STUDENT ATTENDANCE SYSTEM USING AWS

I. PROJECT IDEA

The Student Attendance System facilitates the monitoring of the attendance of students enrolled in a particular class. The system provides an efficient and online solution to the process of recording attendance, thereby reducing the need for manual labor and minimizing the errors generated during manual processes. The system also generates visualizations that help in understanding trends in students' attendance over the semester.

Student Attendance will be based on the department and section. According to the department wise and section-wise, the attendance will be marked for the students. It includes present, absent and leaves a column for each student so that they would mark the attendance like period wise.

This system is based on Amazon AWS and implements the following services - Amazon S3, Amazon DynamoDB, AWS Lambda, Amazon Rekognition and Amazon API Gateway. The interactions among these services are explained by means of the architecture diagram.

II. DELIVERABLES

An automated attendance system deployed in a web/mobile app based on high accuracy face recognition technology. Following are the features of this application which will be available to professors:

1. Recording and detecting which 'enrolled' students are actually present in the class by matching the face of each student entering the class against existing photos from the database of enrolled students. Our system will perform this task two times - the start and the end of the class.
2. Generating a list of absent students and notifying the professor at the end of every class.
3. Facilitating the professor to perform a 'final approve' on the list of absent students generated by the system. This has been done keeping the following scenarios in mind:
    a. A student may have left in between the class or a student may have entered late due to a genuine reason (we assume here that this reason has been communicated to the professor). The student will be marked absent by the system as their face would have got successfully detected only in one round of attendance and not both.
    b. Being realistic, we are assuming that there is a very small chance of an error occuring wherein the system might not successfully match the face of a person who is actually enrolled and is present in the class. In this case, the screen will display a message

immediately, signaling the student to inform the professor of this error and verify their enrollment in the class.

4. Generating monthly and weekly visualizations enabling the professor to observe the attendance data and take actions accordingly.

5. Detecting people who are not a part of the university and are present in the class. These people should not be allowed to enter.

III. IMPLEMENTATION

- **API Gateway**

  /enrolled-students

  1. GET
     a. Integration type : Lambda
     b. To retrieve all the enrolled students

  /temp-attendance

  1. GET
     a. Integration type : Lambda
     b. Retrieves the absentees list detected by Rekognition

  2. POST
     a. Integration type : Lambda
     b. Approves the newly generated attendance

  /upload-image /{bucket} /{key}

  1. PUT
     a. Integration Type : S3
     b. Uploads the class image to S3 bucket (using which attendance is to be marked)

- **S3**

  Two S3 buckets are created namely “project-frontend” and “project-backend”. The frontend bucket is where the website and all the related web pages are hosted. The backend is for storing the student profile photos and classroom images for facial recognition processing.

- **Rekognition**

  We use `detect_faces` rekognition API to get face boundary of each detected face in the uploaded image. We then use these face boundaries to crop faces from the uploaded image. We pass each cropped face to `search_faces_by_image` rekognition API which searches a given face in enrolled students collection. We have saved photos of enrolled students in this collection. If there is a face match, we mark that student present in the DynamoDB attendance table.

- **Lambda**

  Multiple lambda functions have been employed in this project in order to carry out all the functionality that is required i.e Getting students, recognizing faces, marking attendance, modifying attendance etc.

  1. Lambda0 - Index photos (adding to collection)
  2. Lambda1 - Called when upload is called (triggered). Call Rekognition on uploaded photos.
  3. Lambda 3 - Display enrolled students (from dynamoDB)
  4. Lambda 6 - approve and submit today's attendance after all the changes required

- **DynamoDB**

  DynamoDB is used as the backend database for the web application and for storing the final student attendance results.

  project_attendance table : Temporary attendance is saved in this table when the approval by professor is pending.

  permanent-attendance table : records are added to this table when the professor has approved the attendance.

