**HIGH LEVEL DESIGN**

**Introduction**

The High-Level Design (HLD) describes the major components of the Automated Attendance System. It outlines the system's architecture, core modules, and how the components interact. The system utilizes QR code-based attendance tracking, role-based access, and real-time monitoring for students, lecturers, and administrators.

**System Architecture Components**

|  |  |  |
| --- | --- | --- |
| **Layer** | **Component** | **Technology / Description** |
| **Presentation Layer** | Mobile Frontend | React.js – User Interfaces for Students, Lecturers, Admins |
| **Application Layer** | RESTful API Server | Node.js – Handles all business logic, data validation, and routing |
| **Data Layer** | Relational Database | MongoDB – Stores all users, courses, attendance, and system logs |
| **QR Code Module** | QR Code Generator & Scanner | Generates time-bound QR codes; integrated with mobile scanning |
| **Authentication** | Auth Module | JWT-based Login System with hashed passwords and role-based access control |
| **Notification Module** | Optional | Email or SMS-based alerts (absences, announcements, etc.) – future expansion |
| **Admin Dashboard** | Admin Interface | Manages courses, users, schedules, and logs |
| **Logs** | Logging/Audit System | Tracks user activity, login/logout records |

**Module Overview**

|  |  |
| --- | --- |
| **Module** | **Functions** |
| **Student Module** | - View attendance history - Scan QR code to check-in |
| **Lecturer Module** | - Generate QR code - Monitor attendance - View enrolled students |
| **Admin Module** | - Manage users, roles, and courses - View system logs |
| **Attendance Module** | - Record attendance from QR scans - Validate student and session details |
| **Course Management** | - Add/update courses - Assign lecturers to classes |
| **QR Management** | - Create time-limited QR codes - Mark QR codes as expired automatically |
| **Login & Security** | - Hash passwords - Generate/verify JWT tokens - Role-based access |

**Technology Stack**

|  |  |
| --- | --- |
| **Component** | **Technology Suggestion** |
| Frontend | React.js |
| Backend API | Node.js |
| Database | MongoDB |
| QR Code Handling | QRCode libraries |

**Security Considerations**

* Passwords stored using strong hashing.
* Role-based access control to restrict data exposure.
* Expiring QR codes to prevent re-use.
* Input validation on all form entries and API endpoints.
* Secure API tokens using HTTPS.

**Future Enhancements**

* SMS/Email Notifications
* Analytics Dashboard for Admin
* Machine Learning for detecting anomalies in attendance
* Biometric integration

HIGH LEVEL DESIGN DIAGRAM

