📘PROMPT.md

Project: Automated Student Attendance Monitoring and Analytics System for Colleges
SIH ID: SIH25016

📌Problem Statement

Attendance in most colleges is still manual (roll call/registers). This wastes class time, allows **proxy attendance**, and provides no useful analytics to teachers/admins.

We need a **secure, proxy-free, gamified, automated attendance system** that works **offline-first**, supports **multi-factor verification**, and provides **analytics dashboards** for all stakeholders.

🕗Objectives

- Build a **Next.js (JavaScript) PWA** with **offline-first support**.
- Use **MongoDB** for centralized database storage.
- Implement **multi-factor attendance verification**:
- **QR Code session join**
- **Wi-Fi / Bluetooth proximity validation**
- **Face / Fingerprint / PIN biometric verification**
- Provide **gamified dashboards** for students (attendance streaks, ranks, daily scores).
- Provide **real-time dashboards** for teachers (live attendance, session stats).
- Provide **analytics dashboards** for admins (graphs, risk students, trends).
- Ensure **security, periodic biometric revalidation, and no proxy attendance**.

⚙️Features

Student Module

- **Login/Registration** → username & password.
- **Onboarding (First-time login)**: upload photo, face scan, fingerprint/PIN, complete profile.
- **Admin approval required** before activation.
- **Reverification every 6 months** → face rescan for biometric update.
- **Join Attendance Session**:
- Scan QR or enter session code.
- System validates Wi-Fi SSID/Bluetooth MAC.
- System verifies biometric (face/fingerprint/PIN).
- If all checks pass → attendance marked ✅
- **Dashboard (Gamified)**:
- Attendance streaks, points, daily score.

- Leaderboard (top attendance).
- Auto-warning if attendance < threshold.
- College notices and announcements.

Teacher Module

- **Start Session** → generates QR code + session code + Wi-Fi/BLE data.
- QR/session validity expires after a few minutes.
- Option to **end/extend session**.
- **Fallback** → manual attendance if tech fails.
- **Dashboard**:
- Live student list.
- Session statistics.
- Export reports.

Admin Module

- **User Management** → approve/reject new students, manage teachers.
- Reset credentials, deactivate accounts.
- **Dashboard**:
- College-wide attendance analytics.
- Pie/bar/line charts.
- Identify low-attendance students.
- Export analytics CSV/PDF.
- **Audit Logs** for all activity.

🛠️Tech Stack

- **Frontend**: Next.js (JavaScript), TailwindCSS, Dexie.js (IndexedDB offline support).
- **Backend**: Node.js + Express (or Next.js API routes).
- **Database**: MongoDB.
- **Authentication**: JWT + refresh tokens.
- **Proximity Validation**: Wi-Fi SSID check + Bluetooth MAC scan.
- **Biometric Verification**:
- Face recognition → `face-api.js` (TensorFlow.js).
- Fingerprint/PIN → Device-based fallback.
- **Charts/Analytics**: Recharts / Chart.js.
- **PWA Features**: Manifest, service workers, offline sync.

🗂️Suggested File Structure

```
attendance-system/
|── backend/
│   ├── models/          # MongoDB schemas (User, Session, Attendance)
│   ├── routes/          # API endpoints
│   ├── controllers/     # Business logic
│   ├── utils/           # Helpers (JWT, QR gen, Wi-Fi/BLE validation)
│   ├── server.js        # Express/Next.js API server
│
|── frontend/
│   ├── pages/
│   │   ├── index.js            # Landing/Login
│   │   ├── student/
│   │   │   ├── dashboard.js
│   │   │   ├── profile.js
│   │   │   └── attendance.js
│   │   ├── teacher/
│   │   │   ├── dashboard.js
│   │   │   └── session.js
│   │   ├── admin/
│   │   │   ├── dashboard.js
│   │   │   └── users.js
│   │   ├── api/                # Next.js API routes (if used)
│   │
│   ├── components/             # Reusable UI (QR Scanner, Charts, Tables)
│   ├── hooks/                  # Custom hooks (useAuth, useOfflineSync)
│   ├── utils/                  # QR decoding, biometric verification
│   ├── styles/                 # Tailwind/global styles
│
|── public/
│   ├── qr-icons/               # QR images
│   ├── manifest.json           # PWA config
│
|── package.json
|── README.md
```

📊Attendance Session Lifecycle

1. Teacher starts session → system generates **QR + session code + Wi-Fi/BLE data**.
2. Student scans QR or enters session code.
3. System runs **3 checks**:
4. ✅Proximity (Wi-Fi/BLE match)

5. ✅Biometric (face/fingerprint/PIN)
6. ✅QR/session validity (not expired)
7. If all checks pass → attendance marked.
8. Student dashboard updates (streaks, points).
9. Teacher dashboard updates (live list).
10. Admin dashboard updates (analytics, logs).

🔐Security Features

- JWT + refresh token system.
- Session QR codes expire in minutes.
- Proximity + biometric combo = no proxy attendance.
- Mandatory biometric revalidation every 6 months.
- Audit logging for admins.

💡 Extra Ideas

- **AI Predictions** → flag students likely to fall short of 75%.
- **Parental Alerts** → auto-notification if attendance < 70%.
- **Gamification Add-ons** → badges, streak rewards, semester-based leaderboards.
- **Cross-Platform Ready** → PWA runs on web + can be wrapped into Android app.

✅Use this file as the **source prompt** inside any AI coding agent to generate the system step by step.