A Minor Project Mid-term Report on

Presence: Automated Attendance System

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Bachelor of Engineering in Software Engineering

under Pokhara University

Submitted by:

Rupesh Budhathoki, 191721

Apsara Bishwokarma, 191704

Sampada Kharel, 191723

Sudarshan Kshetri, 191729

Under the supervision of:

Yogesh Deo

Date:

01 July 2023

Department of Software Engineering

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

Abstract

The ”Presence” project is an innovative approach to attendance tracking in college classrooms. With the help of automation and data analysis, the system aims to provide a more efficient and accurate way of keeping track of student attendance while also providing valuable insights into student behavior and academic performance. The system uses cameras to detect students' arrival. The data collected is then analyzed to provide a real-time attendance report that instructors can access from their devices. This eliminates the need for manual attendance-taking and reduces the potential for errors or inaccuracies. In addition to real-time attendance tracking, the system also provides detailed reports on student attendance patterns, subject preferences, and behavior in class. By analyzing this data, instructors, and administrators can identify students who may be struggling with attendance or academic performance and take proactive steps to address these issues. They can also identify trends and patterns in attendance that can inform decisions about course scheduling and curriculum design.

The ”Presence” project is a valuable tool for both students and faculty. For students, it provides a streamlined attendance tracking process that eliminates the need for manual sign-ins and helps ensure that they are meeting attendance requirements. For faculty, it provides real-time insights into student attendance and behavior that can inform teaching strategies and improve academic outcomes.

Keywords– *automated attendance system, data analysis, college classrooms, attendance tracking, attendance report, academic performance, attendance patterns, subject preferences, student behavior, teaching strategies, curriculum design*

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation/Problem Statement

In today's technology-driven era, the trend is to switch from traditional systems to fast, smart, and interactive systems that can be linked to web applications, enabling users to access the system from anywhere at any time. In academic settings, regular attendance is crucial for students to acquire knowledge, participate in class discussions, and engage with their peers. However, traditional methods of taking attendance, such as calling out names or manually signing in, can be tedious, prone to errors, and inefficient.

The "Presence" project is being developed to address the challenges associated with manual attendance-taking in college classrooms. Traditional methods of attendance tracking, such as paper sign-ins or roll calls, are time-consuming, prone to errors, and may not provide accurate or timely data for instructors and administrators. Additionally, manual attendance-taking does not provide insights into student behavior or attendance patterns, which can be valuable for improving academic outcomes and making data-driven decisions.

1.2 Project Objectives

To address these challenges, we propose the design and implementation of a face detection and recognition system called "Presence." The "Presence" system is an automated attendance system that utilizes facial recognition technology to identify and mark the attendance of students attending a lecture in a classroom. By automating the attendance process, "Presence" aims to save time for instructors, reduce errors, and provide a more accurate and efficient way to monitor student attendance.

The proposed project has put forward the following objectives:

- To automate attendance system for college classrooms
- To provide insights into attendance patterns and behavior.
- To support data-driven decision-making for improved student success.

1.3 Significance of the study

The "Presence" project holds significant importance as it introduces an automated attendance system that addresses the limitations of manual methods. It streamlines attendance tracking, provides real-time reports, and offers insights into attendance patterns and student behavior. This empowers educators to make data-driven decisions, optimize instructional practices, and enhance student success in college classrooms.

2 Scope and Limitation

The scope of the ”Presence” project is wide-ranging and encompasses various aspects of attendance tracking and data analysis in college classrooms. It will be designed to accommodate multiple subjects and courses, providing flexibility for instructors and students. The system's scope also extends to generating comprehensive attendance reports and analysis, enabling instructors and administrators to gain valuable insights into attendance patterns, subject preferences, and student behavior. Additionally, the project has the potential for future expansion and integration with other educational technologies, further enhancing its scope and usability.

The following are the limitations of the project that are realized:

- The system may not work accurately in low lighting or poor camera quality environments.
- The system will have difficulty detecting faces if students are wearing masks or other facial coverings.
- Potential technical issues or errors with sensors and data analysis algorithms.
- Lack of consideration for the specific reasons behind a student's absence, such as temporary absences or emergencies.

3 Literature Review

This section consists description of the literature study performed during the development of this proposal.

3.1 Fareclock

Fareclock[1] is a web-based time clock software that offers features for employee time tracking, attendance management, and payroll processing. It provides businesses with tools to efficiently track employee work hours, monitor attendance, and generate accurate payroll reports. Fareclock allows employees to clock in and out using various methods such as biometric fingerprint scanning, web punch, mobile app, or phone call. The software also provides features like overtime tracking, PTO management, and shift scheduling. Fareclock aims to streamline the time management process and simplify payroll administration for businesses of all sizes.

3.2 Learning Management System (Moodle)

Moodle is a free and open-source learning management system. It has a variety of plugins to expand the features. There is a mod-attendance[2] plugin also called Attendance Activity[3] where the instructor clicks on the ”Update Attendance” button and is presented with a list of all the students in that course, along with configurable options and comments. The default options provided are Present, Absent, Late Excused. Instructors can download the attendance for their course in Excel format or text format. Sessions can also be configured to allow students to record their own attendance and a range of different reports are available.
Presence offers several advantages over the mod-attendance plugin in Moodle, making it a more beneficial solution for automated attendance tracking in college classrooms.

Firstly, Presence provides a seamless and automated attendance tracking process, eliminating the need for manual input by instructors. Unlike the mod-attendance plugin, which requires instructors to manually update attendance by selecting options for each student, Presence utilizes sensors or other automated methods to track student entry into the classroom. This not only saves instructors significant time and effort but also ensures more accurate and reliable attendance data.

Secondly, Presence offers advanced data analysis and reporting capabilities. While the mod-attendance plugin in Moodle allows instructors to generate basic attendance reports, Presence takes data analysis to a higher level. It compiles and analyzes attendance data, providing valuable insights into attendance patterns, subject preferences, and even student behavior. This level of analysis can help instructors and administrators make informed decisions regarding teaching strategies, course planning, and student support, ultimately improving overall educational outcomes.

However, since Moodle is open-source we can use this technology and integrate it with Moodle.

3.3 Existing Similar Appilcations

While there may be similar plugins or programs available that offer alternative approaches to attendance tracking, ”Presence” stands out by providing a comprehensive solution. It detects student arrivals, allowing for a more thorough analysis of attendance patterns. This valuable data can then be utilized to design improved timetables and curricula, ensuring optimal scheduling and enhancing the overall learning experience. By offering this unique functionality, Presence sets itself apart as a powerful tool for attendance management and curriculum optimization.

4 Methodology

This section describes the methodology that we have been following during the development of our project, which is the waterfall model.

4.1 Software Development Life Cycle

For the development of our project, we have chosen to follow the waterfall model[4] of the software development life cycle as depicted in Figure 1. The reason for choosing this model is the lack of sufficient time duration for agile and iterative methods, as well as very low chances of changes of requirements in the process of development.

