ABSTRACT

This project presents the development of SmartClass, a secure and efficient web-based attendance management system designed to meet the evolving needs of educational institutions. Built using HTML, CSS, JavaScript, Python (Flask), and MySQL, SmartClass focuses on enhancing convenience, automation, and security in classroom management.

The frontend is developed using HTML, CSS, and JavaScript, ensuring a clean, responsive, and user-friendly interface. On the backend, Flask (Python) handles the application logic, seamlessly integrating advanced features like QR code scanning, face recognition via OpenCV, and real-time data management with MySQL.

SmartClass takes attendance management a step further by verifying both a dynamically generated QR code and the student's face, effectively minimizing the risk of proxy attendance. Teachers have the flexibility to generate QR codes, mark attendance manually when needed, and download detailed reports in PDF or Excel formats using Pandas and ReportLab.

More than just an attendance system, SmartClass introduces innovation into the classroom. Features like live scanning, automated reporting, and planned integration with Power BI for analytics reflect our commitment to modernizing education.

SmartClass isn't just a tool — it's a move toward smarter, safer, and more efficient learning environments. Empowering educators with technology, SmartClass helps bring meaningful change to how student presence is monitored and managed.

Table of Contents

1. INTRODUCTION

1.1 Problem Statement

In India, traditional attendance systems in educational institutions continue to face numerous inefficiencies and challenges. These systems often rely on manual methods such as roll-calling or paper-based registers, which are time-consuming and susceptible to human errors and fraudulent practices like proxy attendance—where one student marks attendance on behalf of another.

Although some institutions have attempted to implement digital alternatives, many still struggle with issues related to security, accuracy, and scalability. There is a noticeable gap in the availability of a secure, fully automated, and scalable solution that can reliably prevent attendance fraud and maintain accurate student records—especially in large or diverse classrooms across urban and rural India.

SmartClass aims to fill this gap by integrating dynamic QR codes and face recognition technology to automate and secure the attendance process. However, a key challenge lies in evaluating the real-world effectiveness of such a system in terms of improving accuracy, efficiency, and security within Indian educational institutions.

1.2 Objective

The primary objective of the SmartClass project is to design and develop a secure, efficient, and scalable attendance management system tailored to the needs of Indian educational institutions. Specifically, the project seeks to:

- Prevent proxy attendance using dynamic QR code scanning combined with facial recognition.
- Eliminate manual errors and ensure accurate student attendance records.
- Streamline the attendance-taking process, saving valuable time for teachers and students alike.
- Provide a scalable solution adaptable to institutions of varying sizes, including schools and colleges in rural and remote regions.
- Deliver a user-friendly interface and seamless experience for both administrators and end-users.
- The overarching goal is to modernize the way attendance is managed, making it more transparent, secure, and efficient.

1.3 Scope

Geographical Scope:

SmartClass is designed for deployment in educational institutions across India, ranging from urban universities to government schools in rural areas.

Technological Scope:
The system incorporates modern technologies such as QR code generation, real-time scanning, facial recognition (OpenCV), and a web-based interface backed by a reliable Flask-Python + MySQL architecture.

Future Scope:
There is scope for integrating AI-based analytics, including student performance tracking, attendance pattern visualization through tools like Power BI, and further enhancements to optimize decision-making in educational administration.

User Scope:
The system targets teachers, students, and administrative staff, offering customized features for each user type to support their respective roles.

1.4 Motivation

The motivation behind SmartClass arises from the ongoing challenges in the Indian education system regarding attendance tracking. Manual systems are still widely used, often resulting in inaccurate data, administrative burden, and academic unfairness due to proxy attendance.

Given the vast student population in India and the increasing need for accountability in academic environments, there's a pressing demand for a transparent, secure, and automated solution.

SmartClass was conceived to meet this demand. By using advanced technologies, the project seeks to eliminate fraudulent practices, simplify attendance workflows, and promote fairness in academic records. This initiative is driven by the vision to transform attendance management into a reliable and efficient process, ultimately contributing to better governance and academic integrity in Indian educational institutions.

2. LITERATURE SURVEY

As educational institutions in India embrace digital transformation, the need for effective, secure, and scalable attendance systems has become increasingly apparent. Traditional roll-call methods are still widespread, particularly in rural and semi-urban regions, but they are vulnerable to proxy attendance, errors, and administrative inefficiencies. To address these challenges, various digital attendance systems have been introduced, each with distinct benefits and limitations. This section explores some of the most commonly implemented systems—QR code-based, biometric fingerprint-based, and RFID-based attendance solutions—and identifies the gaps that SmartClass aims to fill.

