Project Documentation

SIH 2025 Internal Hackathon - BCIIT

Submission Deadline: 29th September 2025, 12:00 PM

DOCUMENT STRUCTURE

1. PROJECT OVERVIEW

Project Title: Automated Student Attendance Monitoring and Analytics System for Colleges.

Team Name: CodeMads

Problem Statement: Automated Student Attendance Monitoring

and Analytics System for Colleges. (PS Number: SIH25012)

Category: Software

Date: 29th September 2025, 12:00 PM

2. TEAM INFORMATION

Team Leader: [Christina Baiju] - [BCA] - [2nd] - [7982082204] - [Christina4406@gmail.com]

Member 2: [Arushi Sharma] - [BCA] - [2nd] - [9540657482] - [arushisharma1505@gmail.com]

Member 3: [Shambhavi Dadhich] - [BCA] - [2nd] - [7011454822] - [shambhavidadhich@gmail.com]

Member 4: [Deepanshi Dev] - [BCA] - [2nd] - [7011940577] - [devdeepanshi@gmail.com]

Member 5: [Mehak Arora] - [BCA] - [2nd] - [9625557419] - [mehakarora2886@gmail.com]

Member 6: [Rishabh Jha] - [BCA] - [2nd] - [8076131136] - [jharishabh017@gmail.com]

TECHNICAL DOCUMENTATION

3. PROBLEM STATEMENT ANALYSIS

3.1 Problem Description:

Most colleges still rely on manual attendance methods such as roll calls or paper registers. This process is time-consuming, error-prone, and often enables proxy attendance. In large classes, these challenges become even more difficult to manage. Moreover, faculty and administrators lack streamlined access to attendance insights, making it harder to identify trends or support at-risk students. With education rapidly digitizing, traditional systems are inefficient and outdated. A modern, automated attendance system with analytics is essential to ensure accuracy, efficiency, and better academic planning.

Impact / Why This Problem Needs to Be Solved

- Saves teaching time by automating attendance.
- Minimizes errors and prevents proxy attendance.
- Provides actionable insights to track engagement and student performance.
- Increases transparency and accountability.
- Supports the digital transformation of higher education.

Stakeholders / Beneficiaries

- Students
- Faculty and academic administrators
- College management bodies
- Education departments and policymakers

3.2 Existing Solutions Analysis:

Current attendance tracking solutions use RFID cards, QR code scanning, biometric systems, and mobile apps to automate recording. They provide real-time data, reports, and integration with learning management systems. However, they often face challenges like high costs, infrastructure dependencies, user complexity, and limited customization for specific educational contexts.

Our approach is simpler and more user-friendly, designed to work effectively in both physical and online environments (Scalable). It prioritizes reducing proxy attendance, saving time, downloading the attendance easily, and delivering actionable analytics tailored to college settings, overcoming common limitations of cost, complexity, and adaptability in existing solutions.

3.3 Solution Objectives:

- Automated attendance system using QR codes.
- Dashboard for administrators and faculty to review attendance records.
- Analytics to identify attendance trends and student engagement levels.
- Compatibility with the Offline setting at the moment, but scalable.

4. SOLUTION ARCHITECTURE

4.1 System Overview:

High-level architecture diagram

[User Interfaces]

|--- Admin Portal

|--- Student Portal

↓

[Web/Application Server]

|--- Attendance Capture Module

|--- Analytics & Reporting Module

↓

[Database Server]

|--- Attendance Records

|--- Reports & Analytics Data

4.2 Technology Stack:

Frontend: [HTML, JavaScript, CSS]

Backend: [SQL, PHP]

Database: [SQL]

Cloud/Hosting: [Deployment platform]

Tools & Libraries: [QR Tiger]

Hardware: [Windows 11, Any Mobile Phone]

4.3 System Design:

Table	Description
users	Stores user information: id, role, username, password, rollno
attendance	Stores attendance data linked to users and timestamps

Users:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int			No	None		AUTO_INCREMENT	Change Drop More
2	role	enum('student', 'admin')	utf8mb4_0900_ai_ci		No	None			Change Drop More
3	username	varchar(50)	utf8mb4_0900_ai_ci		No	None			Change Drop More
4	password	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
5	rollno	varchar(20)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

Attendance:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	int			No	None			Change Drop More
3	Date	date			No	None			Change Drop More
4	Status	varchar(50)	utf8mb4_0900_ai_ci		No	None			Change Drop More

5. IMPLEMENTATION DETAILS

5.1 Core Features Implemented:

- Role-based access (Student/Admin)
- Secure login with password hashing for students
- Real-time attendance marking and management
- Admin dashboard with detailed reports

- QR Code Scanner to move to the Attendance Marking Page.
- Visually impactful pie chart to Showcase Attendance for every day (changing).
- Easily accessible CSV file to download the attendance.