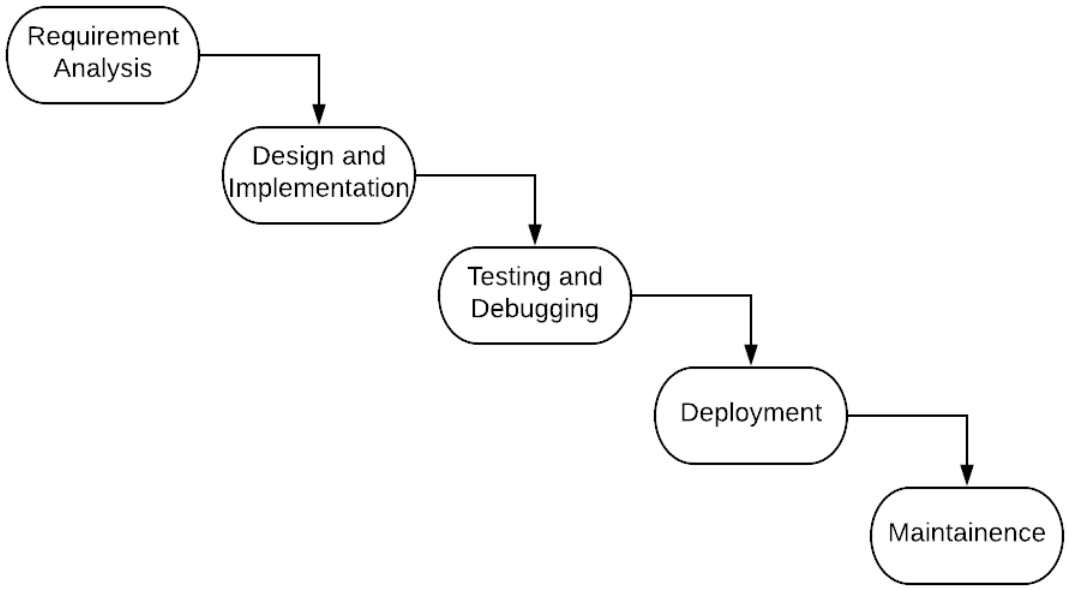

Figure 1: Software development life cycle

For the development of our project, we have adopted the waterfall model as our chosen software development life cycle. In this model, the project life cycle begins with the collection and evaluation of requirements for the application. We then proceed to the design and implementation phase, where we design and build both the API services and client applications. By the end of this phase, we aim to have a minimal viable product (MVP) constructed. In the testing and debugging phases, the quality control methods will be applied to both API and application. Finally, the application will be deployed to Docker containers at the end of the deployment phase. However, there might be slight modifications in the original waterfall model where the design and implementation may be changed slightly after the testing phase if seen as reasonable.

4.2 Technical Architecture

The application is being built upon the client-server web architecture, as illustrated in Figure 2.

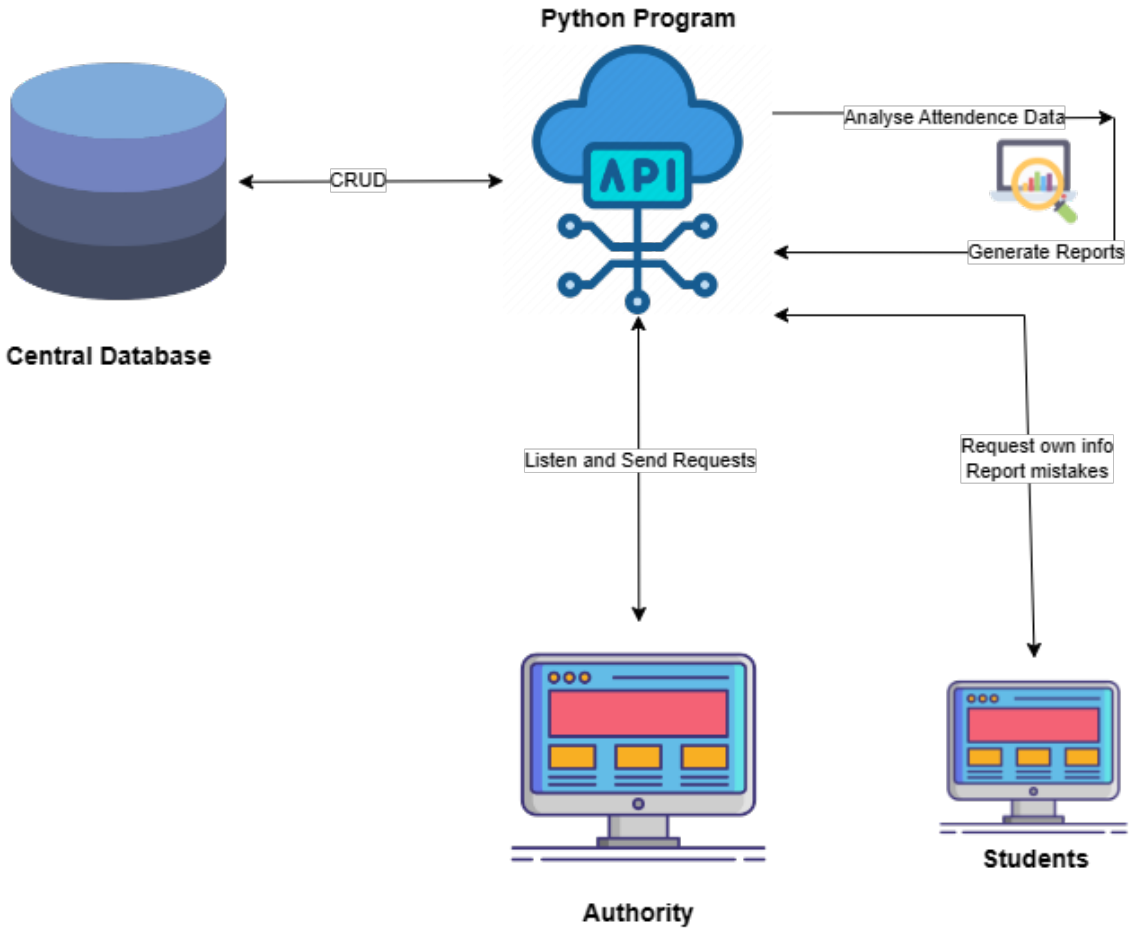

Figure 2: Proposed architecture of the application

The architecture consists of a database that handles data storage and management. Python is responsible for handling Create, Read, Update, and Delete (CRUD) operations on the database. It also provides WebSocket communication capabilities to the client application.

The client application allows students to access their personal attendance information securely. Only authorized personnel, such as administrators or instructors, can access the entire dataset via the client application using WebSocket communication.

Python program acts as the intermediary between the client application and is also responsible for generating reports based on the attendance data. This communication ensures seamless integration and efficient report generation.

