# Project Report

**Student Name:** Pawan Raj
**UID:** 24MCA20152
**Branch:** MCA
**Section/Group:** 3-A
**Semester:** 1st
**Date of Performance:** 29/10/2024
**Subject Name:** PYTHON PROGRAMMING LAB
**Subject Code:** 24CAH-606

## 1. Aim of the project:

Make a Image Viewer app with tkiner.

## 2. Software Requirements:

1. **Operating System**:
   - **Windows**: Windows 10 or later. ○ **macOS**: macOS 10.14 (Mojave) or later.
   - **Linux**: Any modern Linux distribution (e.g., Ubuntu 20.04 LTS, Fedora, etc.).
2. **Python Installation**:
   - **Python Version**: Python 3.12.6 or later. Download the latest version from the official Python website\

**3. Install Anaconda and Jupyter Notebook:**

- Downloads and install Anaconda from https://repo.anaconda.com/archive/Anaconda32022.05-Windows-x86_64.exe.
- Open "Anaconda Prompt" by finding it in the windows (start) Menu. ○ Type the command in (python - -version) Anaconda was installed.

**4. Start Jupyter Notebook:**

- Type the command in (Jupyter Notebook") to Start Jupyter Notebook.

# 3. Project Overview:

This project, 'Simple Image Viewer', is a GUI-based Python application developed using the Tkinter library and PIL (Python Imaging Library) for displaying images from a selected folder. The application allows users to browse images in a folder using 'Next' and 'Previous' and ‘Zoom in’ & ‘Zoom out’ buttons and provides a simple, user-friendly interface for viewing images

## Modules Used:

1. Tkinter: Used for creating the graphical user interface (GUI).
2. PIL (Pillow): Used for image processing tasks like opening, resizing, and displaying images.
3. os: Used for file and folder navigation.

## Code Explanation:

1. **Importing Modules:**
   - Imports the necessary modules for GUI creation, file management, and image processing.

2. **ImageViewer Class:**
   - Manages the primary window (GUI) and its functionalities.
   - Initializes variables for tracking images and sets up the interface layout.

3. **create_widgets Method:**
   - Creates the GUI components (widgets) including frames and buttons.
   - 'Open Folder' button to select image folder, and 'Previous', 'Next', and 'Exit' buttons.

4. **open_folder Method:**
   - Opens a dialog for selecting a folder, reads all image files, and updates the image list.

5. **show_image Method:**

- Displays the current image in the window by loading, resizing, and setting it in the GUI.

6. **Navigation Methods:**

- show_previous: Displays the previous image in the folder list.
- show_next: Displays the next image in the folder list.

## Main Functionality:

This program allows users to open a folder with images, view images in sequence using navigation buttons, and exit the application. It provides a simple interface where images are resized to fit within the display area while maintaining aspect ratio.

