SMART CAMPUS: SMART ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION

Anushka Kalra[1], Reva Teotia[2], Sanskriti Mamgain[3], Abhishek Singh[4]

[1]Student, Dept. of Computer Science and Engineering, Inderprastha Engineering College, Uttar Pradesh, India
[2]Student, Dept. of Computer Science and Engineering, Inderprastha Engineering College, Uttar Pradesh, India
[3]Student, Dept. of Computer Science and Engineering, Inderprastha Engineering College, Uttar Pradesh, India
[4]Student, Dept. of Computer Science and Engineering(AIML), Inderprastha Engineering College, Uttar Pradesh, India

---

Abstract - *: This research introduces a revolutionary way of attendance management based on facial recognition technology. The solution uses Amazon Rekognition, a highly accurate and scalable service, to automate attendance recording in real-time. Faces are spotted and recognized from live images, removing the need for manual attendance recording. This abstract outlines the project's core characteristics, including precise facial recognition, real-time attendance tracking, scalability, security, and increased efficiency. The project highlights the efficiency of facial recognition technology in attendance management, providing a strong and dependable solution for a variety of institutions and organizations.*

***Key Words*: Attendance, Amazon Rekognition, Amazon Web Services, facial recognition, face**

1.INTRODUCTION

Attendance management continues to be a critical component of corporate and educational environments, assuring responsibility and efficiency. Traditional solutions, such as manual recording or swipe cards, have proven costly and susceptible to errors. In response to these issues, present-day technology presents a viable solution: facial recognition.

The evolution of attendance systems has seen a shift towards automation, with AI (Artificial Intelligence) and facial recognition technology leading the charge. Facial recognition, enabled by modern algorithms and machine learning, has transformed how we track attendance. By analyzing unique facial features, such systems can accurately identify individuals in real time, eliminating the need for manual input and reducing the risk of errors and fraudulent practices such as proxy attendance.

The project "Smart Attendance System Using Face Recognition" proposes a ground-breaking approach for streamlining attendance monitoring operations by incorporating facial recognition technology. This system, which has been developed using Amazon Rekognition, a cutting-edge AI-powered picture and video analysis tool, has the potential to revolutionize attendance management.

Attendance systems have evolved dramatically, moving from manual to automated solutions. With breakthroughs in Artificial Intelligence and facial recognition technologies, attendance monitoring has reached new levels of efficiency and precision. AI-powered technologies, such as Amazon Rekognition, have emerged as game changers, providing unprecedented capabilities in facial recognition and biometric identification.

By harnessing the power of AI and facial recognition technology, the Smart Attendance System aims to automate the attendance tracking process, eliminating the need for manual intervention and reducing administrative burden. This project represents a significant leap forward in attendance management, promising real-time, accurate, and efficient tracking across diverse organizational settings.

In summary, this project represents a pioneering effort in the field of attendance management, leveraging the latest advancements in AI and facial recognition technology. Through the integration of Amazon Rekognition, the system endeavours to revolutionize traditional attendance tracking methodologies, offering a more efficient, accurate, and reliable solution for various organizational settings.

2.PROJECT REQUIREMENTS

The automatic attendance marking system makes tracking attendance easier by utilizing facial recognition technology. It makes use of several AWS (Amazon Web Service) services, such as DynamoDB for database management, Simple Storage Service (S3) for storage, and Amazon Rekognition for facial analysis. Flutter and React are used in front-end development to provide an intuitive user interface for system interaction.

2.1 Technologies Involved

To build this system, the following technologies have been used:

2.1.1 Amazon Web Services Technologies

Amazon Rekognition: Amazon Rekognition is a cutting edge service that analyzes images and videos using deep learning. It has features such as facial recognition, object detection, and more. Amazon Rekognition is critical to this project's ability to automatically mark attendance. It accomplishes this by recognizing and evaluating faces in real-time and then comparing them to a pre-defined set of faces stored in a collection. The program also returns crucial facial traits that aid in identity confirmation and, as a result, promote proper attendance marking.

Amazon S3: Amazon S3 is a dependable and highly scalable object storage solution. Amazon S3 is used in this project for a variety of reasons. It provides a centralized repository for the project's facial recognition requirements by storing student photos and facial data for later use. It also aids in the management of these image sets, which are essential to the facial recognition procedure. The data needed for the automated attendance marking system is continuous and secure because of Amazon S3's redundancy and durability, which guarantee that all stored data is safeguarded against loss.