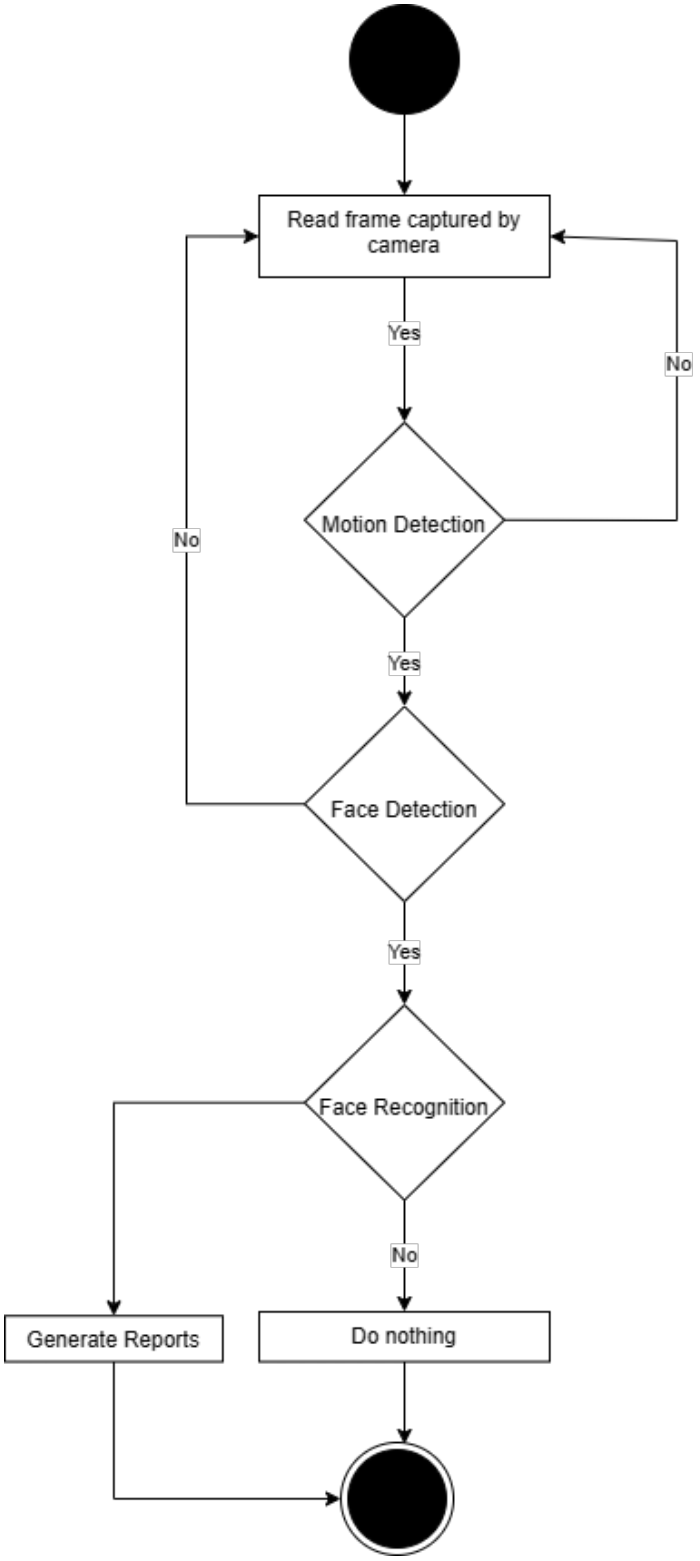

Figure 3: Activity Diagram[5] of the application

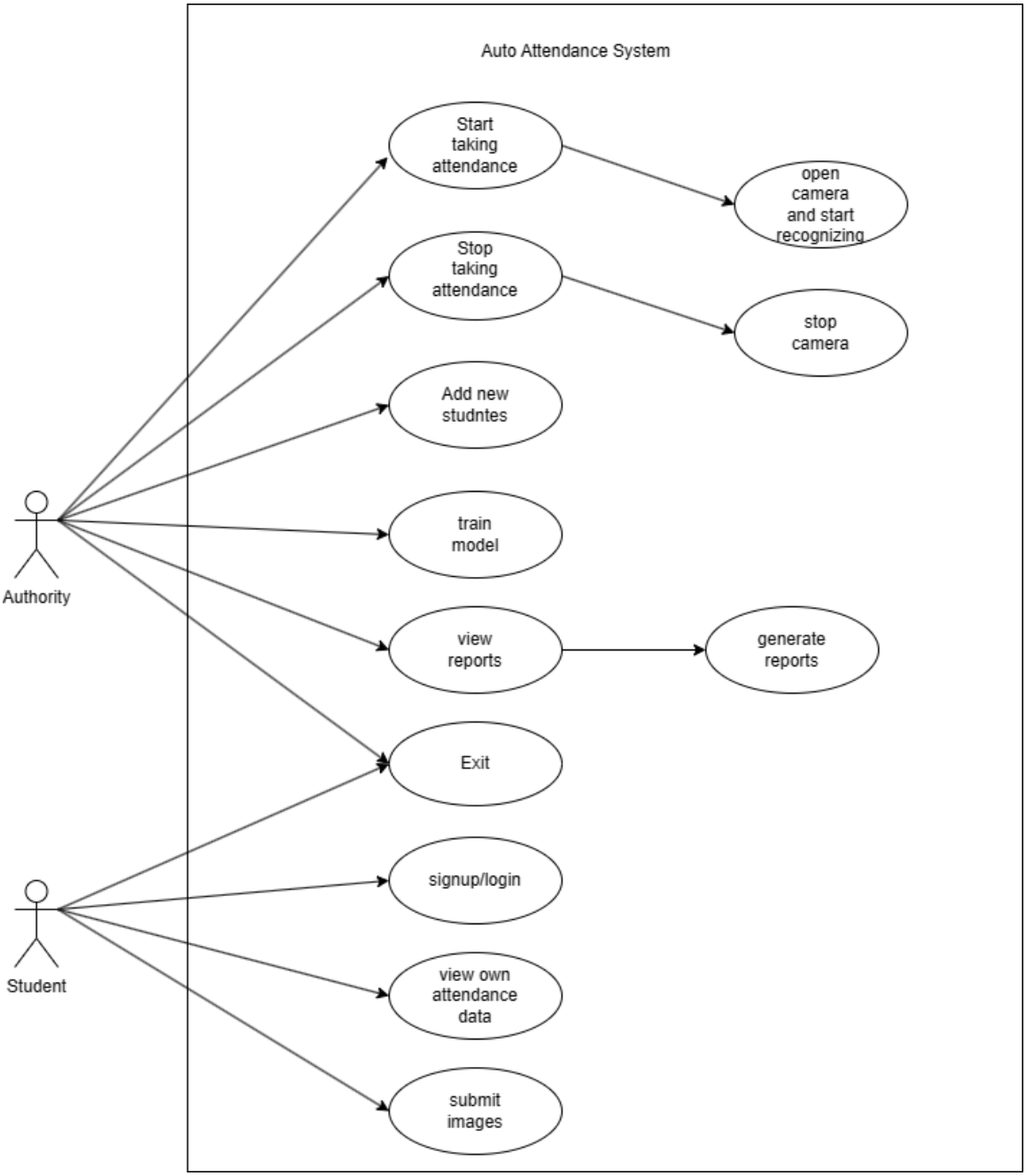

Figure 4: Use-case Diagram of the application

4.3 Technologies

Table 1 presents the major technologies that we are using for the development and deployment of our application.

Subject	**Technology**
Backend Database	SQLite
Backend Service	django, Python
API Communication	WebSocket
Frontend(Client)	Next.js(React), Typescript, TailwindCSS
Deployment	Docker Containers

Table 1: Technologies proposed to be used

4.4 Face detection algorithm

For the development of our project, we are utilizing the OpenCV library in Python to train our model and perform face detection for attendance purposes. In this regard, we have two options for face detection: HOG(Histogram of Oriented Gradients) or CNN.

The HOG face detector offers faster processing speed, which aligns well with our limited resources and time constraints. It operates by dividing an image into small cells, computes each cell's gradient orientation and magnitude, and then aggregates the gradient information into a histogram of oriented gradients. These histograms describe the image features and detect objects within an image. However, it may be less accurate when dealing with changes in the viewing angle or rotation of faces.

For more robust face detection, we can employ the CNN face detector. This approach requires more computational resources, resulting in slower processing times. Nonetheless, it offers higher accuracy and is more resilient to variations in face rotation and viewing angles.

Furthermore, if we have access to a GPU, we can leverage it to run the CNN face detector, enabling real-time face detection. Combining the CNN face detector with a GPU delivers the perfect combination of deep neural network accuracy and the efficiency of a less computationally demanding model.