# 4. CODE:

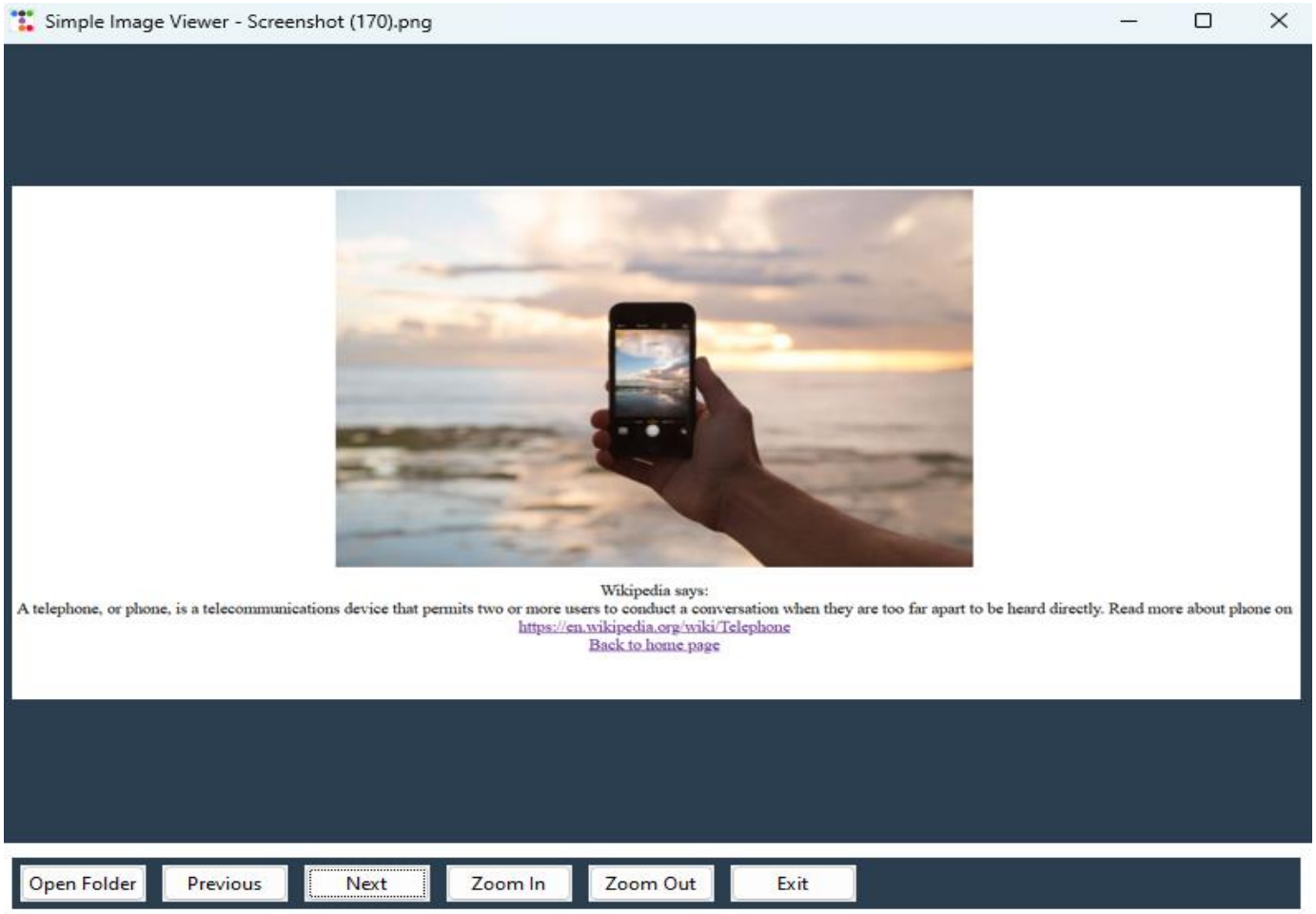
```
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
import os
import ttkbootstrap as ttk

class ImageViewer:
    def __init__(self, master):
        self.master = master
        self.master.title("Simple Image Viewer, By Pawan Raj")
        self.master.geometry("800x600")
        self.master.configure(bg="#FFFFFF")

        self.image_list = []
```

```
        self.current_image = 0
        self.zoom_level = 1.0
        self.create_widgets()

    def create_widgets(self):
        self.image_frame = tk.Frame(self.master, bg="black")
        self.image_frame.pack(side=tk.TOP, fill=tk.BOTH, expand=True)

        self.image_label = tk.Label(self.image_frame, bg="black")
        self.image_label.pack(fill=tk.BOTH, expand=True)

        button_frame = tk.Frame(self.master, bg="#FFFFFF")
        button_frame.pack(side=tk.BOTTOM, fill=tk.X, padx=10, pady=10)

        style = ttk.Style()
        style.configure("TButton", font=("Arial", 12), padding=10)

        ttk.Button(button_frame, text="Open Folder", command=self.open_folder, style="TButton").grid(row=0,
column=0, padx=5, pady=5)
        ttk.Button(button_frame, text="Previous", command=self.show_previous, style="TButton").grid(row=0,
column=1, padx=5, pady=5)
        ttk.Button(button_frame, text="Next", command=self.show_next, style="TButton").grid(row=0, column=2,
padx=5, pady=5)
        ttk.Button(button_frame, text="Zoom In", command=self.zoom_in, style="TButton").grid(row=0, column=3,
padx=5, pady=5)
        ttk.Button(button_frame, text="Zoom Out", command=self.zoom_out, style="TButton").grid(row=0, column=4,
padx=5, pady=5)
        ttk.Button(button_frame, text="Exit", command=self.master.quit, style="TButton").grid(row=0, column=5,
padx=5, pady=5)
```

```
def open_folder(self):
    folder_path = filedialog.askdirectory()
    if folder_path:
        self.image_list = []
        for filename in os.listdir(folder_path):
            if filename.lower().endswith(('.png', '.jpg', '.jpeg', '.gif', '.bmp')):
                self.image_list.append(os.path.join(folder_path, filename))

        if self.image_list:
            self.current_image = 0
            self.show_image()
        else:
            messagebox.showinfo("Info", "No images found in the selected folder.")

def show_image(self):
    if 0 <= self.current_image < len(self.image_list):
        image_path = self.image_list[self.current_image]
        image = Image.open(image_path)

        width, height = image.size
        new_size = (int(width * self.zoom_level), int(height * self.zoom_level))
        image = image.resize(new_size, Image.LANCZOS)

        image.thumbnail((780, 520))

        photo = ImageTk.PhotoImage(image)
        self.image_label.config(image=photo)
        self.image_label.image = photo  # Keep a reference

        self.master.title(f"Simple Image Viewer - {os.path.basename(image_path)}")
```

```
else:
            self.image_label.config(image=None)
            self.master.title("Simple Image Viewer")

    def show_previous(self):
        if self.image_list:
            self.current_image = (self.current_image - 1) % len(self.image_list)
            self.show_image()

    def show_next(self):
        if self.image_list:
            self.current_image = (self.current_image + 1) % len(self.image_list)
            self.show_image()

    def zoom_in(self):
        self.zoom_level += 0.1
        self.show_image()

    def zoom_out(self):
        self.zoom_level = max(0.1, self.zoom_level - 0.1)
        self.show_image()

if __name__ == "__main__":
    root = ttk.Window(themename="superhero")
    app = ImageViewer(root)
    root.mainloop()
```

## 4.RESULT:


Simple Image Viewer - Screenshot (170).png
Wikipedia says:
A telephone, or phone, is a telecommunications device that permits two or more users to conduct a conversation when they are too far apart to be heard directly. Read more about phone on
https://en.wikipedia.org/wiki/Telephone
Back to home page
Open Folder
Previous
Next
Zoom In
Zoom Out
Exit

## 5. Learning outcomes (What I have learnt):

- **Understanding of Tkinter:** Gained knowledge on creating a basic application using Tkinter.
- **Folder and File Management:** Using file dialogs to open folders and read files allows for a dynamic application that can handle various user inputs.
- **Event Handling:** Implementing event handlers for button clicks allows the application to respond to user inputs (e.g., opening folders, navigating images).
- **Image Navigation:** Implementing features for navigating between images (previous, next) requires managing lists of images and maintaining the current index.

## 6. Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
|---|---|---|---|
| 1. | Worksheet | | 8 Marks |
| 2. | Viva | | 10 Marks |
| 3. | Simulation | | 12 Marks |
| | Total | | 30 Marks |

**Teacher Signature**