**AMITY UNIVERSITY**

**AMITY INSTITUTE OF INFORMATION TECNOLOGY ,PATNA**

**CETPA Summer Training 2024**

****

**Submitted by:- Submitted to:-**

**Harsh Srivastava (08)**

**Raunak Kumar (02) Sourav Raj**

**Rajni Shreasth (06)**

**MCA (2023-2025)**

**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')}.jpg"**

**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')}.jpg"**

**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()**

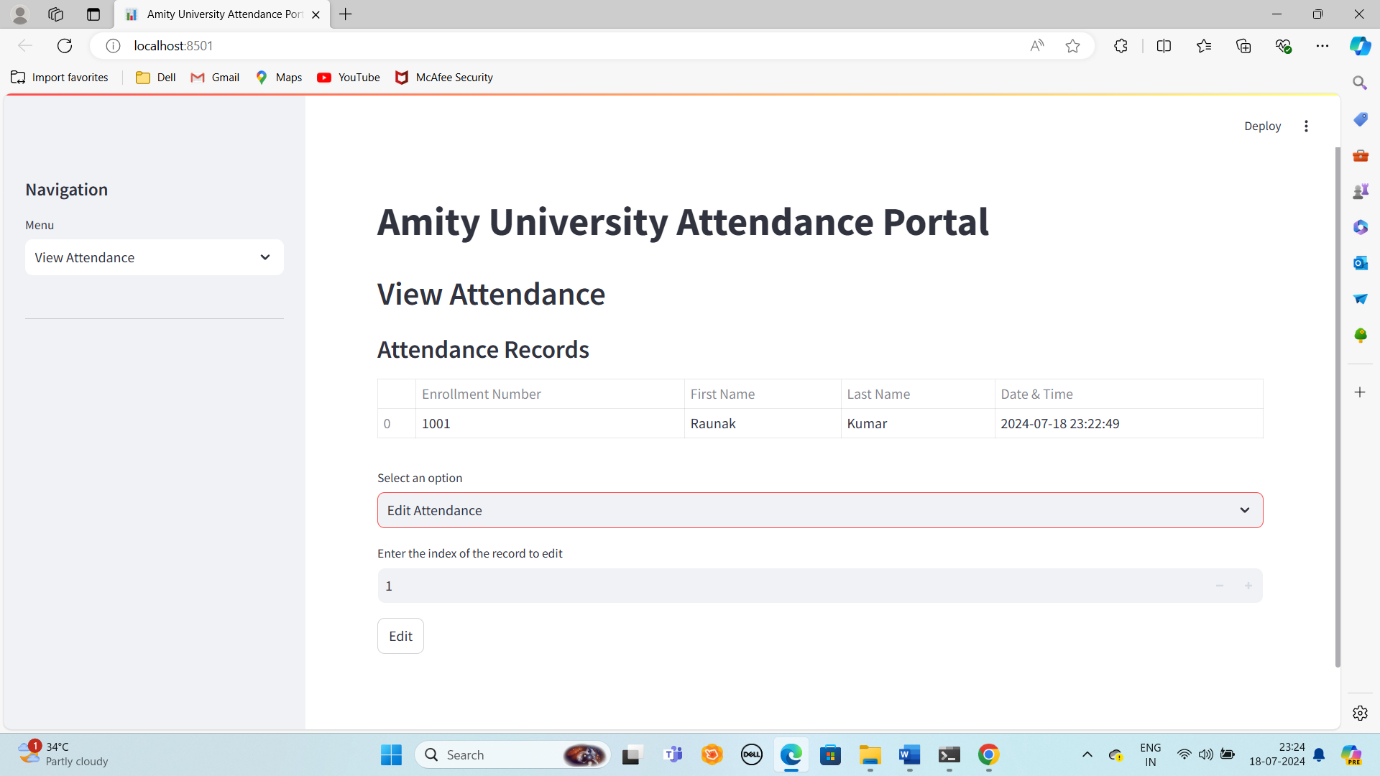
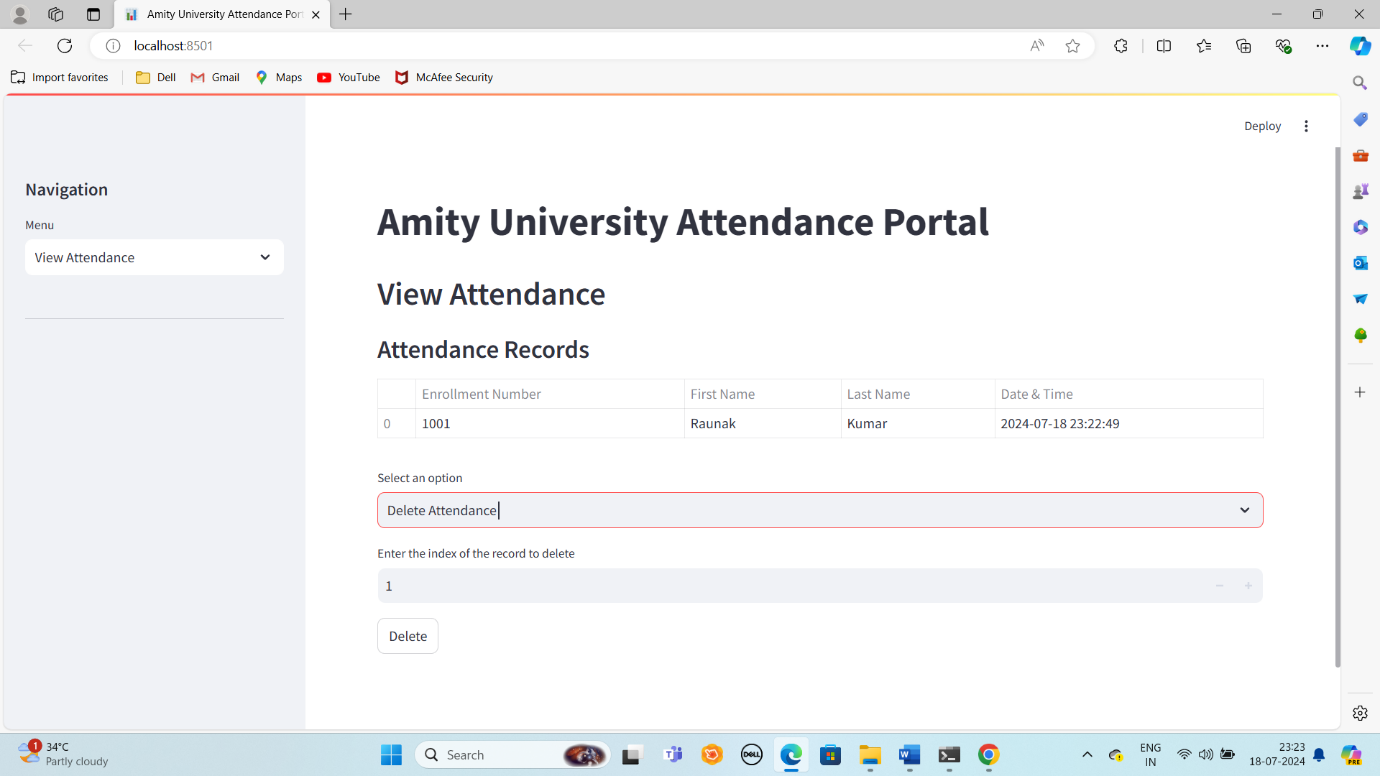