Amazon DynamoDB: A fully-managed No Structured Query Language (NoSQL) database service designed for scalability and high availability is Amazon DynamoDB. DynamoDB is essential to our project since it supports the automated attendance marking system. It forms the foundation of the system's data structure by storing vital student data such as names, student identifications (IDs), and face IDs. Additionally, the service keeps track of attendance logs, recording pertinent data such as student IDs and timestamps. Front-end apps can retrieve data quickly because of the data structure and strong storage, which makes it easier to integrate and interact with the data in real time. DynamoDB's scalability and high availability enable the system to manage growing loads and guarantee steady performance.

Amazon Application Programming Interface (API) Gateway: Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With API Gateway, developers can effortlessly craft Representational State Transfer (RESTful) APIs and WebSocket APIs, enabling the development of real-time, bidirectional communication applications. API Gateway supports containerized and serverless workloads, as well as web applications. Its benefits include efficient API development, performance at any scale, cost savings at scale, easy monitoring, flexible security controls and RESTful API options.

Amazon CloudWatch: Amazon CloudWatch is a service that monitors applications, responds to performance changes, optimizes resource utilization, and offers operational health information. CloudWatch provides visibility into system-wide performance by gathering data across AWS services. It also allows users to trigger alarms, automatically respond to changes, and receive a consolidated picture of operational health.

Amazon Lambda function: AWS Lambda is an event-driven, serverless computing platform provided by Amazon as part of Amazon Web Services. It lets developers run code without having to set up or manage servers. It executes code in response to events and automatically manages the computer resources needed for the function. It supports many programming languages and may be activated by a variety of AWS services or Hypertext Transfer Protocol (HTTP) requests, making it perfect for developing event-driven and microservices systems.

Amazon Amplify: AWS Amplify is a set of tools and services that make it simple for developers to create and launch apps on AWS. It includes code libraries, ready-to-use components, and a built-in Command Line Interface (CLI). AWS Amplify features a diverse set of open-source libraries and drag-and-drop User Interface (UI) components that developers can utilize as building blocks for their apps. The framework is compatible with iOS, Android, Web, and React Native mobile apps, as well as React, Ionic, Angular, and vue.js web applications.

2.1.2 Software Requirements

Flutter: Flutter UI framework is an open-source software development kit created by Google to facilitate the development of cross-platform applications. This project uses Flutter to create the mobile application that will be used by administrators and students to access and manage the attendance system. The framework's user-friendly interface makes it simple for users to check attendance records and related data. Flutter's ability to support numerous platforms—including iOS and Android—using a single codebase is a major advantage. This makes development and maintenance easier while guaranteeing a consistent user experience across various devices.

React: React is a JavaScript package used to create dynamic user interfaces, mostly for online applications. In this project, React is used to create a web-based front-end that is mostly used for administrative duties and monitoring. React's powerful capabilities enable real-time data updates from back-end services, allowing administrators to stay up to current on attendance status and other key information without having to manually reload the page. In addition,

React provides an interactive user experience, making attendance system management easier and allowing administrators to work more efficiently. This mix of real-time interactivity and ease of use makes React an excellent choice for developing the project's front-end components.

2.2 Hardware Requirements

Although this project's primary focus is on software and cloud-based services, certain hardware components are required for its operation. Cameras are required to take photographs of students, which are then analyzed by Amazon Rekognition for facial recognition and attendance monitoring. Users must also have a smartphone or tablet to engage with the Flutter-based software, which allows them to monitor attendance data and control relevant information. Administrators require Personal Computers (PCs) to access the React-based web application, which allows them to manage the system, track attendance, and conduct other administrative activities. These physical components work together with the software and cloud infrastructure to provide a streamlined and functioning attendance marking system.

2.3 Other Requirements

Security and Privacy: Since this project involves facial recognition and the handling of sensitive personal data, implementing robust security and privacy measures is crucial. All data stored in Amazon S3 and DynamoDB must be encrypted to ensure confidentiality and prevent unauthorized access. Additionally, the project must implement strict authentication and authorization protocols for users accessing the mobile app and the web-based application, ensuring that only authorized personnel can interact with sensitive data. Compliance with relevant data protection laws and regulations, such as the General Data Protection Regulation (GDPR) or Central Consumer Protection Authority (CCPA), is also essential to protect user privacy and maintain ethical standards. These security and privacy measures create a secure framework for the project, safeguarding user data while allowing the attendance system to function effectively.

3.DESIGN AND IMPLEMENTATION

3.1 Proposed Methodology

1. Requirement Analysis: Conduct a comprehensive analysis of the requirements and objectives of the Smart Attendance System using Face Recognition project and identify key functionalities, user roles, technologies, integration points with AWS services and frontend frameworks along with the deadlines.