2.1 QR Code-Based Attendance Systems

QR code systems are favored in many colleges for their low cost and ease of integration. For instance, Patel et al. (2018) implemented such a system in a Gujarat-based college where teachers generated QR codes and students scanned them to mark attendance. While this method significantly reduced time and administrative workload compared to traditional roll-calls, it had a major flaw: QR codes could be easily shared among students, leading to proxy attendance. Furthermore, the use of static (unchanging) QR codes meant that the system was vulnerable to misuse and offered limited security.

2.2 Biometric (Fingerprint) Attendance Systems

Biometric systems, particularly fingerprint-based ones, are recognized for their accuracy and reliability. A study by Mishra et al. (2017) in a government engineering college in Uttar Pradesh showed a marked improvement in attendance authenticity after deploying fingerprint scanners. These systems are difficult to tamper with and help eliminate impersonation. However, they come with challenges: hardware costs, maintenance needs, hygiene concerns, and reduced usability in resource-constrained environments. Rural areas often face power outages and limited tech support, making these systems less viable outside major urban centers.

2.3 RFID-Based Attendance Systems

RFID systems require students to carry ID cards embedded with chips, which are scanned upon entry. According to Reddy et al. (2019), RFID systems efficiently record attendance and maintain centralized logs. Despite their benefits, they are not foolproof—students can lend cards to others, enabling proxy attendance. The infrastructure demands (e.g., card readers in every classroom) and replacement costs for lost or damaged cards can be burdensome, especially for underfunded institutions.

2.4 Summary of Existing Gaps

While each of the above systems brings improvements over manual methods, none fully address the combined needs of affordability, scalability, security, and accessibility—particularly in the context of India's diverse educational landscape. QR codes are affordable but insecure. Biometric systems are secure but expensive and difficult to maintain. RFID systems offer efficiency but are prone to misuse and logistical challenges.

This underscores the need for a balanced solution that blends security, ease of use, cost-effectiveness, and adaptability to both urban and rural environments.

2.5 Our Contribution – The SmartClass Approach

To bridge these gaps, SmartClass introduces an innovative hybrid system that combines dynamic QR codes with real-time face recognition to prevent proxy attendance and enhance

overall efficiency. The system is designed with India's infrastructural diversity in mind, offering the following key contributions:

- **Dynamic QR Codes:** These regenerate every few seconds, making it nearly impossible for students to reuse or forward codes.
- **Real-Time Face Verification:** Ensures that the student scanning the code is physically present and not impersonating someone else.
- **Manual Mode Backup:** In cases of low internet connectivity or device failure, teachers can switch to manual mode—ensuring reliability and continuity.
- **Hardware Independence:** SmartClass operates on web-enabled devices like smartphones and laptops, eliminating the need for expensive hardware like fingerprint scanners or RFID readers.

2.6 COMPARISION WITH OTHER EXISTING SYSTEMS

Feature	Manual Attendance	Biometric System	RFID-Based System	SmartClass (Proposed)
Proxy Prevention	Susceptible to proxy attendance	Effective prevention	Susceptible via card sharing	High security through QR + face recognition
Hardware Cost	Very low, no hardware required	High (requires biometric scanners)	Medium (RFID cards + readers)	Low (uses smartphones and webcams)
Internet Dependency	Not required	Not required	Not required	Required for full use; manual mode available offline
Real-Time Verification	Not supported	Supported upon fingerprint scan	Supported at card scan	Supported via QR scan and face verification
Rural Deployment	Easily deployable	Challenging due to infrastructure	Challenging due to hardware	Easily deployable with minimal infrastructure
Maintenance Required	Minimal	High (hardware maintenance)	Medium (hardware + software)	Low (software-based, minimal hardware)
Scalability	Difficult in large institutions	Scalable but costly	Scalable but infrastructure-intensive	Highly scalable, low-cost expansion
User-Friendly Interface	Labor-intensive, error-prone	Generally user-friendly; hygiene issues	Requires basic device familiarity	Intuitive for both teachers and students
Dynamic Attendance Security	Absent	Strong but static	Static security; card-based	Strong with dynamic QR and live verification
Backup Attendance Support	Always available	Not available	Not available	Manual mode available in case of failures

SmartClass offers a modern, secure, and scalable solution that overcomes the limitations of traditional and hardware-dependent attendance systems. Its blend of QR and face recognition ensures accuracy, flexibility, and ease of deployment.