5.2 Development Approach:

Methodology: Hybrid approach combining Agile iterative development with Waterfall-style initial planning.

Planning: Gathered requirements from problem analysis and discussions; defined key features and security needs.

Design: Created wireframes and database schema to visualize the app flow before coding.

Development: Built with core PHP and MySQL, used Chart.js for visualization, followed Git version control with iterative reviews.

Testing: Manual testing, validations, and basic security implementation; automated tests planned for future.

Deployment: Prepared on local XAMPP.

Maintenance: Managed via version control with debugging, user feedback, and planned feature upgrades.

Task distribution among team members:

1. *Christina Baiju:* QR_HTMLJS.php,Login_page.php, Attendance_Admin_Dashboard.php, Database (attendance_system), Documentation

2. *Arushi Sharma:* Login_page.php, Login_CSS.css, Attendance_Admin_Dashboard.php, Documentation

3. *Shambhavi Dadhich:* Database (attendance_system)

4. *Deepanshi Dev:* Login_page.php (Data entry - scenarios), Documentation

5. *Mehak Arora:* Attendance_Admin_Dashboard.php

6. *Rishabh Jha:* Presentation

5.3 Challenges Faced & Solutions:

Challenge 1: Adding the QR Code Scanner using JavaScript effectively → [const qrScanner = new Html5Qrcode("qr-reader");]

Challenge 2: Making sure the attendance of students actually gets stored in the table → [Using MySQLi]

Challenge 3: Allowing only the faculty with admin credentials to view the dashboard to avoid student misuse of the switch roles option→ [Using a Prepared statement to ensure safe credential login.]

Challenge 4: Ensuring the data stored in the table is visually displayed correctly through the Pie chart→ [Chart.js]

Challenge 5: Creating an option for the admin to download the attendance by getting the records from the table.→ [fetch_csv to download attendance using GET.]

6. INSTALLATION & SETUP GUIDE

6.1 System Requirements:

Operating System: Windows 11

Software Dependencies: XAMPP Server, QR Tiger,

PhpMyAdmin

Hardware Requirements:

Component	Requirement
Camera	Smartphone/Laptop with working camera
Processor	1 GHz+ (for local PHP server)
RAM	2 GB+ (for XAMPP/local use)
Storage	Standard SSD/HDD, ~1 GB free
Browser	Modern browser (Chrome, Firefox, Edge)
Network	Stable internet for online operation
Server	Shared/VPS/cloud hosting with PHP+MySQL
Display	Projector/screen (optional, for QR code)

Network Requirements:

INTERNET - Wifi

APACHE (80,443)

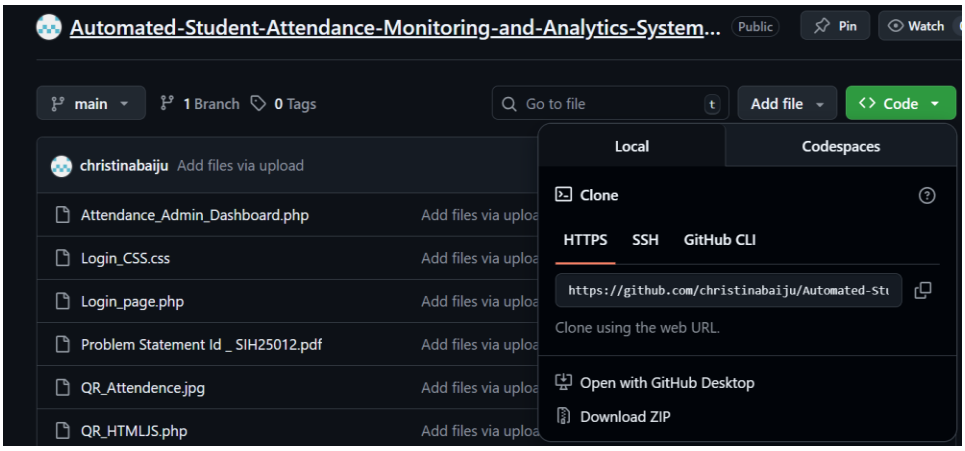
MySQL (3306)

6.2 Installation Steps:

STEP 1: Enter the following link in your web browser

https://github.com/christinabaiju/Automated-Student-Attendance-Monitoring-and-Analytics-System-for-Colleges.git

STEP 2: Go to code and download ZIP.