We have planned to use HOG(Histogram of Oriented Gradients) because of low resources and less time.

Algorithm	**HOG**	**CNN**
Approach	Histogram of Oriented Gradients (HOG) feature extraction	Convolutional Neural Network (CNN)
Training Complexity	Less computationally intensive	More computationally intensive
Speed	Faster	Slower
Accuracy	Less accurate, especially with changes in viewing angles	More accurate and robust to face rotation
Face Rotation Tolerance	Low	High
GPU Support	Limited (may not utilize GPU)	Utilizes GPU for improved performance
Model Size	Smaller	Larger

Table 2: Comparison of HOG and CNN algorithms for face detection.

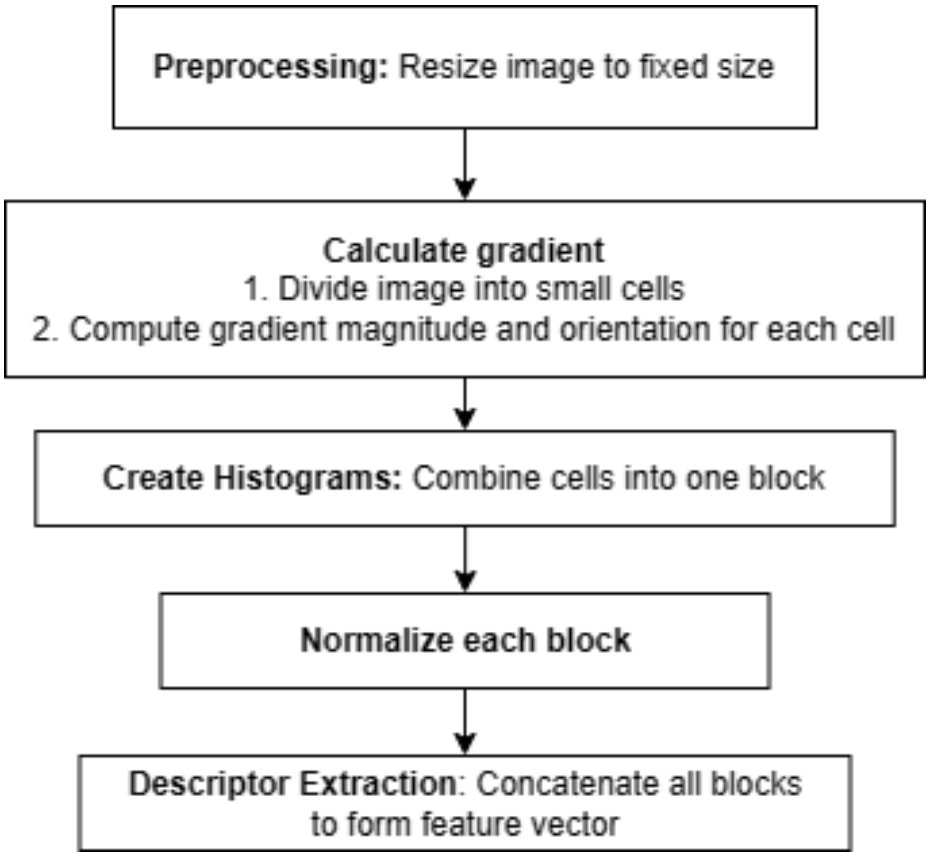

Figure 5: Hog process of feature extraction

5 Tasks Done so far

We understood the significance of time and aimed to make the most of our development period by maximizing our progress.Within this timeframe, we have made significant progress and successfully completed the following tasks:

- **Research and Requirements Gathering**: Conducted thorough requirement gathering to ensure alignment with project objectives and user needs.
- **Design and User Interface Development**: Successfully designed the system architecture and developed a user-friendly interface including login, sign up, forget password functionality, admin panel, and dashboard.
- **Face Detection and Recognition Technology**: Researched and implemented advanced face detection and recognition technology for accurate attendance tracking.
- **Dataset and Model Training**: Made a portal to gather dataset images from students and train the facial recognition model using OpenCV for accurate attendance tracking.
- **Database Design and Implementation**: We have successfully designed and implemented the necessary tables, relationships, and entities to store student information, attendance records, and other relevant data while prioritizing efficiency, scalability, and data integrity.
- **REST API and websockets**: We connected the backend and frontend using REST APIs for on-demand communication and WebSockets for real-time communication while securing every APIs.

By completing these major tasks, we are progressing towards achieving our project goals of developing an automated attendance system that effectively tracks and analyzes student attendance.

6 Result and Discussion

To ensure the security and accessibility of the system, we have implemented various authentication mechanisms, including:

- **Login**: Users can securely login using their credentials to access the system's features and functionalities.

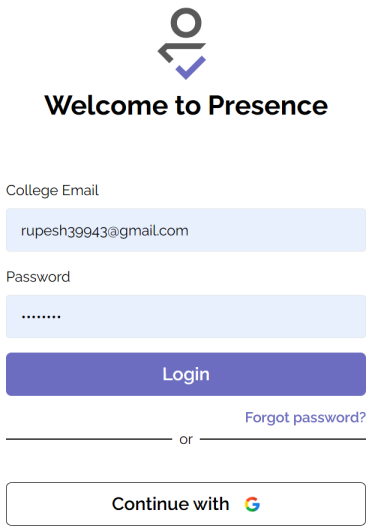

Figure 6: Login form for both users and admin

- **Login with Google**: Users have the option to authenticate themselves using their Google accounts, providing a convenient and streamlined login experience.

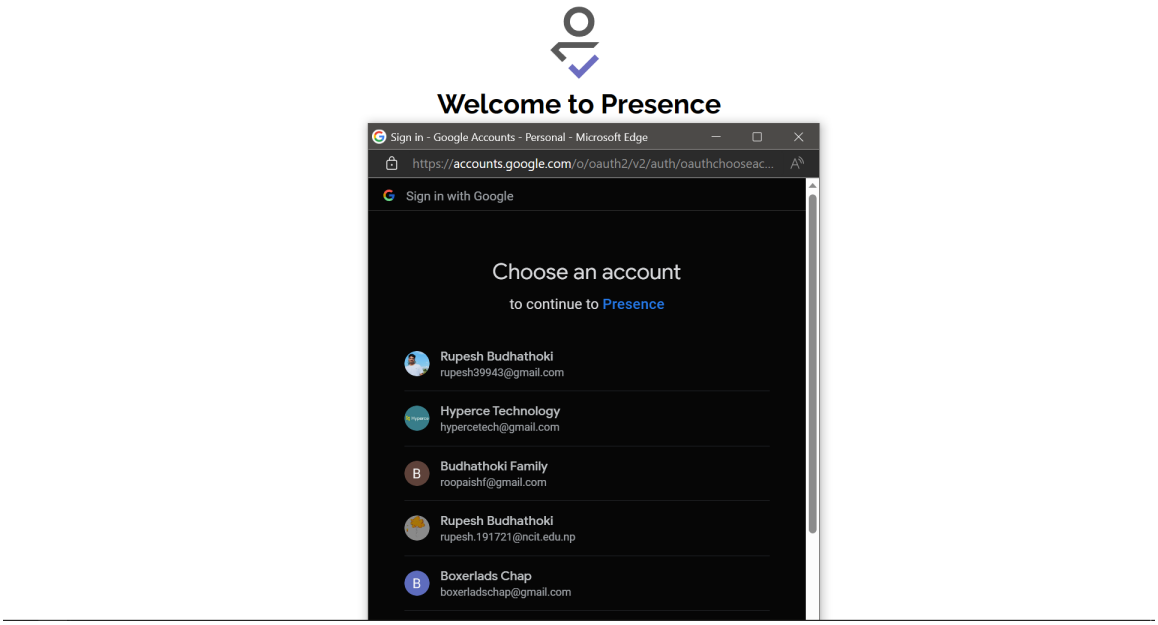

Figure 7: Login with Google functionality

- **Forgot Password**: In the event of a forgotten password, a password recovery mechanism allows users to reset their passwords securely.