3. SYSTEM REQUIREMENTS

To ensure the effective deployment and functioning of the SmartClass Attendance System, the following hardware and software specifications are recommended. These requirements are aimed at providing a smooth user experience for both development and practical implementation in educational institutions.

3.1 Software requirements

Component	Minimum Requirement
Operating System	Windows 10/11, Linux (Ubuntu), or macOS
Programming Language	Python (Flask), JavaScript, HTML, CSS
Backend Framework	Flask (Python-based)
Frontend Technologies	HTML5, CSS3, JavaScript
Database	MySQL
Libraries/Packages	OpenCV, Flask-Login, qrcode, jsQR, Pandas
Web Browser	Google Chrome / Mozilla Firefox (latest version)
Development Tools	Visual Studio Code / PyCharm / MySQL Workbench

3.2 Hardware Requirements

Component	Minimum Requirement
Processor	Intel Core i5 / AMD Ryzen 5 or higher
RAM	8 GB minimum
Storage	250 GB HDD or SSD
Camera	Integrated or external webcam (720p or higher)
Internet Connection	Stable broadband or Wi-Fi
Display	13-inch or larger screen
QR Scanner (Optional)	Smartphone camera / Webcam

4. SYSTEM DESIGN PROCESS

4.1 Use Case Diagram

The Use Case Diagram illustrates the primary interactions between the two main users of the SmartClass system — Students and Teachers. It highlights key functionalities such as face recognition, QR code scanning, attendance checking for students, and QR code generation, manual attendance marking, and report downloading for teachers. This diagram provides a clear overview of system capabilities from the user's perspective.

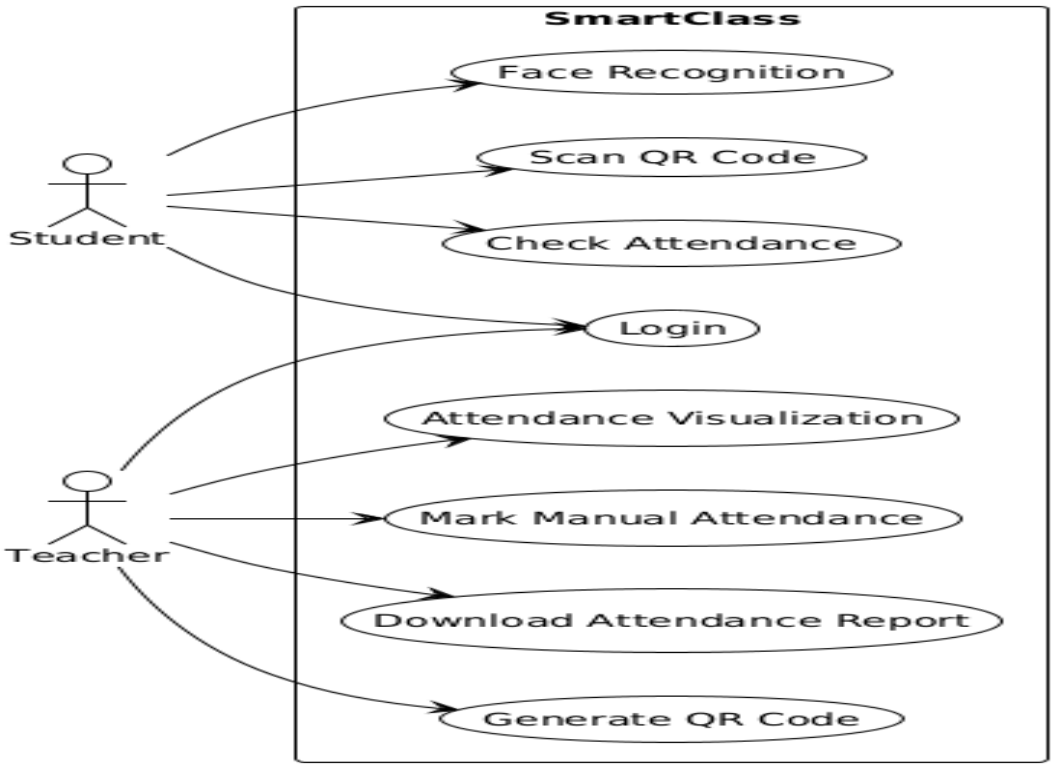


Figure 4.1: Use Case Diagram for SmartClass System

Actors:

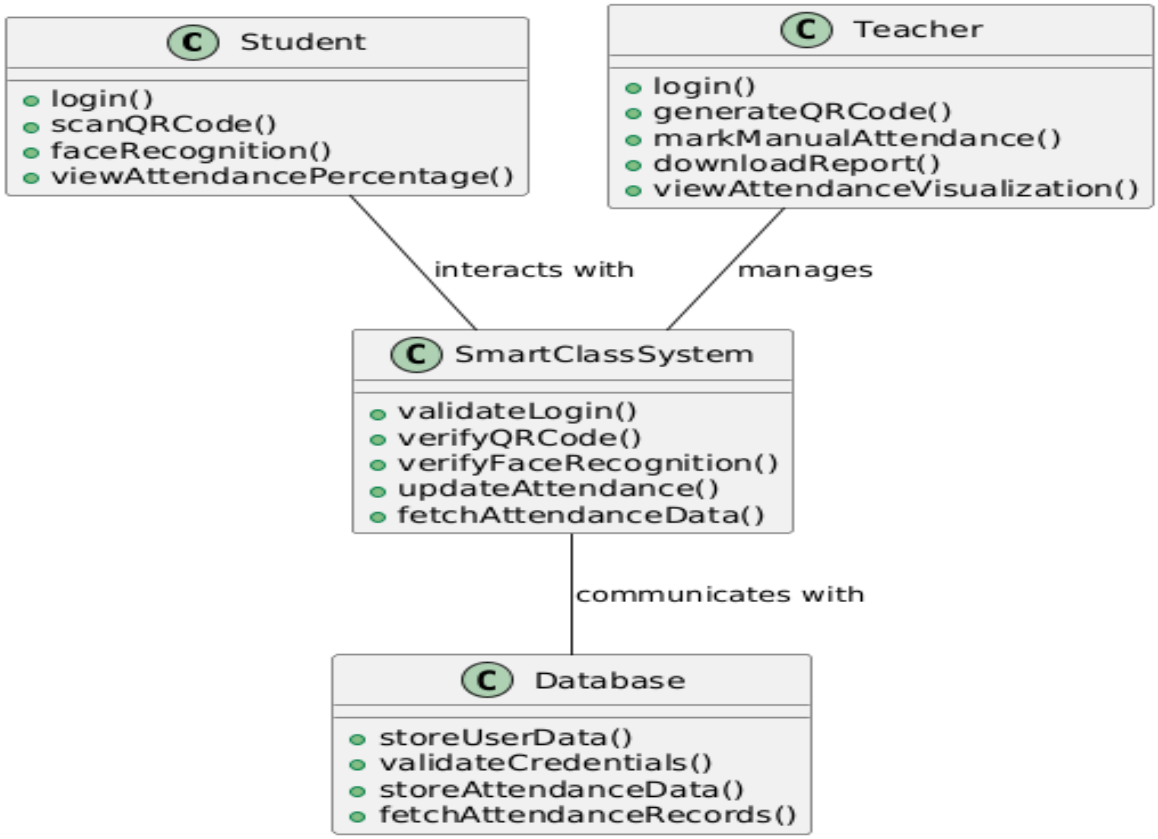
1. Teacher
2. Student

Use Cases:

1. Login
2. Generate QR
3. Scan QR
4. Face Recognition
5. Attendance Visualization
6. Mark Manual Attendance
7. Download Attendance Report
8. Check Attendance Percentage

4.2 Class Diagram

The class diagram illustrates the core components of the SmartClass attendance system and their interactions. Students and teachers access the system through distinct interfaces, performing actions such as login, QR code operations, and attendance management. The SmartClassSystem class handles authentication, verification, and data processing, acting as a bridge between users and the Database. The Database class manages user information and attendance records. This structure ensures secure, efficient, and organized attendance tracking for educational institutions.