IV. ARCHITECTURE

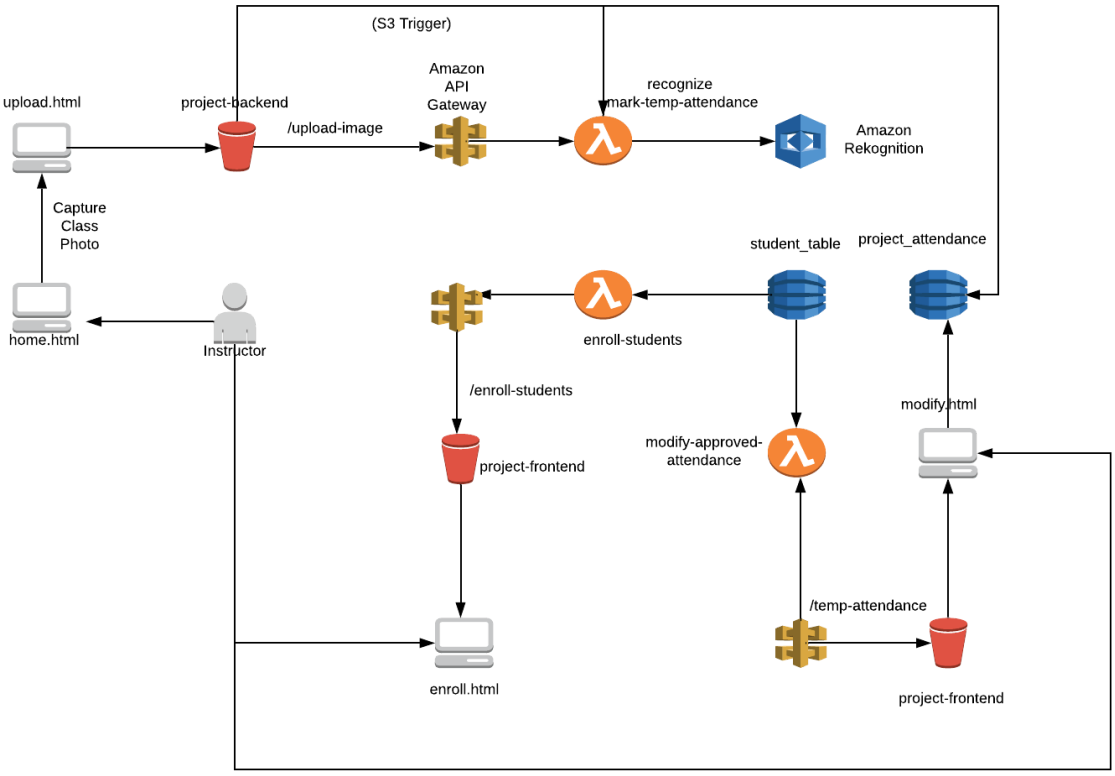

Figure 1: Architecture of Student Attendance System

V. SYSTEM FLOW

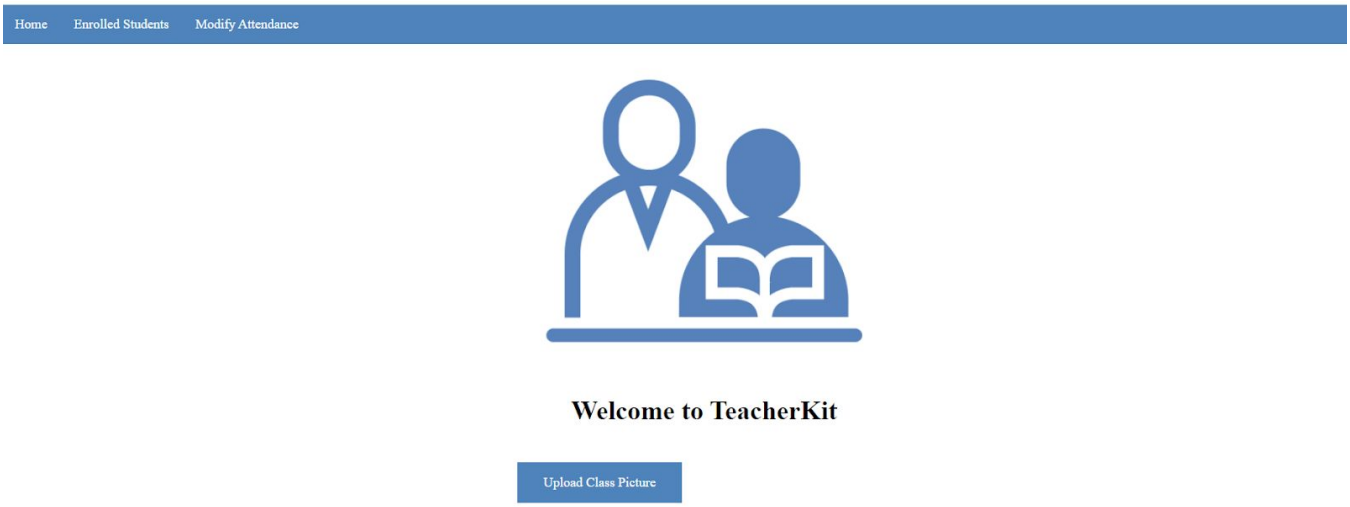