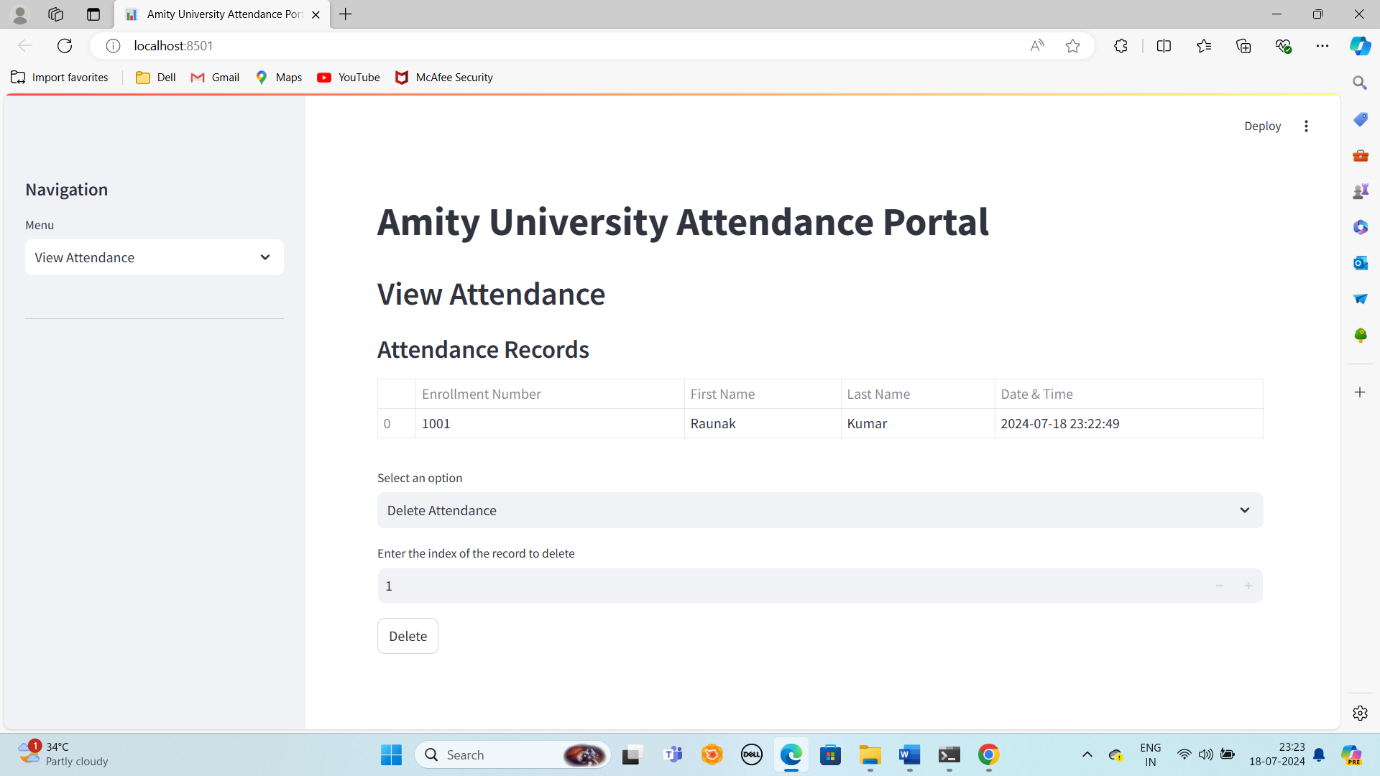
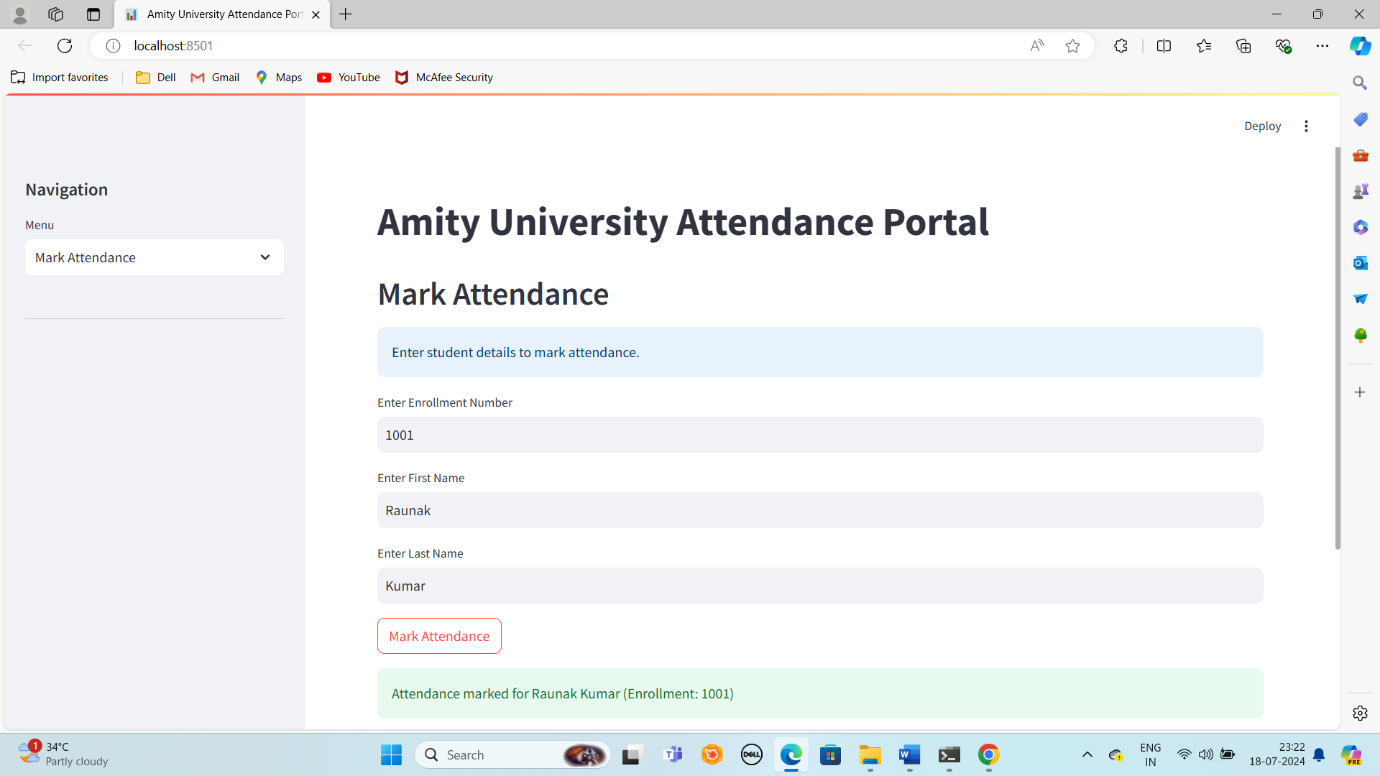
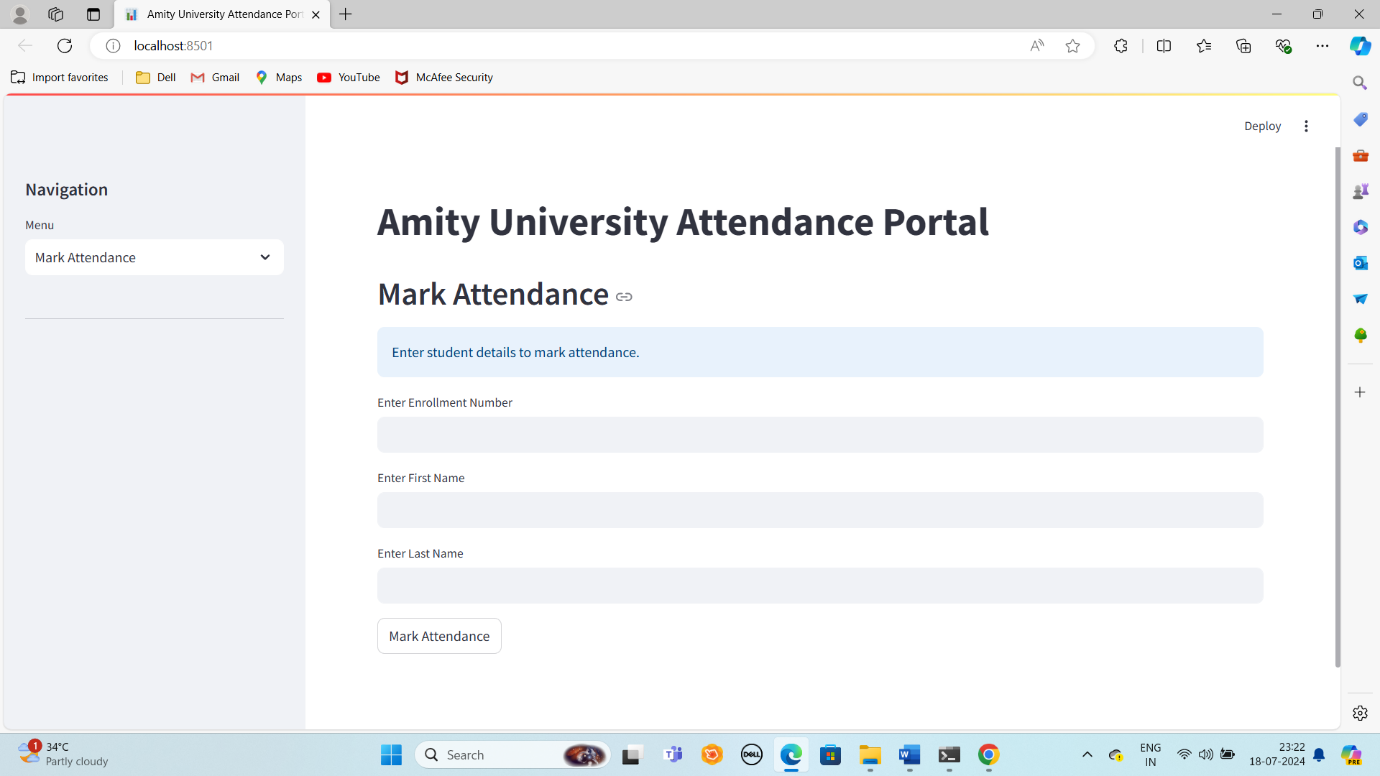
**elif choice == "Capture Image":**

