Of course. Here is the information structured into a formal project document that you can share with your teammates.

**Project Brief: Automated Face Recognition Attendance System**

**Version:** 1.0 **Date:** August 28, 2025

**1.0 Project Vision**

This document outlines the architecture and features for an automated attendance system designed to replace manual, error-prone methods. The system will leverage facial recognition and in-classroom device validation (via Bluetooth) to ensure accurate, secure, and effortless attendance tracking. The primary goals are to eliminate proxy attendance, save valuable class time, and provide actionable analytics to faculty to improve student engagement.

**2.0 Core Problems to Solve**

* **Inefficient Manual Processes:** Eliminate time-consuming roll calls.
* **Inaccuracy & Proxy Attendance:** Prevent fraudulent attendance marking.
* **Lack of Actionable Data:** Convert raw attendance data into useful insights for educators.

**3.0 System Architecture & User Roles**

The system will use a modern client-server architecture with three main components: a **Web Frontend**, a **Backend API**, and a **Database**.

There are three defined user roles:

* **Administrator:** Manages the system's foundational data.
* **Teacher/Faculty:** Accesses analytics and manages courses.
* **Student:** Marks their own attendance.

**4.0 Detailed Feature Breakdown**

**Module 1: Administrator Portal (The Foundation)**

This is a secure, admin-only web interface for initial system setup and ongoing management.

* **User Management:** A portal to create, edit, and delete student and faculty accounts. This includes uploading initial photos for face recognition enrollment.
* **Course & Timetable Management:** An interface to define courses, assign faculty, and create/upload the master college timetable. Each timetable entry will contain the course name, teacher, schedule, and classroom number.
* **Classroom Device Mapping:** A critical feature to map a classroom to the unique Bluetooth ID of teacher assign to that class form her/his phone . This provides room-level location accuracy.
  + *Example:* Room 3B -> BenQ-SmartBoard-A4B9

**Module 2: Student Attendance Workflow (The Core User Journey)**

This workflow is designed to be seamless for the student and require zero action from the teacher.

1. **Login & Dashboard:** The student logs into the web application. The dashboard automatically displays their daily schedule. If a class is currently active, a "Mark Attendance" button will be visible.
2. **Initiate Attendance:** The student clicks the button to start the process.
3. **Validation Step 1: Schedule Check (Backend):** The backend verifies that the request is within the scheduled start and end times for that lecture.
4. **Validation Step 2: Location Check (Frontend):** The web app uses the **Web Bluetooth API** to scan for the classroom's pre-registered smart device. If the device isn't detected, the process fails.
5. **Validation Step 3: Identity Check (Frontend/Backend):** If the location is verified, the front camera activates. A photo is captured and sent to the backend, which uses facial recognition to confirm the student's identity.
6. **Confirmation:** Upon successful validation, the attendance is logged in the database, and the student receives a success message.

**Module 3: Teacher Analytics Dashboard (Actionable Insights)**

A powerful interface for faculty to monitor and analyze attendance data.

* **Data Filtering:** Allows teachers to filter data by class, student, and date range.
* **Key Performance Indicators (KPIs):** Displays the overall attendance percentage for the filtered selection.
* **Trend Analysis:** A line graph visualizing attendance over the selected period to identify patterns.
* **"Students at Risk" Identifier:** An automatically generated table that flags students with the lowest attendance, enabling proactive intervention.
* **Reporting:** A one-click feature to export detailed attendance logs as a CSV or PDF file for official records.

**Module 4: Backend Automation Engine (The "Magic")**

This module ensures "true automation" by removing the teacher from the process entirely.

* **Scheduled Job:** A background process runs on the server every minute.
* **Function:** The job queries the timetable database and automatically updates the status of classes to "ACTIVE" when they start and "FINISHED" when they end.
* **Result:** The student's dashboard is controlled entirely by these automated status updates. The teacher is not required to start, stop, or manage any part of the attendance-taking process.

**5.0 Proposed Technology Stack**

* **Frontend:** React, Vue, or Angular (for a rich UI). Key browser API: **Web Bluetooth API**.
* **Backend:** Python with **FastAPI** or **Flask** (for building the API).
* **Database:** **PostgreSQL** or **MySQL** (for structured data like users and timetables).
* **Key Libraries:**
  + face-recognition / dlib for facial recognition.
  + pandas for data manipulation and report generation.
  + APScheduler for running the automated background job.
  + Chart.js (or a similar JavaScript library) for the dashboard visualizations.