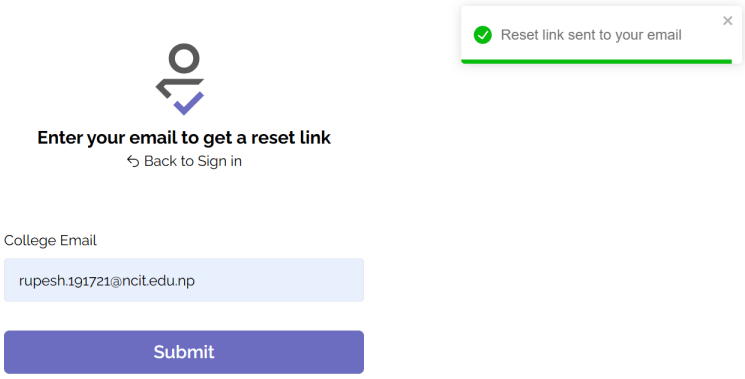

Figure 8: Login with google functionality

- **Reset Password**: Users can initiate the password reset process and set a new password to regain access

to their accounts.

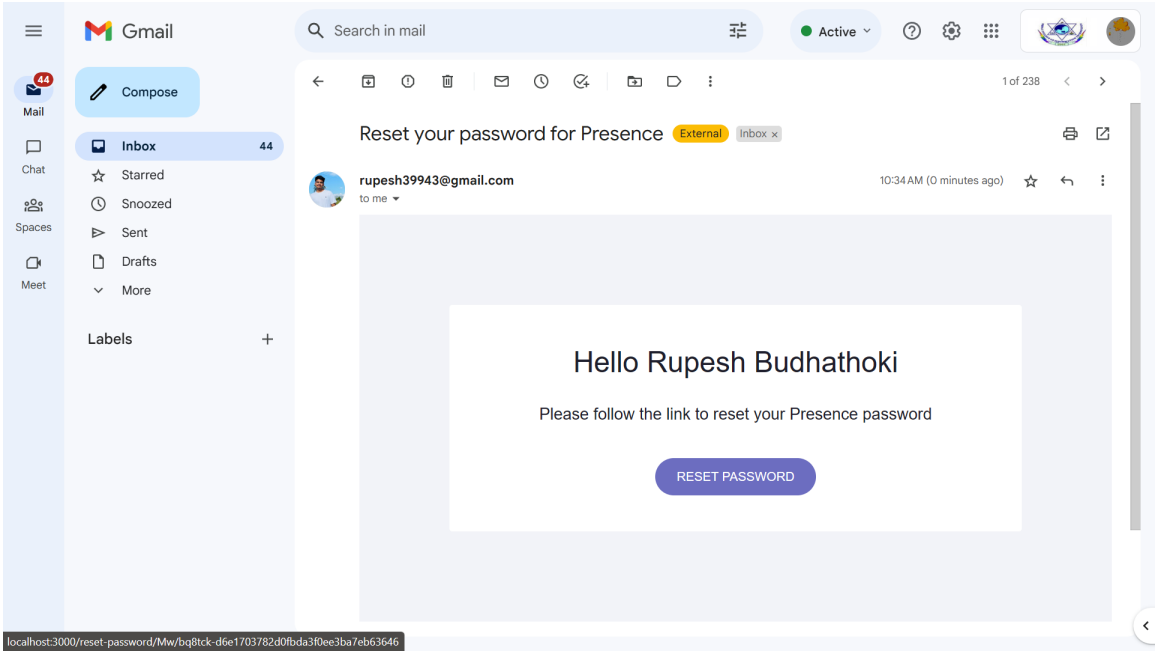

Figure 9: Reset Password Email

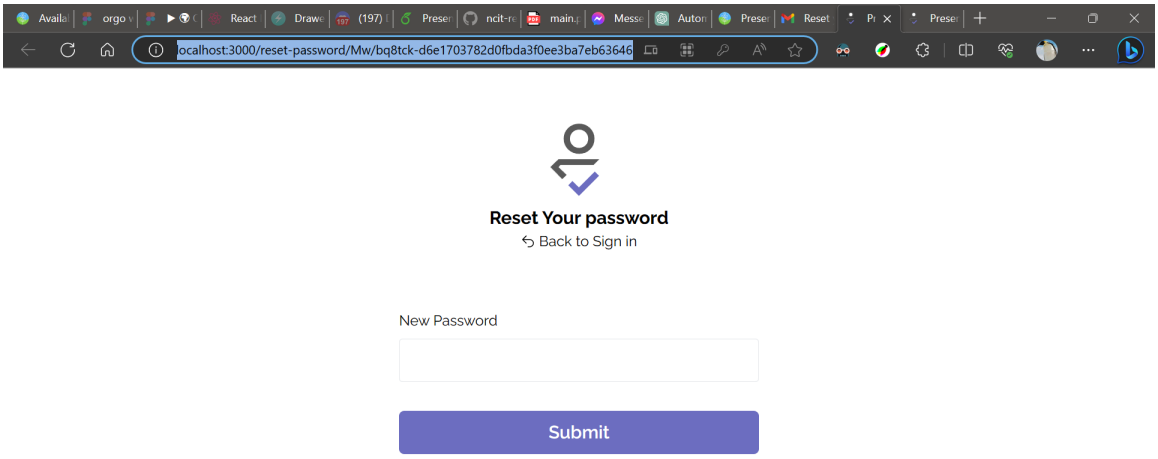

Figure 10: Reset password

Key Features for Students:

- **Image Submission**: Students can submit their images to be used for facial recognition and attendance

tracking.