STEP 3: After completing the download, save it in the following location - C:\xampp\htdocs

STEP 4: Now turn on both Apache and MYSQL and write the code given in database.txt to create the table in PhpMyAdmin.

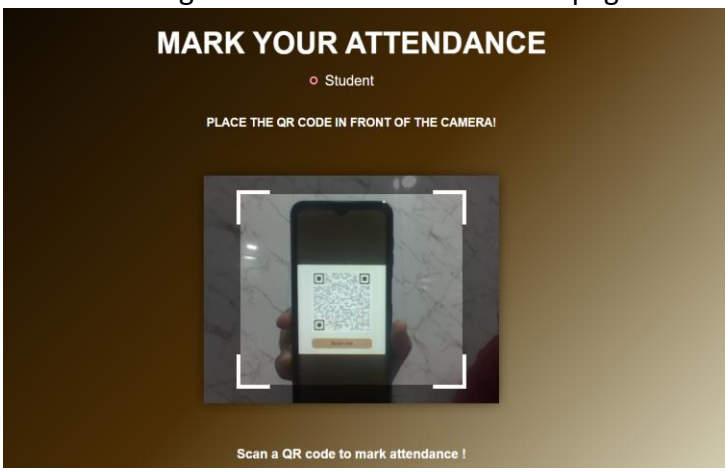
STEP 5: Your program is now ready to run start with the QR_HTMLJS.php and scan the QR_Attendence.jpeg.

7. USER GUIDE

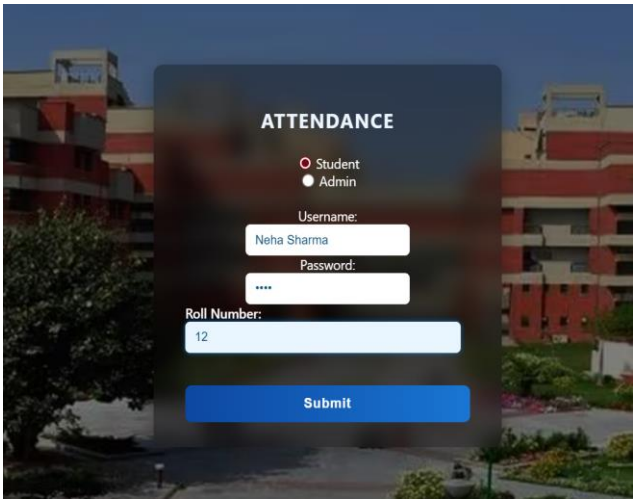
7.1 User Interface Overview:

STEP 1: 1st, the Student scans their QR Code available

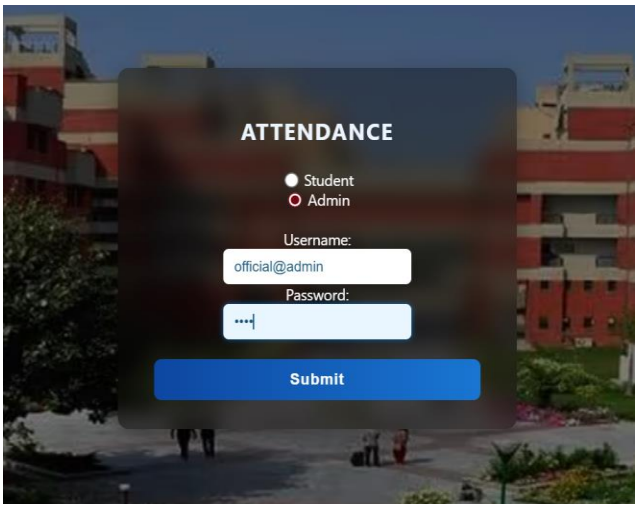
with them to gain access to the attendance page.



