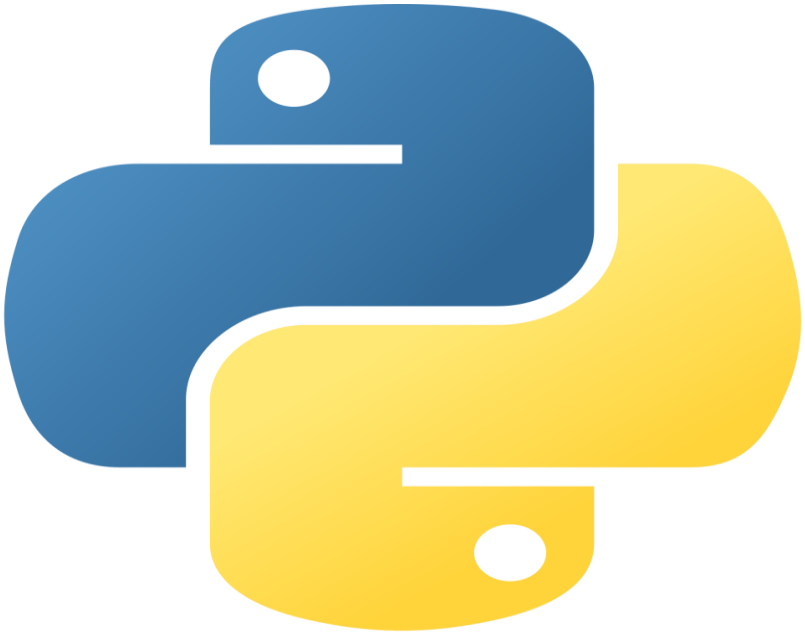
AMITY UNIVERSITY

AMITY INSTITUTE OF INFORMATION TECNOLOGY ,PATNA

CETPA Summer Training 2024



python™

Submitted by:-

Harsh Srivastava (08)
Raunak Kumar (02)
Rajni Shreasth (06)
Kishan Kumar (BCA)

Submitted to:-

Sourav Singh

Amity University Attendance Portal Documentation

Introduction

This Streamlit application serves as an attendance portal for Amity University. It allows users to mark attendance, view attendance records, and capture images from a webcam. The application utilizes Python with libraries such as Streamlit, OpenCV (cv2), Pillow (PIL), and os for image handling and data management.

Key Features

1. Mark Attendance:

- **Users can input student details (enrollment number, first name, last name) and mark their attendance.**
- **Attendance details are recorded in a CSV file (attendance_records.csv) along with timestamped entry.**
- **Offers the option to capture an image of the student using a webcam, which can be saved and optionally modified or deleted.**

2. View Attendance:

- **Displays a table of recorded attendance from the CSV file.**
- **Allows editing or deleting specific attendance records directly from the UI.**
- **Provides interactive options for managing attendance data effectively.**

3. Capture Image:

- Enables users to capture an image using the webcam, displaying it in the UI.
- Users can save the captured image and optionally modify it (rotate) or delete it directly from the UI.

Components and Functions

- **Streamlit UI Components:**
  - Uses Streamlit for creating a user-friendly web interface.
  - Includes navigation sidebar (st.sidebar) for menu options and main content area for displaying different functionalities (st.header, st.info, st.button, st.table, etc.).
- **Functions:**
  - mark_attendance(enrollment_number, first_name, last_name):
    - Records attendance by appending details to attendance_records.csv.
    - Captures an image from the webcam, allowing for modification and deletion.
  - capture_image():
    - Provides functionality to capture an image using the webcam.
    - Options to save, modify (rotate), or delete the captured image.

- view_attendance():
  - Reads attendance records from attendance_records.csv and displays them in a tabular format.
  - Allows editing or deleting attendance records based on user selection.

Usage

1. Mark Attendance:
   - Enter student details and click "Mark Attendance" to record attendance.
   - Optionally capture an image of the student using the webcam.
2. View Attendance:
   - Navigate to "View Attendance" to see a table of recorded attendance.
   - Edit or delete specific attendance records as needed.
3. Capture Image:
   - Select "Capture Image" to capture an image using the webcam.
   - Save, modify (rotate), or delete the captured image interactively.