2. Technology selection: Choose appropriate technologies for frontend and backend development after careful assessment of viable options. AWS services were preferred for the backend due to their accuracy, security and privacy. For the front-end, Flutter and React were chosen for the mobile application and web application respectively.

3. System Architecture and Design: Draft a general architecture of the system based on the objectives of the project in such a way that it defines the interaction between AWS services, frontend frameworks, and the database. Design Unified Modeling Language (UML) diagrams to understand user flow and data flow.

4. AWS Setup and Configuration: The project requires AWS services for implementation. Therefore, the creation of an AWS account, and setting up of IAM users, roles and permissions is the primary step. Post the setup of the account, it is essential to install and set up AWS CLI and AWS SDK (Software Development Kit) as per application requirements.

5. Setup AWS Resources and Services: Set up and configure AWS services, including creating Lambda functions for backend logic, setting up S3 buckets for storage, configuring DynamoDB tables for data persistence, implementing API Gateway endpoints for communication, and integrating CloudWatch for monitoring. Integrate Amazon Rekognition for facial recognition capabilities.

6. Frontend Development: Develop the user interfaces for the mobile and web applications using Flutter and React, respectively. Implement features for attendance tracking, student assistance resources, and user authentication.

7. API Development: Design and implement HTTP APIs using API Gateway to enable communication between the frontend applications and backend services.

8. Testing: Conduct thorough testing of the system to ensure functionality, performance, and security.

9. Deployment and Monitoring**:** Deploy the system to AWS infrastructure using Amplify for frontend deployment and AWS CLI for backend deployment. Set up monitoring and alerting using CloudWatch to track system performance and respond to any anomalies.

10. Documentation: Create comprehensive documentation including reports, review paper and research paper that outline the project.

3.2 General Architecture

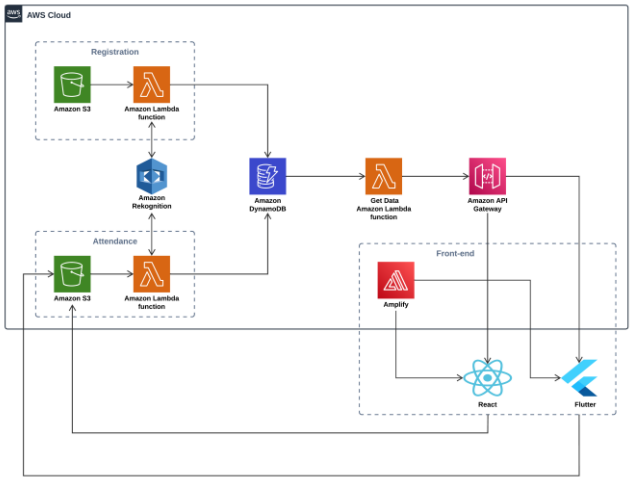

Fig -1: Architecture of Facial Recognition-based Attendance System

Figure 1 demonstrates the architecture used to set up AWS (Amazon Web Services) tools and services to build the facial attendance system. The AWS services include S3 bucket, Lambda function, Amazon Rekognition, Amplify, DynamoDB and API Gateway. The S3 buckets are used to store images, the lambda function includes the code to retrieve and process the data. The DynamoDB stores student data along with the attendance records that are later retrieved by the frontend technologies via API gateway.

4.RESULTS

4.1 Brief Result

The project successfully performs automated attendance marking using face recognition technology. Each student, upon being recognized accurately, is marked present and the attendance timestamp is recorded to verify the data. The confidence score for the recognized faces is above 95 which can be verified using the Amazon CloudWatch logs of the corresponding Lambda function. The application also incorporates comprehensive student assistance resources, extending support to the students beyond campus.