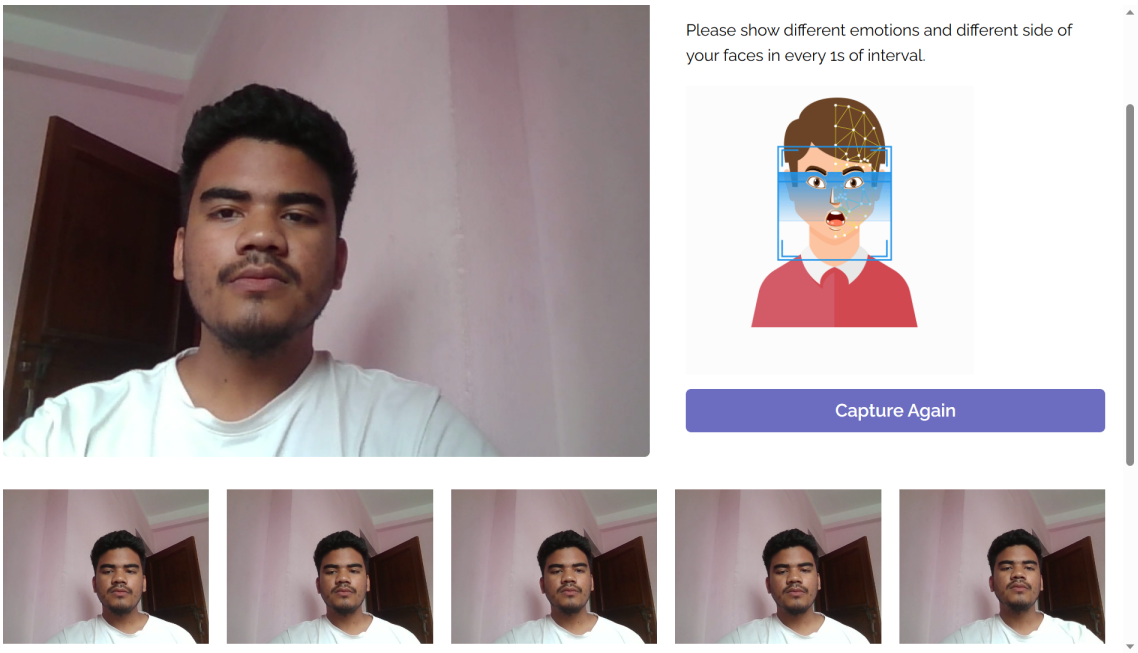

Figure 11: Image submission form

- **Attendance Tracking**: Students have access to view their attendance records, providing insights into their attendance patterns and history.

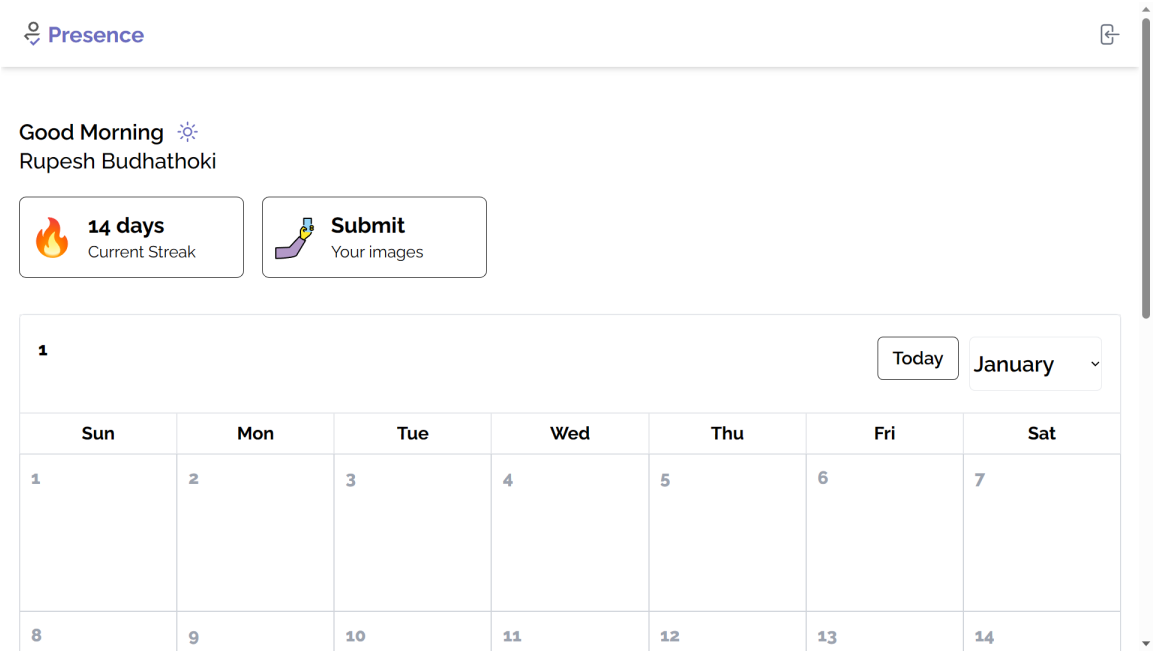

Figure 12: Student dashboard

Key Features for Authorities:

- **Student Management**: Authorities can add new students to the system, maintaining an up-to-date database of enrolled individuals.

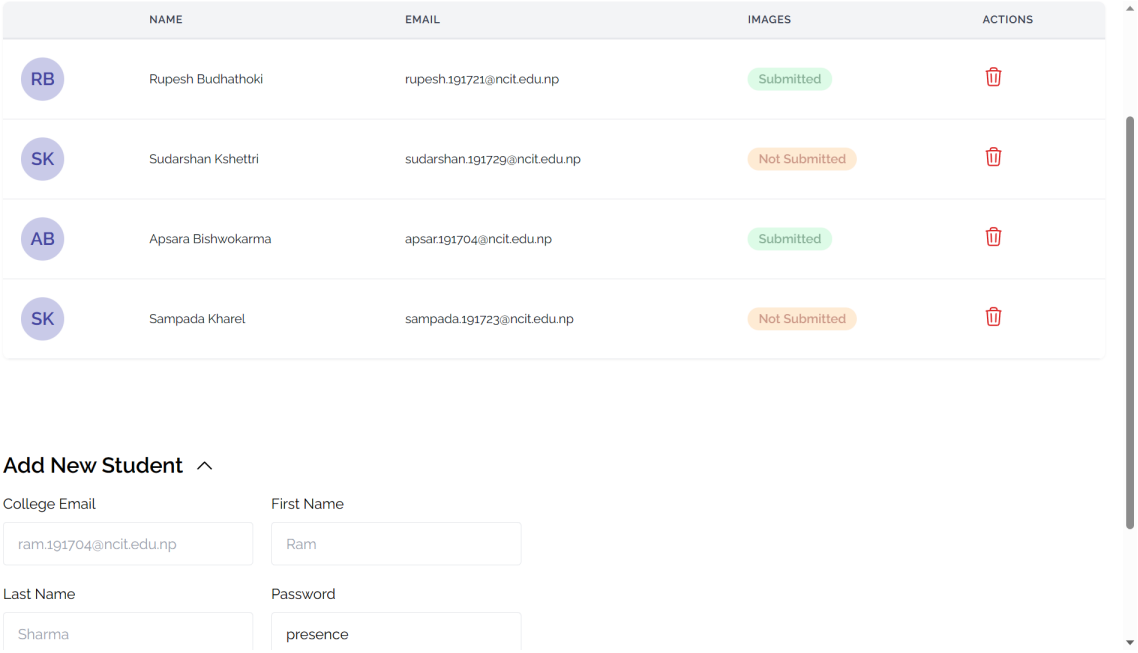

Figure 13: Student Management

- **Model Training**: Authorities have the ability to train the facial recognition model using images submitted by students, ensuring accurate and reliable attendance detection.

