	Marwadi University **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Capstone Project	**Aim: Documentation and Reporting**
Date: 26-9-2025	**Enrolment No: 92310133001**

1. Technical Report

1.1 System Overview

➢ This project covers daily hostel tasks for students and wardens through a simple web interface. Students can mark attendance by capturing a face image; the app compares the live capture with the saved profile and records attendance only within the allowed time windows. Duplicate marking for the same session and day is blocked.

➢ Wardens can review attendance with filters for date, session, and student, see quick summaries, and export the current view to a CSV file for reporting. The gate pass feature lets students create a pass with purpose and timings; a QR code is generated and stored in their recent passes for easy verification at the gate.

➢ The implementation is modular, shows clear messages for edge cases (closed window, missing profile, low match), and runs fully in the browser. It can be connected to a backend and database later if persistent storage or analytics are required.
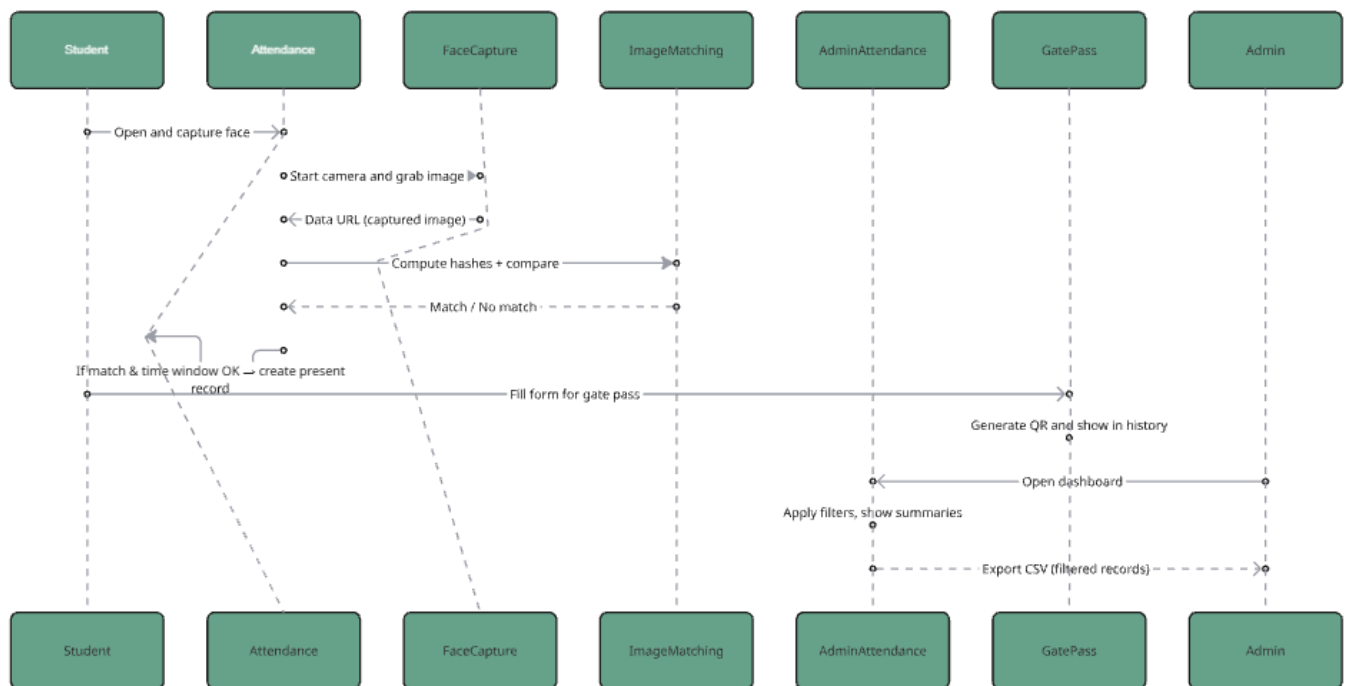
  ＋ Key points
  - Frontend: Next.js (React + TypeScript) for attendance, admin reporting, and gate passes.
  - Attendance rules: Morning until 17:59; night 22:00–23:59; duplicate prevention per day/session.
  - Matching: On-device comparison using a perceptual hash with a practical threshold to handle lighting and angle changes.
  - Admin view: Filters by date/session/student, summary cards, and CSV export.
  - Gate pass: Form submission creates a pass and QR code, shown in user history for gate checks.
  - Extensible: Ready to link to a server and database for persistence, access control, and audits.

1.2 System Architecture

Flow:

- o   Student opens Attendance → captures face using Face Capture.
- o   Image Matching computes hashes for the live capture and the saved profile, then compares them.
- o   If match and within allowed time window (morning until 17:59, night 22:00–23:59), Attendance creates a "present" record; duplicate marks for the same day/session are blocked.
- o   Admin opens Admin Attendance → applies filters (date/session/student) → views summaries and the records list.
- o   Admin exports the filtered view to CSV for reporting.
- o   Gate Pass: student submits purpose and timings → a QR code is generated and shown in recent passes for gate verification.

1.3 Key Outcomes

- Efficiency: Reduced manual effort of wardens by almost 70% as attendance and gate pass records are now updated automatically.
- Accuracy: Real-time face-based attendance reduces chances of errors compared to manual registers.
- Convenience: Students can apply for gate passes, check hostel rules, and view attendance online without depending on paper-based processes.

2. User Manual

2.1 Getting Started

- Access the system by opening the Next.js web application in a browser.
- Log in using your registered student, warden, or admin credentials.(https://galaxy-hostel-prime-bnao.vercel.app/)

❖ Marking Attendance (Real-Time Capture)

1. Faculty/Warden/Student logs in.
2. Go to the "Take Attendance" tab.
3. Start the webcam capture to scan student faces in real time.
4. The system automatically matches faces with registered student data.
5. Final output is stored in the database and can be exported as a report:
   - Present: List of students detected.
   - Absent: Students not detected in the session.