4.2 Mobile Application Screens

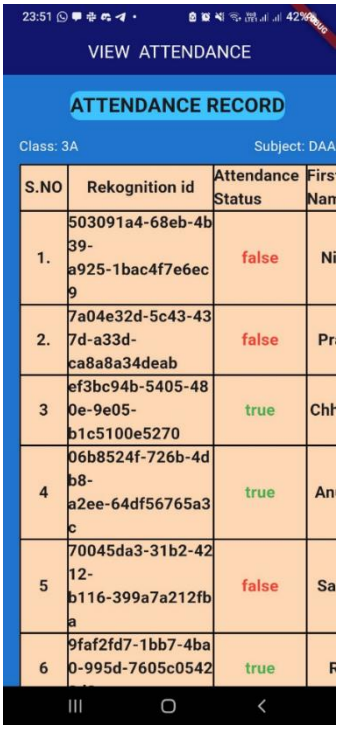

Fig -2: View Attendance Screen 1

Fig -3: View Attendance Screen 2

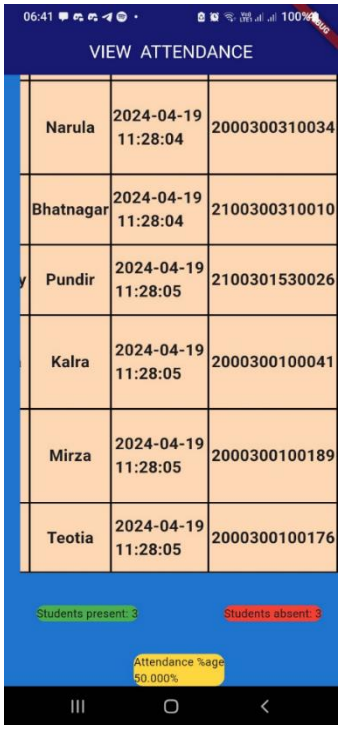

Fig -4: View Attendance Screen 3

Figures 2, 3 and 4 are horizontally scrollable and depict the attendance records of the students in the form of a table.

4.3 Web Application Screens

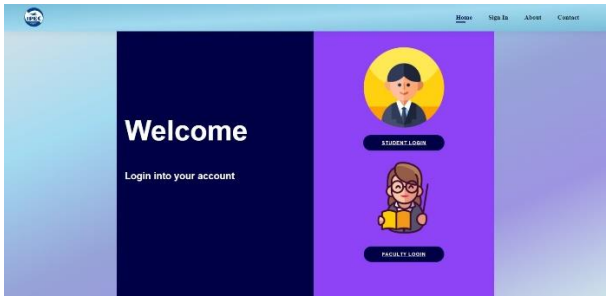

Fig -5: Home Screen

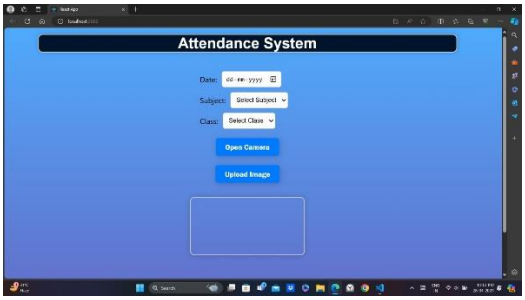

Fig -6: Mark Attendance Screen

Figures 5 and 6 depict some of the screens from web application built using React.

5. CONCLUSION

In conclusion, the development of the "Smart Attendance System using Face Recognition" project marks a significant milestone in the evolution of attendance tracking methodologies, particularly within educational and organizational contexts. The project has successfully developed a strong and all-inclusive attendance management solution by utilizing advanced technologies such as Amazon Rekognition and numerous AWS services, including Lambda, S3, DynamoDB, Amplify, CloudWatch, and API Gateway, in conjunction with frontend frameworks like Flutter and React.

The integration of facial recognition technology enables real-time and accurate identification of individuals, eliminating the need for manual attendance processes and reducing administrative burden. Utilizing AWS services also guarantees the system's scalability, dependability, and security, enabling seamless operation in a variety of organizational settings.

Moreover, the inclusion of comprehensive student assistance resources extends the functionality of the application beyond attendance tracking, providing valuable support and resources to students both on and off campus. This feature emphasizes how committed the initiative is to improving student success and the educational process as a whole.

To put it simply, the "Smart Attendance System using Face Recognition" project is an attempt to solve the drawbacks of conventional attendance monitoring techniques by combining cutting-edge technologies with user-centric design principles.

6. FUTURE SCOPE

1. **Geolocation Integration:** Integrating geolocation features into the attendance system can enable administrators to track the location of students or employees when they mark their attendance. This can help prevent attendance fraud by verifying that individuals are physically present at the designated location during attendance marking.

2. **Cross-Institution Integration:** Cross-institution integration allows multiple educational institutions or organizations to share attendance data securely and efficiently. This can be particularly useful for students or employees who belong to different institutions or departments and need to maintain a consolidated attendance record.

3. Feedback Mechanism: Implementing a feedback mechanism within the attendance system allows students or employees to provide feedback on their attendance experience. This feedback can be used to identify areas for improvement, address concerns, and enhance user satisfaction with the attendance marking process.

4. Emotion Recognition: Integrating emotion recognition technology into the attendance system can provide valuable insights into the emotional state of students or employees during attendance marking. This information can be used to identify potential issues affecting attendance and students in general, such as stress, anxiety, or disengagement, and take appropriate actions to address them.