Dependencies

- Streamlit: For creating the web-based user interface.
- OpenCV (cv2): For webcam access and image capture.

- **Pillow (PIL): For image manipulation and saving.**
- **os: For file operations such as image deletion.**

File Management

- **attendance_records.csv: Stores attendance records with columns for enrollment number, first name, last name, and timestamp.**

Notes

- **Ensure the webcam is correctly configured and accessible to capture images.**
- **Modifications to attendance data (edit/delete) are reflected immediately in attendance_records.csv.**

Conclusion

This application provides a robust solution for managing attendance records and capturing student images interactively. It leverages modern Python libraries and Streamlit's capabilities to offer a user-friendly experience for both recording and viewing attendance data.

Code:-

```
import streamlit as st
import csv
from datetime import datetime
import cv2
from PIL import Image
import numpy as np
import os

# Function to mark attendance and save to CSV
def mark_attendance(enrollment_number, first_name, last_name):
    # Append attendance record to CSV file
    with open('attendance_records.csv', 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([enrollment_number, first_name, last_name, datetime.now().strftime('%Y-%m-%d %H:%M:%S')])
    st.success(f"Attendance marked for {first_name} {last_name} (Enrollment: {enrollment_number})")

    # Option to capture image from webcam
    st.header("Capture Image from Webcam")

    # Initialize webcam capture
    cap = cv2.VideoCapture(0)

    if st.button("Capture Image", key='capture_button'):
        if cap.isOpened():
            ret, frame = cap.read()
```

```
        if ret:
            # Convert frame from BGR to RGB
            frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

            # Convert to PIL Image
            pil_image = Image.fromarray(frame_rgb)

            # Display captured image
            st.image(pil_image, caption='Captured Image',
use_column_width=True)

            # Save image to disk (optional)
            image_path =
f"captured_image_{datetime.now().strftime('%Y%m%d%H%M%S')}.jp
g"
            pil_image.save(image_path)

            st.success(f"Image captured and saved as {image_path}")

            # Option to delete or modify the captured image
            delete_option = st.checkbox("Delete Image")
            modify_option = st.checkbox("Modify Image")

            if delete_option:
                os.remove(image_path)
                st.warning("Image deleted.")

            if modify_option:
```

```
        # Provide options to modify the image (example: rotate, resize, etc.)
                rotation_angle = st.slider("Rotation Angle", -180, 180, 0)
                if st.button("Apply Rotation"):
                    modified_image = pil_image.rotate(rotation_angle)
st.image(modified_image, caption='Modified Image',
use_column_width=True)
                    modified_image.save(image_path)  # Save modified image

    # Release the webcam
    cap.release()

# Function to capture image from webcam and save
def capture_image():
    st.header("Capture Image from Webcam")

    # Initialize webcam capture
    cap = cv2.VideoCapture(0)

    if st.button("Capture Image", key='capture_button'):
        if cap.isOpened():
            ret, frame = cap.read()
            if ret:
                # Convert frame from BGR to RGB
                frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

                # Convert to PIL Image
                pil_image = Image.fromarray(frame_rgb)
```

```
            # Display captured image
            st.image(pil_image, caption='Captured Image',
use_column_width=True)

            # Save image to disk (optional)
            image_path =
f"captured_image_{datetime.now().strftime('%Y%m%d%H%M%S')}.jp
g"
            pil_image.save(image_path)

            st.success(f"Image captured and saved as {image_path}")

            # Option to delete or modify the captured image
            delete_option = st.checkbox("Delete Image")
            modify_option = st.checkbox("Modify Image")

            if delete_option:
               os.remove(image_path)
               st.warning("Image deleted.")

            if modify_option:
               # Provide options to modify the image (example: rotate, resize,
etc.)
               rotation_angle = st.slider("Rotation Angle", -180, 180, 0)
               if st.button("Apply Rotation"):
                  modified_image = pil_image.rotate(rotation_angle)
                  st.image(modified_image, caption='Modified Image',
use_column_width=True)
                  modified_image.save(image_path)
```