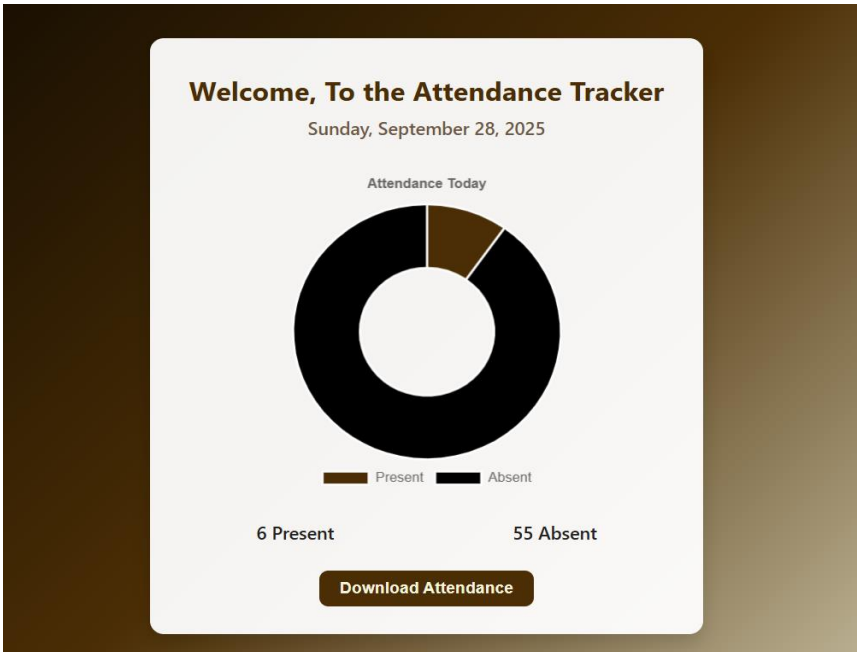
STEP 2: Once the QR code is scanned, it redirects the user to the following page where they can choose their role as their student to mark their attendance.



STEP 3: On the same web page if the Faculty/Professor enter their admin credentials ie; (official@admin, 1432). They will gain access to the Attendence tracker dashboard.



STEP 4: The database showcases all the students who were present that day from the total strength of 61 students.



STEP 5: Once the admin clicks the button to download the attendance, A CSV file gets downloaded with the records of students who marked their attendance that day.

	A	B	C	D	E
1	ID	Role	Username	Roll Number	
2	1	admin	official@admin		
3	38	student	Arushi	34	
4	39	student	Mehak	21	
5	40	student	Deepansh	54	
6	41	student	Shambhav	8	
7	42	student	Neha Shar	12	

8. TESTING & VALIDATION

8.1 Known Issues & Limitations:

Dependency on Smartphones and Internet:

QR code attendance requires students to have smartphones with a camera and internet connectivity for scanning and real-time attendance recording. Lack of devices or connectivity issues can hinder attendance marking.

QR Code Tampering and Duplication:

QR codes need secure generation and management. If codes are leaked or duplicated outside authorized channels, fraudulent attendance marking may occur.

No Biometric Identity Validation:

Unlike face recognition or fingerprint systems, QR code scanning alone cannot verify that the actual student is present, reducing security against impersonation.

Limited Offline Functionality:

Most QR code attendance systems require a network connection to sync attendance to the central database in real time. Offline scenarios need explicit handling to avoid data loss.

User Training and Device Compatibility:

Students and staff must be familiar with scanning apps and compatible devices, or else usability issues can arise.

8.2 Future improvements needed:

- Admin passwords are currently stored in plain text (consider hashing for security)
- Roll number input validation can be improved
- Mobile UI tweaks needed for smaller screens
- Add the ability to change profile settings and FAQ's in the Dashboard for effective use.
- Add option for course/class-based attendance.

9. DEPLOYMENT & LIVE DEMO

9.1 Deployment Information:

Demo Credentials for admin: [official@admin/1432]

Repository: https://github.com/christinabaiju/Automated-Student-Attendance-Monitoring-and-Analytics-System-for-Colleges.git

10. FUTURE SCOPE & SCALABILITY

10.1 Immediate Enhancements:

Performance optimizations

- Admin passwords are currently stored in plain text (consider hashing for security)
- Roll number input validation can be improved
- Mobile UI tweaks needed for smaller screens
- Add the ability to change profile settings and FAQ's in the Dashboard for effective use.

12.3 Bibliography:

[1] Perplexity Ai

[2] Chatgpt

[3] Copilot

13. TEAM CONTRIBUTIONS:

- *Christina Baiju:* QR_HTMLJS.php, Login_page.php, Attendance_Admin_Dashboard.php, Database (attendance_system), Documentation
- *Arushi Sharma:* Login_page.php, Login_CSS.css, Attendance_Admin_Dashboard.php, Documentation
- *Shambhavi Dadhich:* Database (attendance_system)
- *Deepanshi Dev:* Login_page.php (Data entry - scenarios), Documentation
- *Mehak Arora:* Attendance_Admin_Dashboard.php
- *Rishabh Jha:* Presentation