REFERENCES

[1] Shyam Sunder Bahety, Kishan Kumar, Vishwadeep Tejaswi, Sharad R Balagar, Anil B C, 2020, Implementation of Automated Attendance System using Facial Identification from Deep Learning Convolutional Neural Networks, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15)

[2] Fenton, A., 2019. Automated Classroom Attendance System (ACAS).

[3] T, Shalini, FACE RECOGNITION BASED ATTENDANCE SYSTEM USING CNN ARCHITECTURE (May 13, 2023). International Journal for Innovative Engineering & Management Research, Forthcoming, Available at SSRN: https://ssrn.com/abstract=4447364

[4] Miller, W.E., 2018. Utilizing Facial Recognition Software to Record Classroom Attendance (Doctoral dissertation, Auburn University).

[5] "API Management - Amazon API Gateway - AWS," Amazon Web Services, Inc. https://aws.amazon.com/api-gateway/

[6] "Serverless Function, FAAS Serverless - AWS Lambda - AWS," Amazon Web Services, Inc. https://aws.amazon.com/lambda/

[7] "Introduction to Amazon CloudWatch," Amazon Web Services, Inc. [Online]. Available: https://aws.amazon.com/cloudwatch/

[8] Wikipedia contributors, "AWS Lambda," Wikipedia, Apr. 17, 2024. https://en.wikipedia.org/wiki/AWS_Lambda

[9] B. Matador, "What is AWS Amplify?," Blue Matador. https://www.bluematador.com/blog/what-is-aws-amplify

[10] "Cloud Object Storage - Amazon S3 - AWS," Amazon Web Services, Inc. https://aws.amazon.com/s3/

[11] "What is Amazon Rekognition? - Amazon Rekognition." https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html

[12] "Build your own face recognition service using Amazon Rekognition | Amazon Web Services," Amazon Web Services, Jun. 24, 2021. https://aws.amazon.com/blogs/machine-learning/build-your-own-face-recognition-service-using-amazon-rekognition/

[13] Ernchi, "GitHub - ernchi/aws-rekog-attendance-sys: Facial recognition based attendance taking system using AWS Rekognition.," GitHub. https://github.com/ernchi/aws-rekog-attendance-sys

[14] "Detecting and analyzing faces - Amazon Rekognition." https://docs.aws.amazon.com/rekognition/latest/dg/faces.html

[15] Ayu, "Face recognition assessment (AWS Face Rekognition) - DSF Web Services Engineering - Medium," Medium, Jan. 07, 2022. [Online]. Available: https://medium.com/dsf-developers/face-recognition-assessment-aws-face-rekognition-15921d65b351

[16] "Fast NoSQL Key-Value Database – Amazon DynamoDB – Amazon Web Services," Amazon Web Services, Inc. https://aws.amazon.com/dynamodb/

[17] C. Wen, "Serverless Deployment with Netlify Lambda (ReactJS FE + GraphQL BE)," Medium, Jan. 26, 2024. [Online]. Available: https://medium.com/@cheahwen1997/implement-a-face-recognition-attendance-system-with-face-api-js-946a1615236d

[18] M. Grover, "Face Recognition using Amazon Rekognition - Malvika Grover - Medium," Medium, Oct. 07, 2023. [Online]. Available: https://medium.com/@grover.malvika1/face-recognition-using-amazon-rekognition-42fb5fe27fee

[19] "Module Three - Build a Flutter Mobile App using AWS Amplify - Part 1," Amazon Web Services, Inc. https://aws.amazon.com/getting-started/hands-on/build-flutter-mobile-app-part-one/module-three/

[20] MadMario, "Deploy lambda using serverless," Flattered With Flutter, Apr. 05, 2022. https://flatteredwithflutter.com/deploy-lambda-using-serverless/

[21] MadMario, "Dart on AWS lambda," Flattered With Flutter, Nov. 03, 2021. https://flatteredwithflutter.com/dart-on-aws-lambda/

[22] Aws-Samples, "GitHub - aws-samples/amazon-rekognition-code-samples: Amazon Rekognition Code Samples," GitHub. https://github.com/aws-samples/amazon-rekognition-code-samples

[23] S. Purkayastha, "Machine Learning Use Case: Facial Recognition Using Amazon Rekognition – RadioStudio," RadioStudio, Jan. 02, 2024. https://radiostud.io/machine-learning-use-case-facial-recognition-using-amazon-rekognition/

[24] G.S SanthoshKumar, Martin Siby, Philson K Philip, Rajat Jaic Mendus, Shine Abraham. 2019. Face-track : Smart Attendance System using Face Recognition. International Research Journal of Engineering and Technology (IRJET). 06(05).