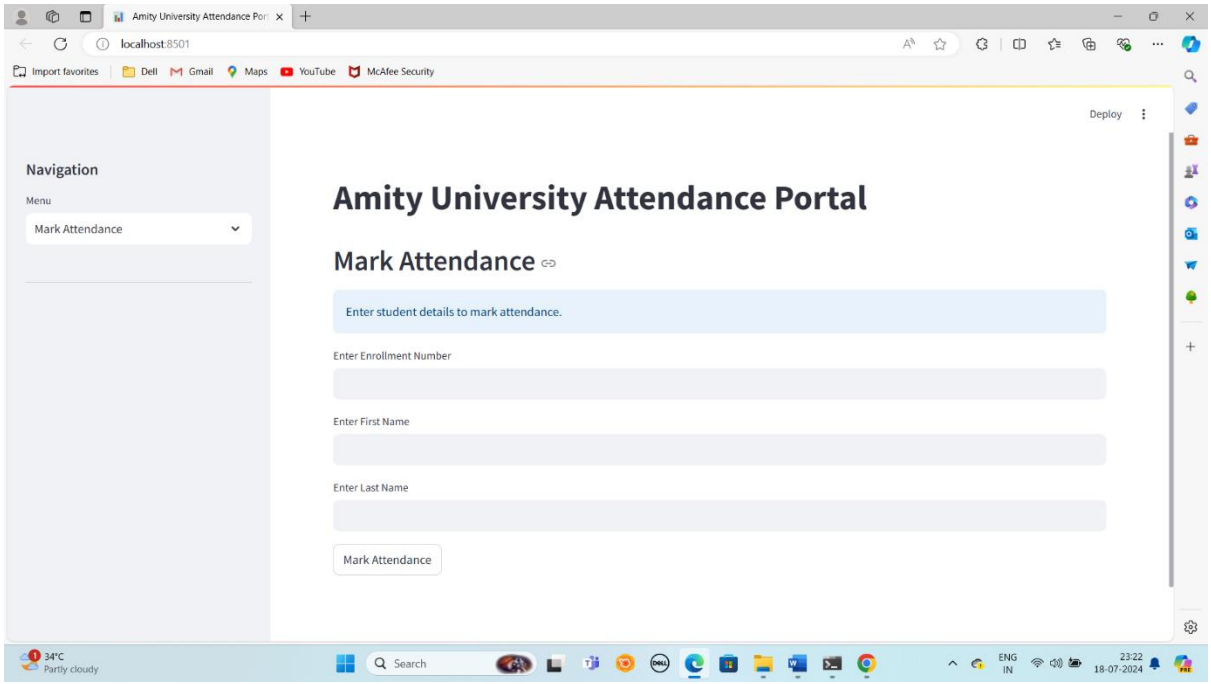
```
# Save modified image

    # Release the webcam
    cap.release()

# Function to view attendance records from CSV
def view_attendance():
    try:
        with open('attendance_records.csv', 'r') as file:
            reader = csv.reader(file)
            attendance_records = list(reader)

        # Display attendance records in a table
        st.write("### Attendance Records")
        if attendance_records:
            # Format data for table display
            formatted_records = []
            for record in attendance_records:
                formatted_record = {
                    "Enrollment Number": record[0],
                    "First Name": record[1],
                    "Last Name": record[2],
                    "Date & Time": record[3]
                }
                formatted_records.append(formatted_record)

            # Display as a table
```