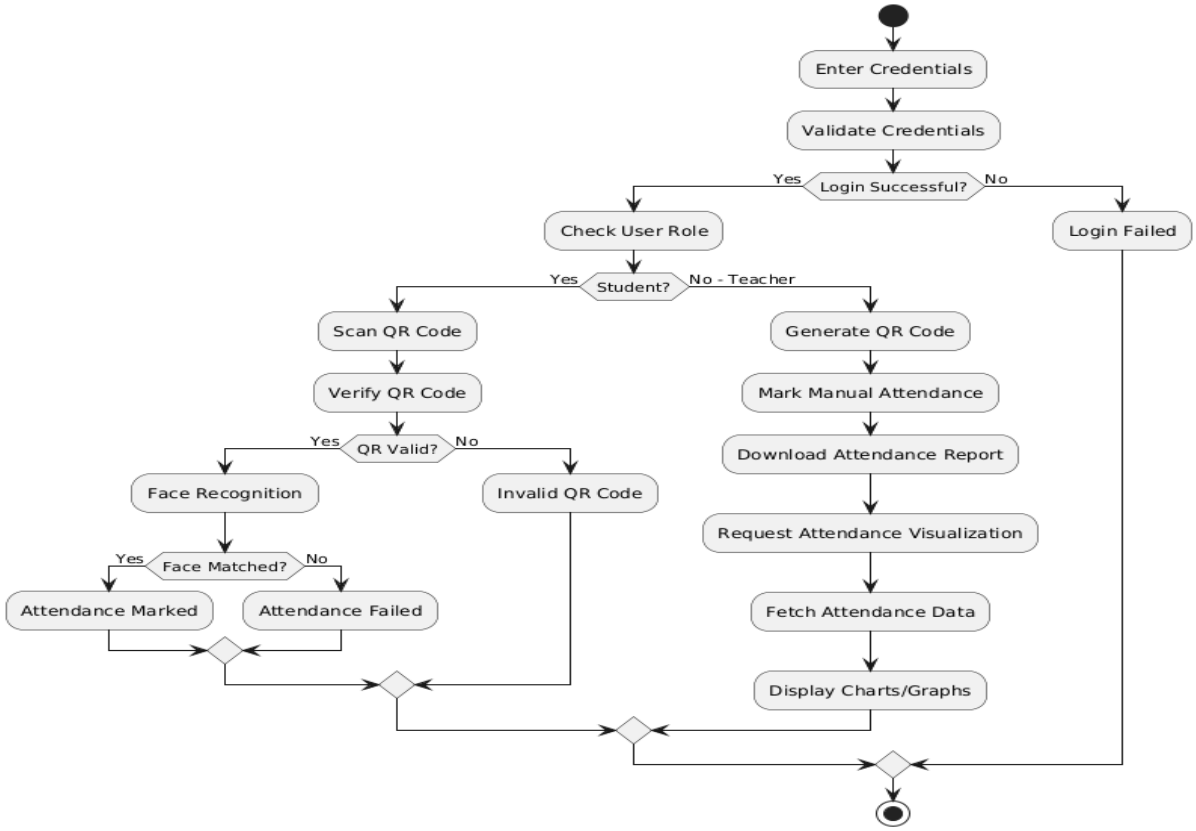


Classes :

