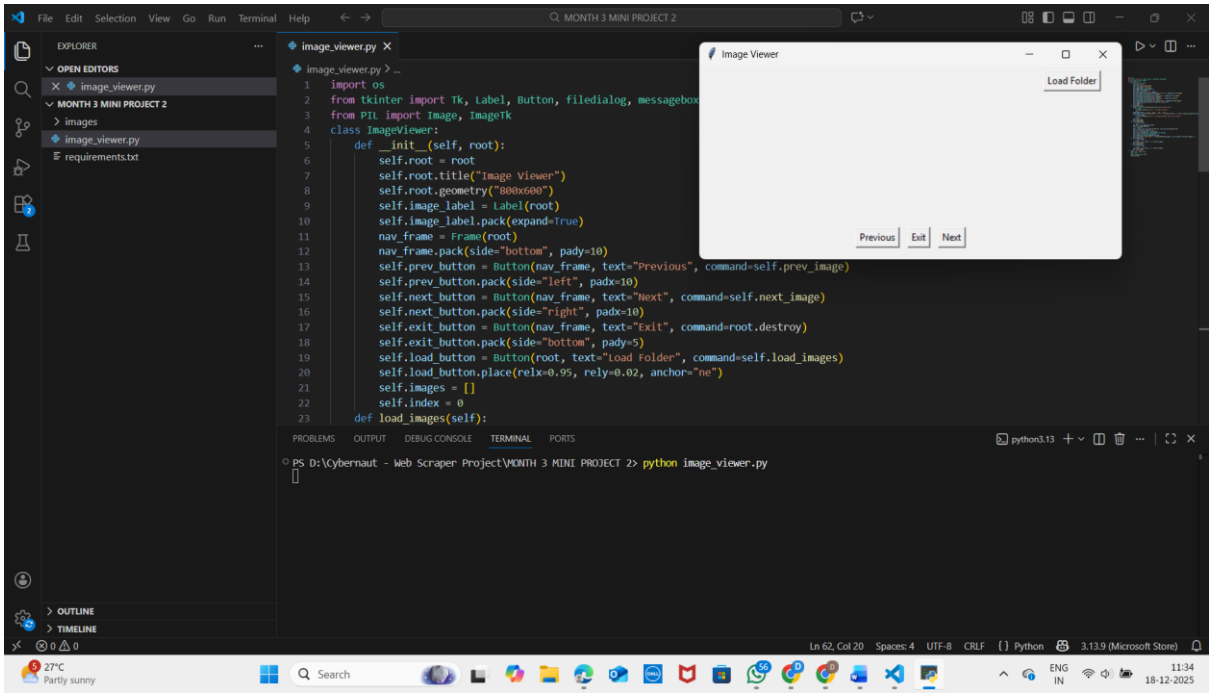
4. ERROR HANDLING MODULE

This module displays appropriate error messages when invalid operations occur.