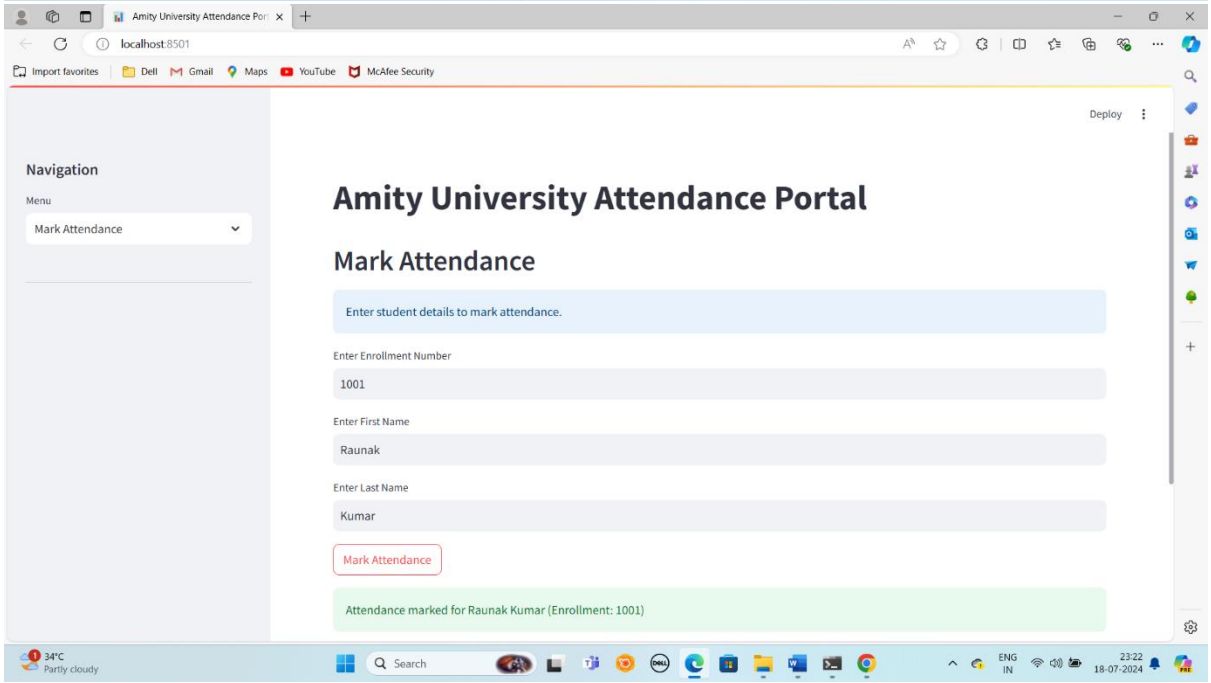
```
        st.table(formatted_records)


        # Modify attendance option
        modify_option = st.selectbox("Select an option", ["", "Edit Attendance", "Delete Attendance"])
        if modify_option == "Edit Attendance":
            edit_index = st.number_input("Enter the index of the record to edit", min_value=1, max_value=len(attendance_records))
            if st.button("Edit"):
                edit_record = attendance_records[edit_index - 1]
                st.write(f"Editing record: {edit_record}")
                # Provide fields to edit (if needed)
                # Example: You can update the attendance data structure based on requirements
        elif modify_option == "Delete Attendance":
            delete_index = st.number_input("Enter the index of the record to delete", min_value=1, max_value=len(attendance_records))
            if st.button("Delete"):
                deleted_record = attendance_records.pop(delete_index - 1)
                # Rewrite the CSV file without the deleted record
                with open('attendance_records.csv', 'w', newline='') as file:
                    writer = csv.writer(file)
                    writer.writerows(attendance_records)
                st.warning(f"Deleted record: {deleted_record}")
                st.success("Attendance record deleted successfully.")


    else:
        st.write("No attendance records found.")
```

