Attendance Log API

-- By Team Machine Learners

**Team Members**:

1. Tirth G Shah
2. Mayurkumar Patel
3. Deep Patel
4. Anushka Rodi

**Abstract**:

This project introduces "Attendance Log API" a sophisticated cloud-based attendance management system that leverages the advanced capabilities of Microsoft Azure and the power of artificial intelligence. The system is designed to automate and streamline the process of attendance tracking in real-time.

At the heart of the system are two main components: a front-end user interface and a dedicated face-recognition module. The front-end, developed using React and hosted on an Azure Virtual Machine (VM), provides an interactive platform. It allows users to access live camera feeds for attendance capturing, manage attendance records, and perform administrative tasks such as adding or removing users.

The face-recognition module, also deployed on a separate Azure VM, is the core technological element of this system. It utilizes advanced face recognition algorithms to analyze camera feed frames, thereby identifying individuals in real time. This module is built using Python libraries such as Flask, OpenCV, and Dlib, alongside a PostgreSQL database for storing attendance logs and image data.

The system is designed with a focus on scalability, efficiency, and cost-effectiveness. The daily operational cost is optimized, making it an affordable solution for real-time attendance management. Moreover, future enhancements are planned to include a more structured and segregated architecture for improved scalability, a more intuitive user interface, and the integration of more accurate and lighter models for face recognition.

"Attendance Log API" represents a significant step forward in integrating cloud computing and AI technologies for practical and innovative solutions in attendance management. Its modular architecture and reliance on Azure's robust cloud services make it an adaptable and efficient tool for modern organizational needs.

**Architecture**:



**"Attendance Log API"** is an Infrastructure as a Service (IaaS) motivated project designed to create an efficient and secure attendance-log API. It employs a bifurcated architecture spread across a virtual network that segregates the system into public and private subnets, ensuring both accessibility and security.

**Architectural Components:**

1. **Internet Gateway:**
	* Serves as the entry point for all incoming traffic from various client devices, directing requests to the appropriate servers within the virtual network.
2. **Public Subnet:**
	* Hosts **Server 1 (Front-end)**, which runs the React application.
	* The React app provides the user interface for real-time interaction with the attendance system.
	* This subnet is accessible from the internet, allowing users to access the application via their devices.
3. **Private Subnet:**
	* Contains **Server 2 (Backend/API)**, the powerhouse of the system where sensitive operations occurs.
	* It includes the Face Recognition Model essential for the identification process in the attendance log system.
	* The PostgreSQL database resides here, managing and storing all attendance data securely.
4. **Routing Tables:**
	* Define rules that determine where network traffic from the subnet or gateway is directed.
	* In this architecture, routing tables ensure that traffic is properly routed between the public and private subnets and the internet gateway.

**Data Flow and Interconnectivity:**

* Traffic from client devices enters through the **Internet Gateway** and is directed to the **Public Subnet** where the React app's user interface allows for interaction.
* For operations requiring backend processing, such as face recognition, requests are routed internally to the **Private Subnet**.
* The Face Recognition Model processes the data, and the PostgreSQL database logs the attendance.
* The system's design ensures that the sensitive backend operations are not directly exposed to the public internet, enhancing security.

**Security and Scalability:**

* The division into public and private subnets enables a secure environment by segregating the user-facing front-end from the critical backend infrastructure.
* The virtual network facilitates scalability, allowing the addition of resources within the subnets to accommodate increased load or enhanced functionality.

This architecture represents a well-structured approach to cloud services, ensuring that the "Attendance Log API" system is robust, secure, and capable of scaling as per the demands of its users. By leveraging cloud infrastructure services, the system provides a seamless and secure experience for real-time attendance management.

**Features of "Attendance Log API":**

1. **Real-Time Video Feed Monitoring:**
	* The React-based front-end displays live video feeds from various classrooms or entry points.
	* This feature allows for the real-time tracking of arrivals, enabling immediate identification and attendance logging.
2. **Search Functionality:**
	* An intuitive search bar is available for administrators or users to quickly find employee attendance records.
	* It simplifies the management of large amounts of data by allowing for a quick lookup of specific individuals.
3. **Last Arrivals Display:**
	* The system provides a continually updated list of the latest arrivals.
	* This feature helps in monitoring punctuality and identifying attendance patterns.
4. **Administrative Control:**
	* An admin section is dedicated to user management tasks.
	* It includes functionality for adding new employees to the system, complete with the capability to upload images for the face recognition database.
5. **Employee Management:**
	* The admin can also delete employee records, maintaining the database's integrity and up-to-date status.
	* This helps in keeping the system efficient and the data relevant.
6. **Face Recognition Integration:**
	* At the backend, the face recognition model processes the video feed to identify individuals and log attendance.
	* The system uses advanced algorithms to ensure accurate and swift recognition.
7. **Database Interaction:**
	* All interactions from the front-end interface are stored and managed in a PostgreSQL database in the backend.
	* This robust database solution ensures secure and reliable data management.
8. **Responsive UI:**
	* The user interface is responsive, meaning it adjusts smoothly to different screen sizes for optimal viewing on various devices.
9. **Secure and Scalable:**
	* The architecture ensures that user interactions with the system are secure.
	* The cloud-based setup offers scalability to handle increased load as the user base grows.

These features highlight "Attendance Log API" as a modern, efficient, and user-friendly system for managing attendance through automated and real-time processes, with a strong emphasis on security and scalability.

**How It Works?**

The "Attendance Log API" automates attendance tracking by capturing live video feeds, essential for identifying individuals in real time. Users interact with the system via a React-based interface, which allows for live monitoring and efficient management of attendance records.

Central to the system is the Face Recognition Module, which processes video streams using advanced algorithms to match individuals against a database of profiles. Successful identification triggers an instant update in the "Last Arrivals" list on the user interface, confirming the attendance entry.

Administrative features enable the addition and removal of employees, with the system automatically retraining its recognition model to incorporate these changes, ensuring up-to-date accuracy.

This dynamic system securely manages data within a PostgreSQL database and operates within Microsoft Azure's scalable cloud infrastructure. The "Attendance Log API" represents a seamless, secure, and adaptable solution for real-time attendance management.

Top of Form

Bottom of Form



**Try it Yourself:**

To access the Attendance Log API:

1. You can go to <http://www.tirthgshah.com/>
2. There the browser will ask for permission to access your camera. Allow the permission.
3. Now, try adding a use from Add user option under Admin Section, once you add it, you can see that the video feed recognizes you and starts logging you.
4. Click refresh in Last Arrivals section and you can see that there is a log with your details.
5. Under Search for an employee, you can type your name and search for all your logs.
6. Once done, you can also delete your details from Admin Section under Delete and Employee.

You can get the files from the GitHub at: [Link](https://github.com/tirth2212/CloudComputing_AttendanceLog_API)