11. OUTPUT SPECIFICATION

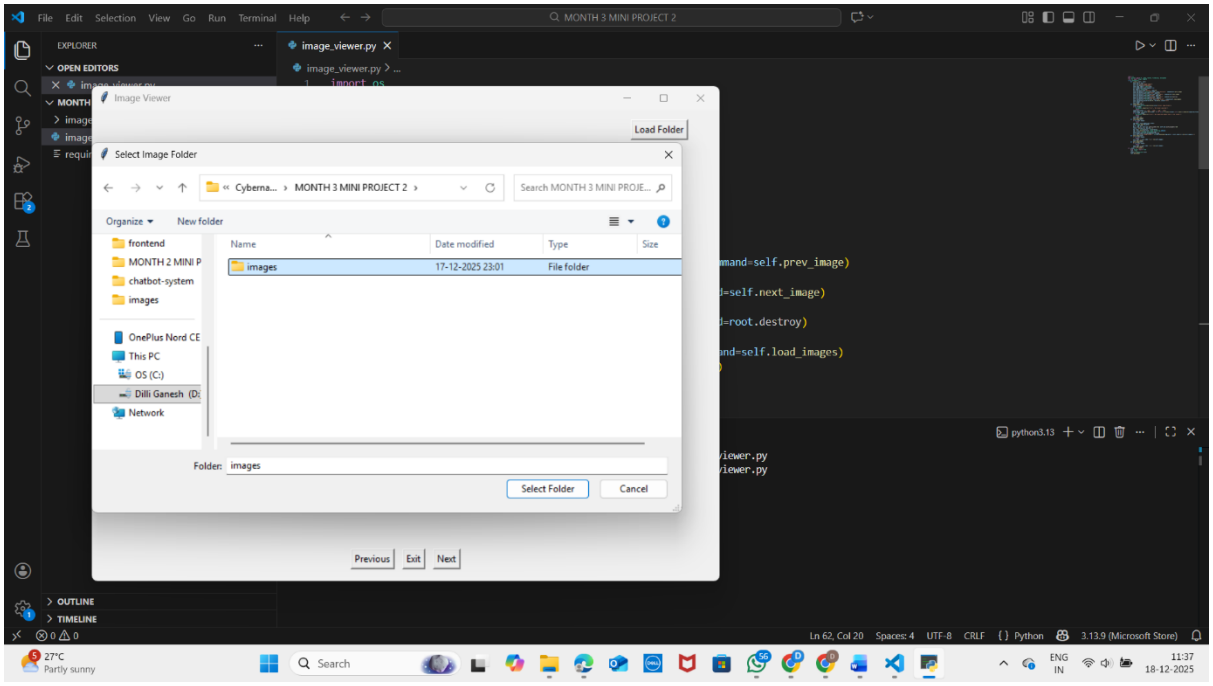
OUTPUT 1 : APPLICATION START

- GUI window opens with:
  - Load Folder button at the top-right corner
  - Navigation buttons at the bottom
  - Image display area in the center



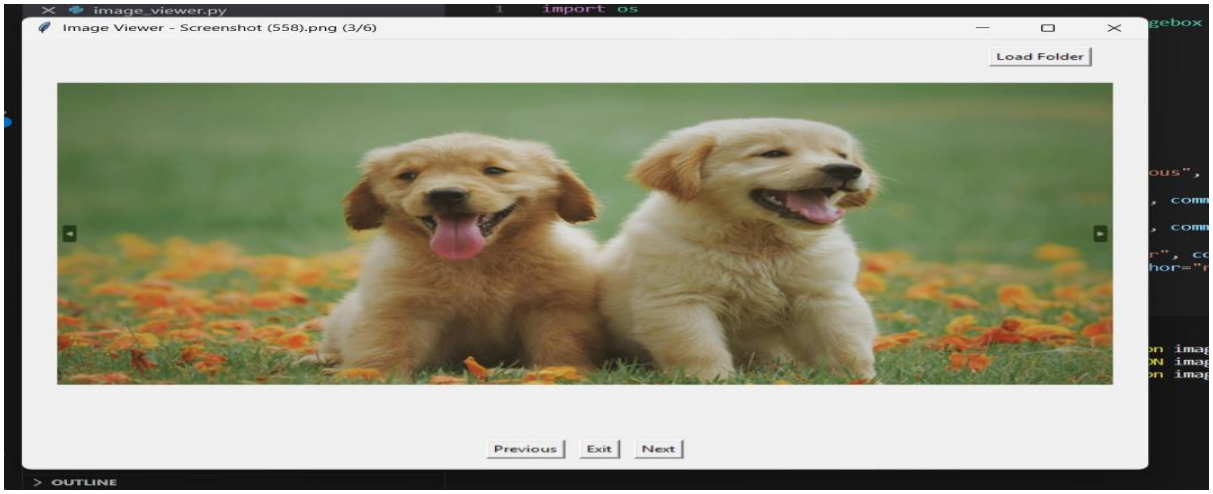
OUTPUT 2 : FOLDER SELECTION

- A dialog box opens for folder selection.
- User selects a valid image folder.