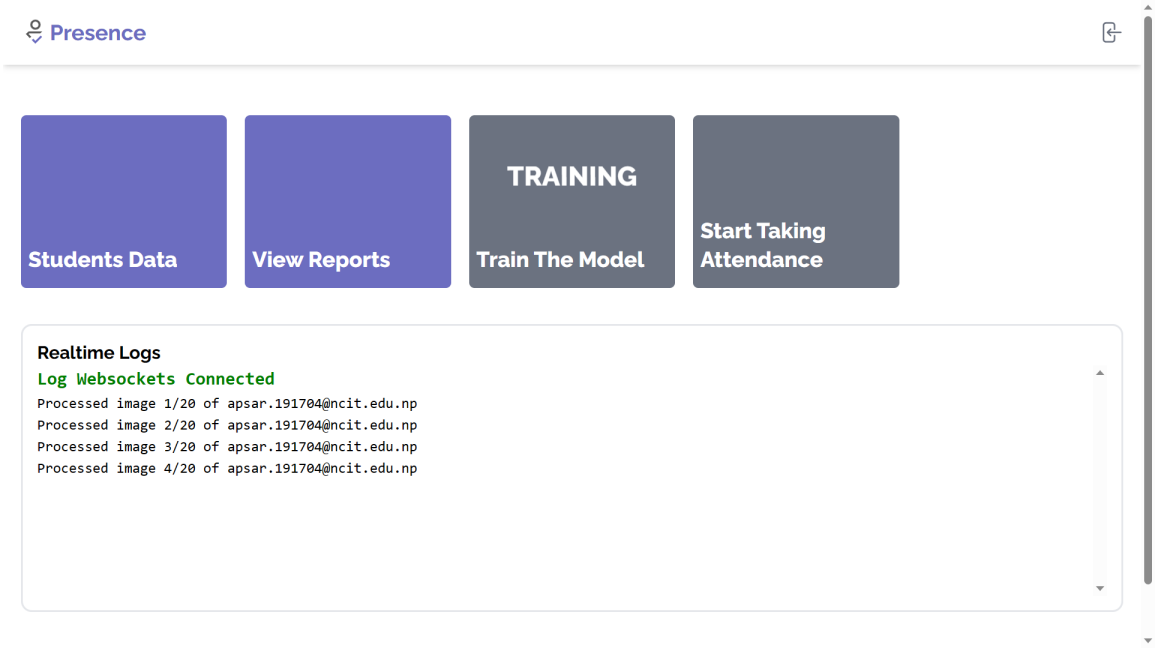

Figure 14: Training model with images submitted by students

- **Real-time Attendance**: Authorities can initiate the real-time attendance process, enabling the system to mark attendance as students are recognized.

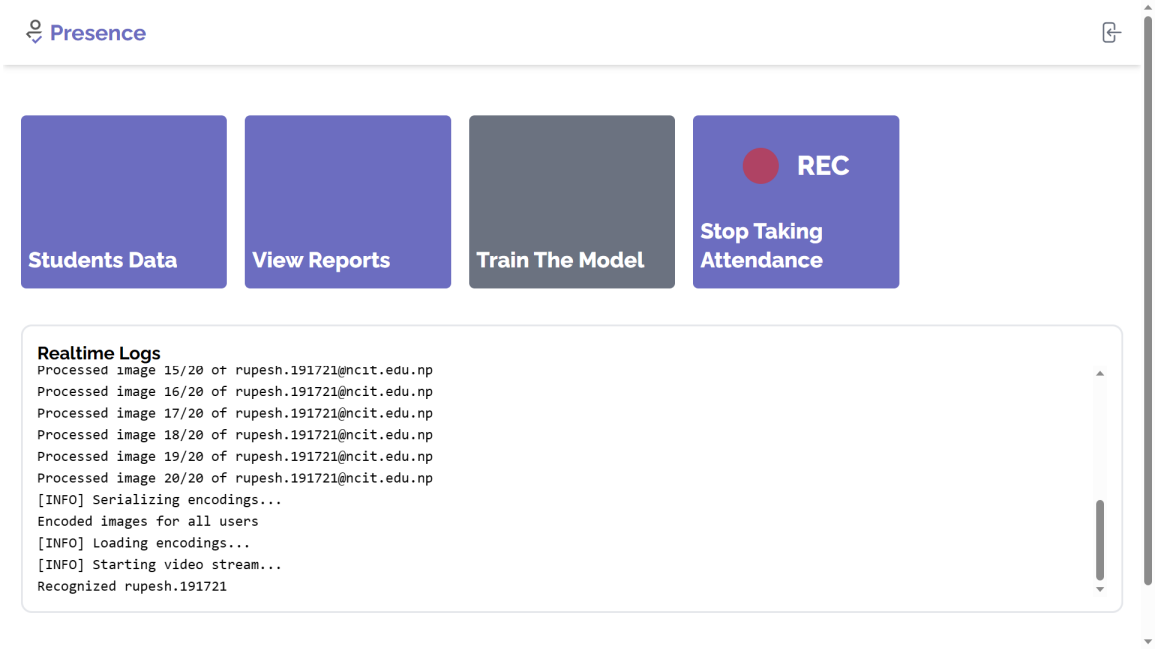

Figure 15: Realtime attendance log

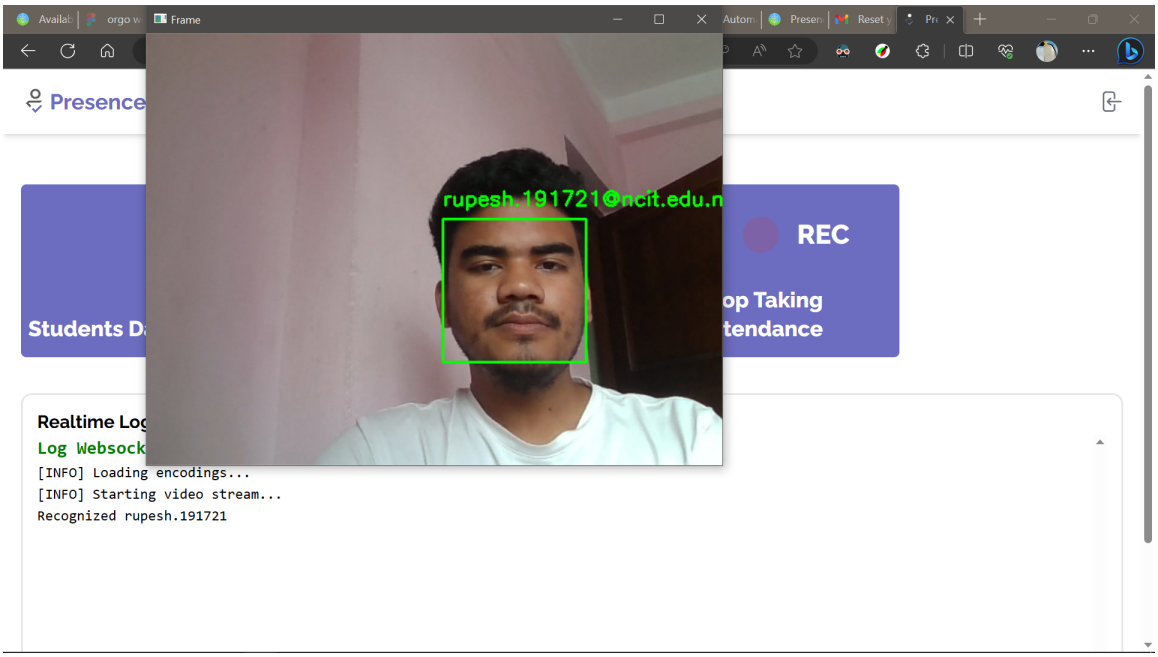

Figure 16: Realtime attendance from cam

7 Performance Analysis and Validation

This section focuses on evaluating the effectiveness, efficiency, and reliability of the implemented system. This involves conducting thorough assessments and tests to ensure that the system meets the required performance standards. Here's an overview of the performance analysis and validation aspects:

- **Facial Recognition Accuracy:** The accuracy of the facial recognition system is a critical factor in reliable attendance tracking. The performance analysis involves measuring the precision and recall rates of the HOG-based facial recognition method. Through validation, we verify that the system can correctly identify and match students' faces for accurate attendance marking. It struggles a bit when the side view of a student's face is shown. This is mainly due to less number of datasets we are using which is 10 images per student.