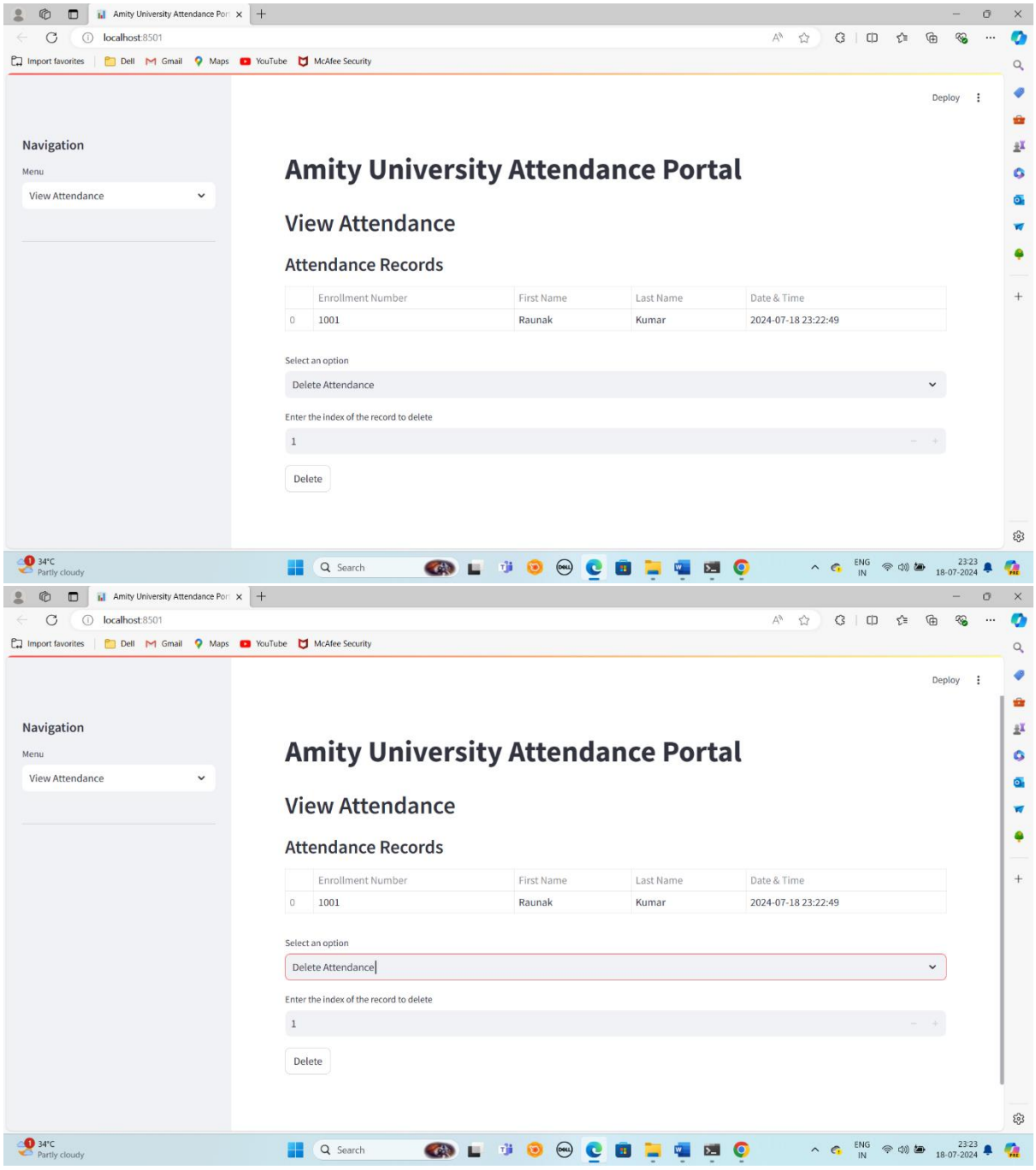
```
    except FileNotFoundError:
        st.write("No attendance records found.")

# Streamlit UI code
def main():
    st.set_page_config(page_title="Amity University Attendance Portal", page_icon=":bar_chart:", layout='wide')

    st.title("Amity University Attendance Portal")
    st.sidebar.header("Navigation")
    menu = ["Mark Attendance", "View Attendance", "Capture Image"]
    choice = st.sidebar.selectbox("Menu", menu)

    st.sidebar.markdown("---")

    if choice == "Mark Attendance":
        st.header("Mark Attendance")
        st.info("Enter student details to mark attendance.")
        enrollment_number = st.text_input("Enter Enrollment Number")
        first_name = st.text_input("Enter First Name")
        last_name = st.text_input("Enter Last Name")
        if st.button("Mark Attendance"):
            mark_attendance(enrollment_number, first_name, last_name)

    elif choice == "View Attendance":
        st.header("View Attendance")
        view_attendance()
```

```
    elif choice == "Capture Image":
        capture_image()

if __name__ == '__main__':
    main()
```