Figure 2: Home Page

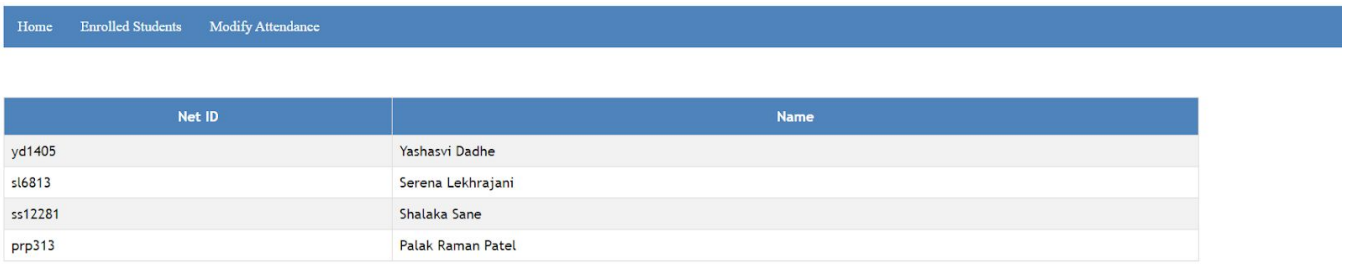

Figure 3:Enrolled Students Page

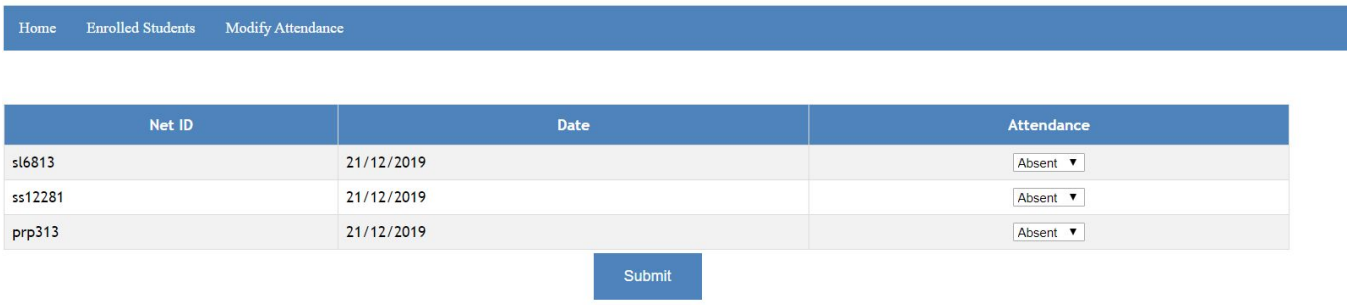

Figure 4:Modify Attendance Page