1.Student
2.Teacher
3.SmartClass System
4.Database

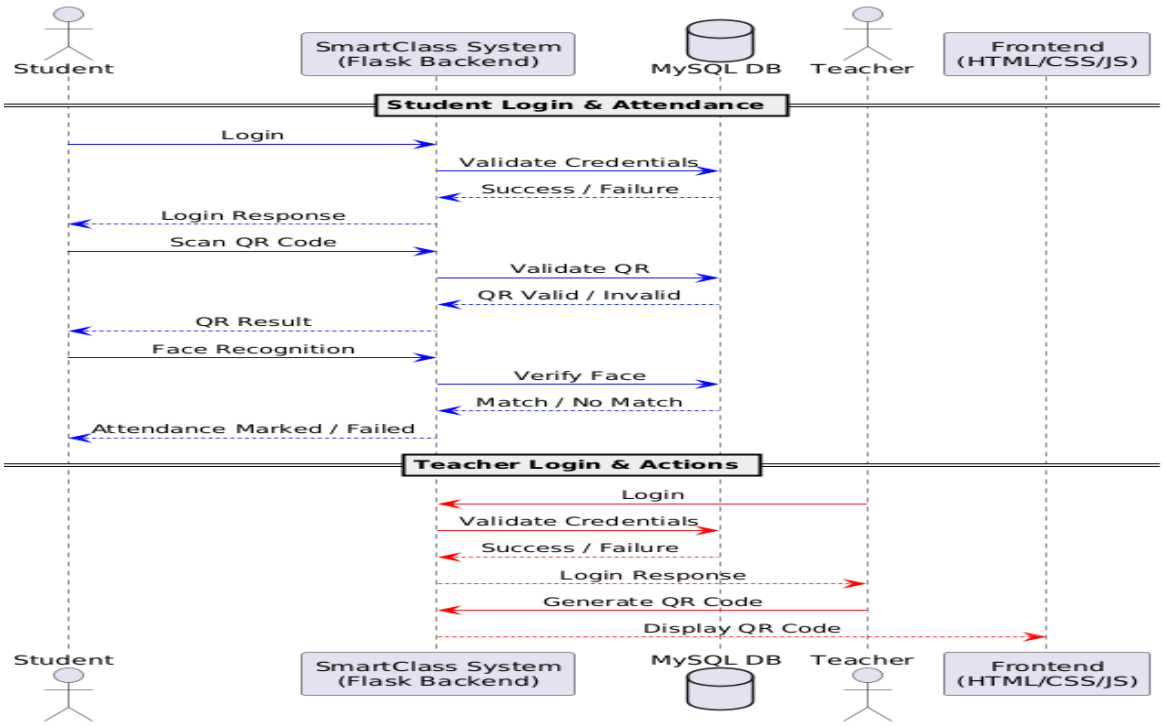
4.3 Activity Diagram

The activity diagram visually represents the workflow of the SmartClass attendance system for both students and teachers. It starts with user login and credential validation, then branches based on user roles. Students follow a path involving QR code scanning and face recognition for attendance marking, while teachers can generate QR codes, mark attendance manually, and access attendance reports and visualizations. The diagram clearly outlines decision points and outcomes, ensuring a streamlined and secure process for managing classroom attendance.