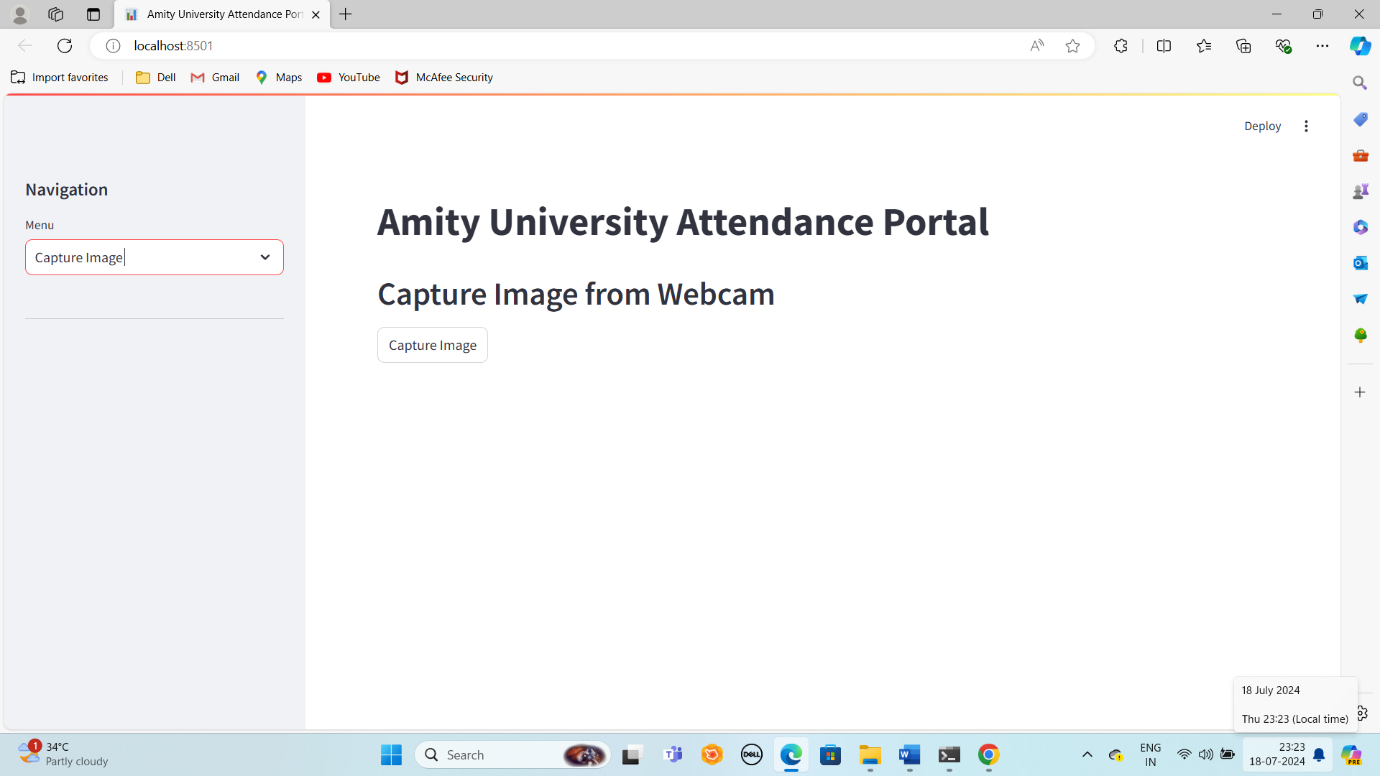
**capture\_image()**

**if \_\_name\_\_ == '\_\_main\_\_':**

**main()**

**Output:-**

****

****