OUTPUT 3 : IMAGE DISPLAY

- First image from the folder is displayed.
- Image is resized to fit the window.
- Window title shows image name and position.



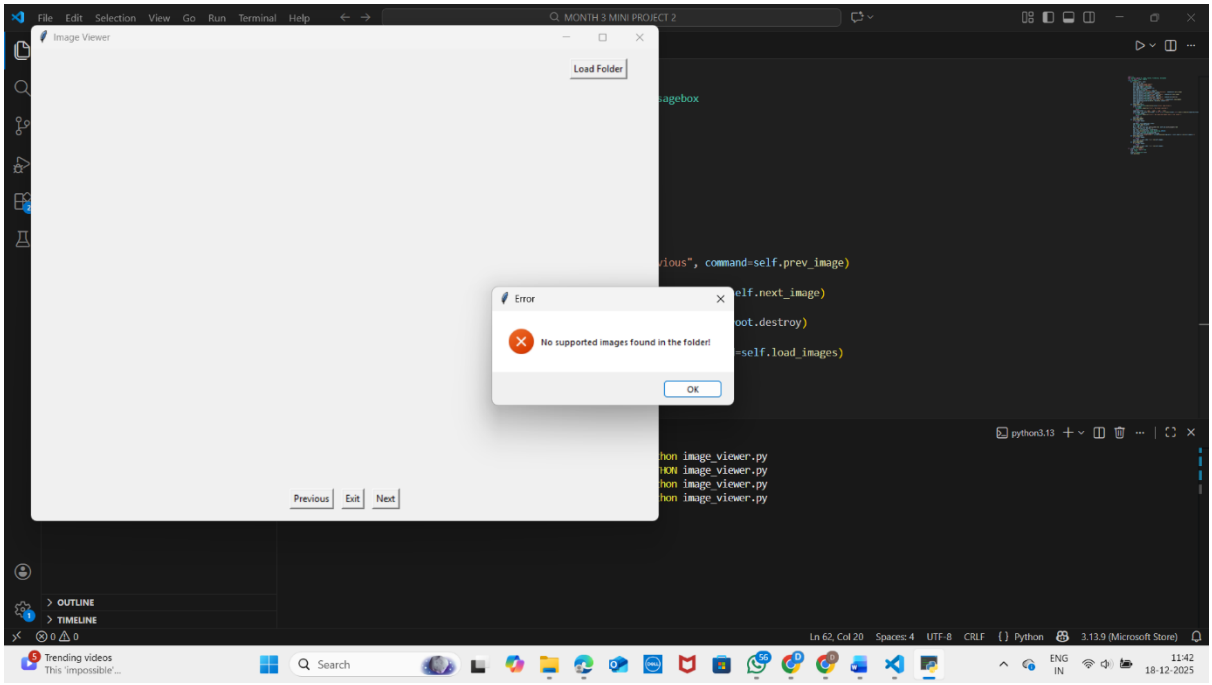
OUTPUT 4 : NAVIGATION

- Clicking Next displays the next image.
- Clicking Previous displays the previous image.

OUTPUT 5 : ERROR HANDLING

If no images are found:

Error: No supported images found in the selected folder.



OUTPUT 6 : EXIT

- Clicking Exit closes the application.

12. CHALLENGES FACED

- Managing dynamic image resizing.
- Handling unsupported image formats.
- Updating deprecated functions in Pillow.
- Designing a clean and intuitive interface.

13. FUTURE ENHANCEMENTS

- Keyboard-based navigation.

- Zoom and slideshow features.
- Support for additional image formats.
- Improved UI themes.

14. CONCLUSION

The **Image Viewer Application** is a simple yet effective project that demonstrates the use of Python for GUI development and image processing. The project successfully meets its objectives and provides a user-friendly experience. It also serves as a strong foundation for implementing advanced features in the future.