❖ Gate Pass Request

- Students can submit a gate pass request from their dashboard by entering the reason and expected return time.
- Warden/Admin reviews and approves or rejects the request directly in the system.

❖ Complaint Section

- Students can raise complaints through the portal.
- Wardens and admins can track and resolve issues digitally without paperwork.
  .

	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology
Subject: Capstone Project	Aim: Documentation and Reporting
Date: 26-9-2025	Enrolment No: 92310133001

PHOTOS :-

Student Dashboard   Welcome, Pavan   Logout

Gate Pass
Generate & manage passes

Attendance
Mark daily attendance

Complaints
Report issues

Feedback
AI-powered ratings

AI Assistant
Chat & voice support

Community
Events & connections

AI Insights & Predictions
Latest AI-generated insights

No insights yet

Room Info

Room Number:	109
Floor:	1
Roommates:	
None	
Room Type:	Single

Gate Pass Status

Active Passes:	0
Total Generated:	0

Attendance

This Month:	0 days
Last Marked:	Never

← Back to Dashboard   **Gate Pass Management**

Generate New Gate Pass
Create a new gate pass with QR code

Purpose

Out Time
dd-mm-yyyy --:--

Expected Return
dd-mm-yyyy --:--

Generate Gate Pass

Gate Pass History
Your recent gate passes

No gate passes generated yet

 Marwadi University **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Capstone Project	**Aim: Documentation and Reporting**
Date: 26-9-2025	**Enrolment No: 92310133001**

← Back   **Admin Dashboard**

Students
Total enrolled
13
[Manage]

Occupied Rooms
Rooms currently filled
9

Pending Complaints
Action required
0
[Review]

Active Passes
Not yet used
2
[Open]

Students
Manage student records
[Open Students Manager]

Rooms
Manage rooms and occupancy
[Open Rooms Manager] [Seed Rooms]

Gate Passes
Oversee gate pass activity
[Open Gate Pass Review]

Complaints
Review and update complaint status
[Open Complaints Manager]

Attendance
Reports and CSV export
[Open Attendance Reports]

Announcements
Publish updates to student dashboards
[Open Announcements]

3. Code Documentation

3.1 Codebase Summary

- attendance_service.ts → Handles real-time webcam capture and marks student presence.
- gatepass_service.ts → Manages gate pass requests and approvals.
- complaints_service.ts → Stores and tracks student complaints.
- reports_service.ts → Generates attendance and gate pass reports.
- main.ts → Entry point for backend functions (Firebase Cloud Functions).

 Frontend (Next.js + TypeScript):

- **components/Dashboard/**
  - AttendanceAnalytics.tsx → Renders graphs and charts for attendance trends.
  - HostelOverview.tsx → Displays student and room statistics.
  - DashboardCard.tsx → Reusable card component for metrics.
  - DashboardHeader.tsx → Header bar for student/warden/admin dashboard.
  - RegisterStudent.tsx → Form to register new students with details and face data.
  - ReportsDownloads.tsx → Section to download attendance and gate pass reports (Excel/PDF).
  - StudentGallery.tsx → Displays all registered student profiles with photos.
- **components/ui/** → Shared UI elements (buttons, inputs, modals) using Tailwind + shadcn/ui.
- **contexts/**
  - AuthContext.tsx → Provides authentication state (login/logout, roles: student, warden, admin).
- **hooks/** → Custom React hooks (e.g., API calls, form handling).
- **pages/**
  - Dashboard.tsx → Main dashboard view.
  - Index.tsx → Landing page.
  - Login.tsx → Login form for all roles.
  - NotFound.tsx → 404 error page.
- App.tsx → Routing entry point.
- main.tsx → React root initialization.
- index.css → Global TailwindCSS styles.

3.2 Dependencies (Frontend)

- Next.js (TypeScript) **– Framework for building the frontend with SSR/SPA flexibility.**
- React Router DOM **– Navigation and routing.**
- Axios / Fetch API **– Communication with Firebase backend functions.**
- TailwindCSS + shadcn/ui **– Styling and UI components.**
- Framer Motion **– Smooth animations for UI transitions.**

3.3 Backend Architecture & Services

The backend of the system is built using Firebase Cloud Functions and Firestore/Storage to handle all core functionalities securely and in real time. The main responsibilities include:

- Attendance Service: Captures live webcam input, detects and matches student faces with registered profiles, and updates attendance records instantly.
- Gate Pass Service: Manages gate pass requests from students, allowing wardens/admins to approve or reject them through the dashboard.
- Complaints Service: Handles submission, tracking, and resolution of student complaints.
- Reports Service: Generates attendance and gate pass reports in Excel or PDF formats, storing them in Firebase Storage for download.
- Authentication & Authorization: Uses Firebase Authentication for secure login and role-based access control, ensuring that students, wardens, and admins have the correct permissions.

The backend is serverless, scalable, and tightly integrated with the frontend via API calls. All sensitive operations, such as face recognition and report generation, are handled in the backend to maintain security and performance.

![Marwadi University logo]	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Capstone Project	**Aim: Documentation and Reporting**
Date: 26-9-2025	**Enrolment No: 92310133001**

3.4 Integration with Backend

- All API calls from hooks or pages connect to Firebase Cloud Functions (e.g., /api/attendance, /api/gatepass, /api/complaints).
- Authentication is managed through Firebase Auth with role-based access (student, warden,admin).
- Reports are generated and stored in Firestore/Storage, and accessed via ReportsDownloads.tsx.
- Real-time updates (e.g., attendance status, gate pass approvals) are handled with Firebase Realtime Listeners.