- **Response Time of APIs and WebSockets**: The speed and responsiveness of the REST APIs and WebSockets are crucial for seamless data transfer and real-time attendance updates. The performance analysis involves measuring the average response time of API requests and WebSocket communication. Validation ensures that the system meets the performance requirements, providing prompt and efficient communication between clients and the server. Some of the API's response time are as follows

  - GET: /all-students 20 to 30ms
  - POST: /create-new-student 300ms
  - Training time per image 1 to 2s

  WebSockets are instantaneous once they are connected.

- **System Stability**: The project is made with new and growing technologies like Nextjs and Django, which provides great scalability in the future and is able to handle many concurrent users easily. However, in our case, the number of people accessing the system is very low.

- **Authentication Mechanisms**: The authentication mechanisms, including login, login with Google, forgot password, and reset the password, are critical for system security and user access control. The performance analysis involves evaluating the response time and security measures of these authentication processes. Validation ensures that the authentication mechanisms function as intended, providing secure and efficient user authentication.

By conducting comprehensive performance analysis and validation, we can identify and address any performance bottlenecks, optimize system components, and ensure that the Presence Project meets the required performance standards. This process helps in delivering a reliable, efficient, and high-performing attendance management solution.

8 Tasks Remaining

With a significant portion of our project already completed, there are still several important tasks remaining to be accomplished. These tasks will require our focused attention and effort to ensure the successful delivery of our automated attendance system. The following are the tasks that are yet to be completed:

- **Record attendance for each subject**: For now we are only recording attendance for a day. We are planning to add functionality to record the attendance of each subject for better reports and insights.
- **Data Analysis and Report Generation**: Analyze the collected attendance data to derive insights and generate comprehensive reports for educators and administrators.
- **Testing and Validation**: Conduct user testing to ensure system functionality, perform integration testing, evaluate performance, and validate the accuracy of attendance tracking.
- **User Testing**: Engage users to obtain feedback on usability and implement improvements based on their suggestions.
- **Deployment** : Make ready-to-use deployment containers with Docker for backend services and host the frontend application to Vercel.

With the remaining time in the project timeline, we will focus on implementing the remaining functionalities, conducting thorough testing and quality assurance, and refining the system based on user feedback. Our objective is to deliver a fully functional and reliable automated attendance system that meets the expectations and requirements of our users.

9 Deliverable/Output

The proposed deliverables for the Presence project align with the defined architecture and project objectives. These deliverables encompass various components and functionalities of the automated attendance system, ensuring a comprehensive solution for tracking and analyzing student attendance.

Database Design and Implementation

The first deliverable involves designing and implementing the database architecture that will serve as the backbone of the Presence system. This includes creating the necessary tables, relationships, and entities to store student information, attendance records, subject details, and other relevant data. The database design will prioritize efficiency, scalability, and data integrity.

Python Application

The Python application will form a crucial component of the Presence system. It will be the main backbone of our system that will recognize students and mark their attendance. It will also handle the communication between the database and the client application, enabling CRUD operations and WebSocket communication. The deliverable will include the development and deployment of the Python application, ensuring its stability, security, and seamless integration with other system components.

Client Application

The client application will provide a user-friendly interface for students to access their attendance information securely. It will be designed with intuitive navigation, responsive design, and robust authentication mechanisms. The deliverable will include the development of the client application, ensuring compatibility across different devices and browsers.

Authority Access and Reporting

The Presence system will offer authorized personnel, such as administrators or instructors, access to comprehensive attendance data and reporting functionalities. This deliverable involves developing a secure login system for authorities, granting them access to view and analyze attendance records, generate reports, and gain insights into attendance patterns and subject preferences. The reporting component will leverage Python programs to generate insightful reports based on the collected data. Also, students can view their own information with their login credentials and report if any mistakes happened.

10 Project Task and Time schedule

The working time period for the project is four months. The project will be completed by the end of the spring semester as per the requirements of the university. The major task division among the team members is mentioned in Table 3.

Team Member	**Assigned Task**
Rupesh Budhathoki	Project Management, Overall Coordination, FullStack Development
Apsara Bishwokarma	Frontend Development
Sampada Kharel	Backend Development
Sudarshan Kshettry	FullStack Development

Table 3: Division of tasks among project team members

The time schedule proposed for the development of the project is illustrated in the following Gantt chart

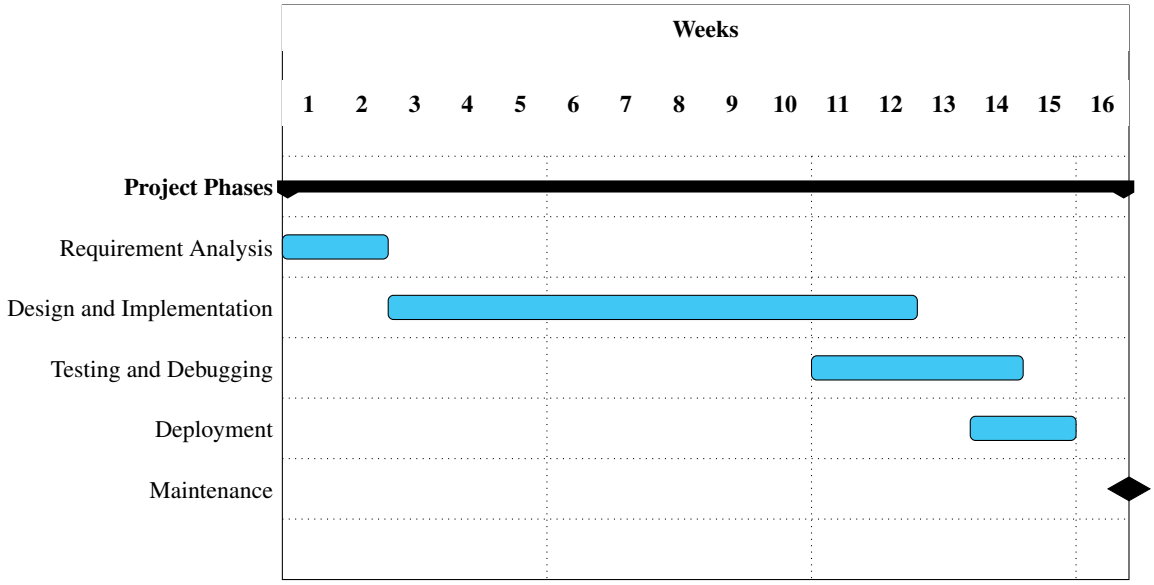

Table 4: Gantt Chart

11 References

[1] K.-O. Goh, C.-Y. Law, C.-K. Tin, C. Tee, and Y.-W. Sek, "Attendance management system with half-covered and full-facial recognition feature," in *Proceedings of the 9th International Conference on Computational Science and Technology: ICCST 2022, 27–28 August, Johor Bahru, Malaysia*, Springer, 2023, pp. 247–258.

[2] D. Marsden, *Mod-attendance plugin on moodle lms*, 2023.

[3] Moodle, *Attendance activity on moodle lms*, 2023.

[4] D. Nurcahya, H. Nurfauziah, and H. Dwiatmodjo, "Comparison of waterfall models and prototyping models of meeting management information systems," *Jurnal Mantik*, vol. 6, no. 2, pp. 1934–1939, 2022.

[5] M. L. Yasmeen, M. H. R. Reddy, B. Mridgala, M. P. Reddy, and P. Chandana, "Smart attendance using facial recognition," *YMER*, vol. 22, no. 1, pp. 370–373, 2023.