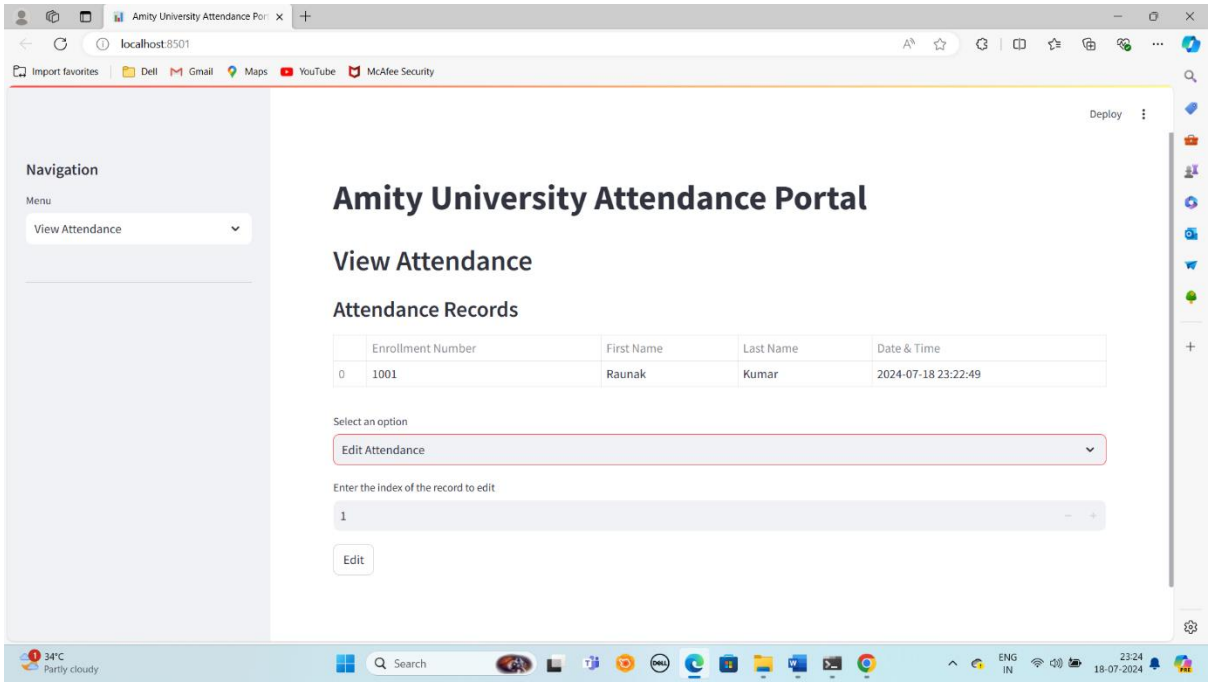
Output:-

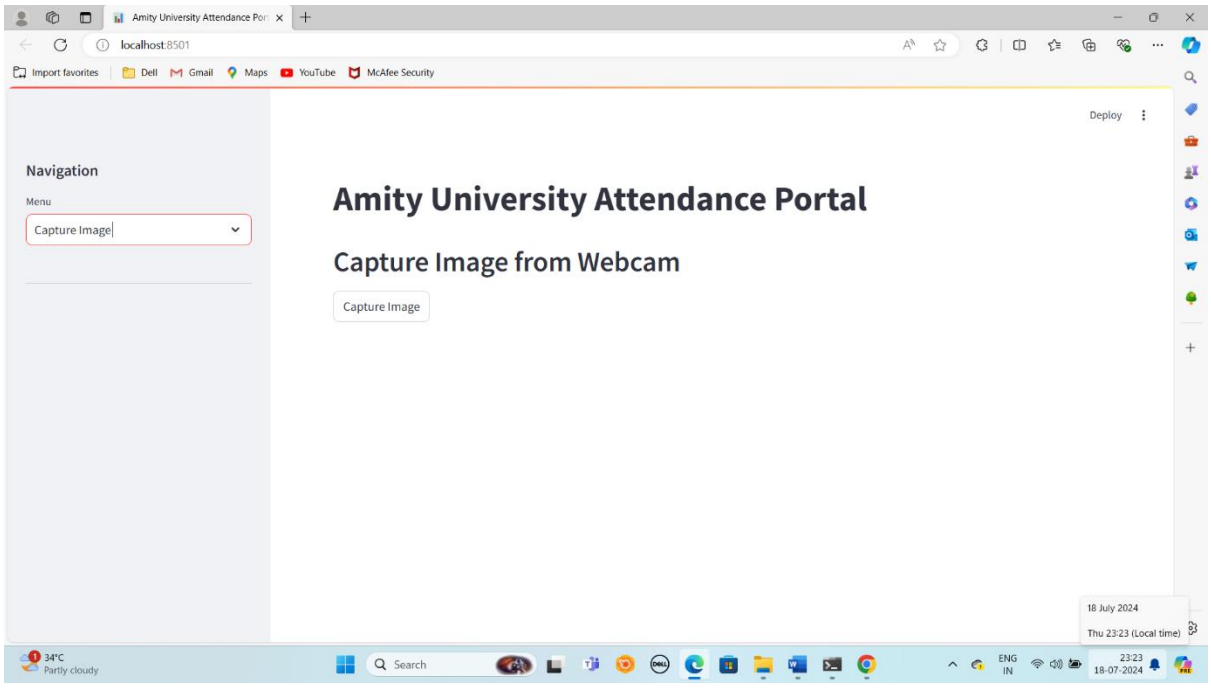





Amity University Attendance Portal
localhost:8501
Import favorites
Dell
Gmail
Maps
YouTube
McAfee Security
Deploy
Navigation
Menu
View Attendance
Amity University Attendance Portal
View Attendance
Attendance Records
Enrollment Number
First Name
Last Name
Date & Time
0
1001
Raunak
Kumar
2024-07-18 23:22:49
Select an option
Delete Attendance
Enter the index of the record to delete
1
Delete
34°C
Partly cloudy
Search
ENG
IN
23:23
18-07-2024

Amity University Attendance Portal
localhost:8501
Import favorites
Dell
Gmail
Maps
YouTube
McAfee Security
Deploy
Navigation
Menu
View Attendance
Amity University Attendance Portal
View Attendance
Attendance Records
Enrollment Number
First Name
Last Name
Date & Time
0
1001
Raunak
Kumar
2024-07-18 23:22:49
Select an option
Delete Attendance
Enter the index of the record to delete
1
Delete
34°C
Partly cloudy
Search
ENG
IN
23:23
18-07-2024


Amity University Attendance Portal
localhost:8501
Import favorites
Dell
Gmail
Maps
YouTube
McAfee Security
Deploy
Navigation
Menu
View Attendance
Amity University Attendance Portal
View Attendance
Attendance Records
Enrollment Number
First Name
Last Name
Date & Time
0
1001
Raunak
Kumar
2024-07-18 23:22:49
Select an option
Edit Attendance
Enter the index of the record to edit
1
Edit
34°C
Partly cloudy
Search
ENG
IN
23:24
18-07-2024


Amity University Attendance Portal
localhost:8501
Import favorites
Dell
Gmail
Maps
YouTube
McAfee Security
Deploy
Navigation
Menu
Capture Image
Amity University Attendance Portal
Capture Image from Webcam
Capture Image
18 July 2024
Thu 23:23 (Local time)
34°C
Partly cloudy
Search
ENG
IN
23:23
18-07-2024