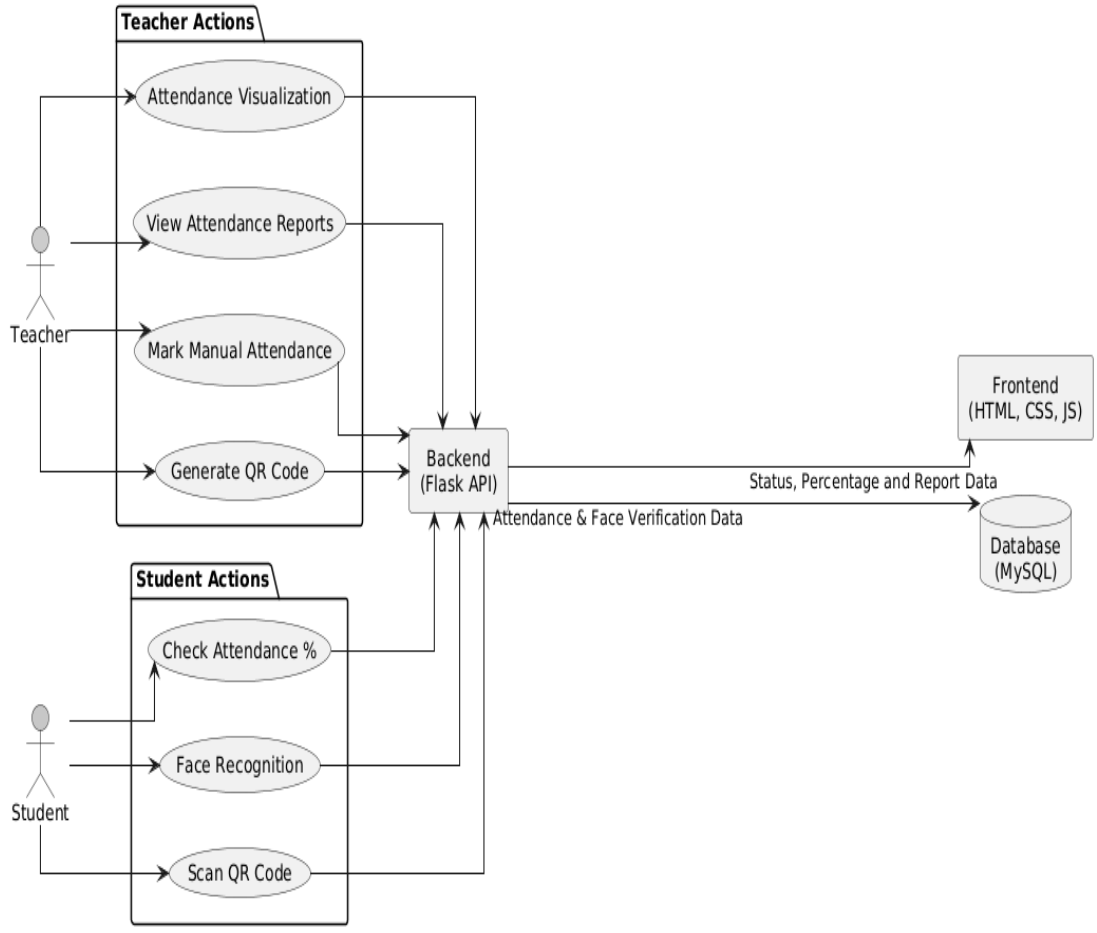


4.4 Sequence Diagram

The sequence diagram depicts the step-by-step interactions between students, teachers, the SmartClass backend system, the MySQL database, and the frontend interface. It details the processes of login, QR code validation, face recognition for students, and QR code generation for teachers. Each message flow illustrates how requests and responses are exchanged among system components to complete attendance and management tasks. This diagram provides a clear visualization of the system's dynamic behavior and the order of operations for both user roles.



4.5 System Architecture



5. SYSTEM MODULES & INTERACTIONS

5.1 System Modules

1. User Authentication

Secure login functionality ensures that only authorized users can access the system. Based on their role—Teacher or Student—users are redirected to their respective dashboards.

- Passwords are securely stored.
- Session management is handled through Flask-Login.
- Prevents unauthorized access to attendance tools and reports.

2. QR Code Module

Teachers can generate dynamic, time-sensitive QR codes that refresh periodically (every 10–15 seconds) to prevent misuse.

- Students must scan the current QR code to proceed.
- Data embedded in each code is unique for every session.

3. Face Recognition Module

Enhances security by integrating OpenCV-based face detection and verification.

- After scanning the QR, students must verify their identity using live face detection.
- Prevents proxy attendance by ensuring the person present matches the stored face data.

4. Attendance Management

Attendance is only recorded when both QR and face recognition are successfully verified.

- Each record is time-stamped.
- Manual attendance option is available for Teachers in case of technical issues.

5. Report Generator

Empowers Teachers to view, filter, and export attendance data in PDF or Excel formats.

- Built using Pandas and ReportLab.
- Enables weekly, monthly, and subject-wise reports.

6. Dashboards and Navigation

Each user type has a dedicated dashboard tailored for their needs:

- **Teacher Dashboard:** Generate QR, mark manual attendance, download reports.
- **Student Dashboard:** Scan QR, verify face, view attendance history.
- Smooth navigation with animated transitions enhances the user experience.

7. Database Integration

All data is securely stored and managed using MySQL.

- User credentials, face encodings, and attendance logs are properly structured.
- Queries are optimized for fast retrieval and scalability.

5.2 Interaction of Modules – System Flow

The SmartClass system follows a clean modular architecture, allowing each component to interact seamlessly across three layers:

Frontend Layer

- Displays dashboards and login interface.
- Manages QR scanning and webcam-based face capture.

- Ensures a responsive, animated user experience.

Backend Logic

- Handles authentication, QR code regeneration, face comparison, and attendance logic.
- Built using Flask and Python libraries.

Database Layer

- Maintains all essential records, including:
  - Registered users
  - Student face data
  - Attendance logs with timestamps
  - QR code metadata

5.3 Example Workflow

1. **Login:** User enters credentials and is authenticated
2. **Dashboard Access:** User is redirected based on their role
3. **QR and Face Verification:** Student scans code and verifies identity
4. **Attendance Marked:** Record is logged in the database
5. **Report Generation:** Teacher views or downloads attendance reports

6. IMPLEMENTATION

6.1 Frontend : HTML,CSS,JS

6.1.1 Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SmartClass Login</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.css">
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
      font-family: Arial, sans-serif;c
    }
    body {
      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background: linear-gradient(135deg, #dbeafe, #bfdbfe);
    }
    .top-bar {
      width: 100%;
      height: 60px;
      background: rgba(255, 255, 255, 0.9);
      backdrop-filter: blur(10px);
      display: flex;
      align-items: center;
      justify-content: space-between;
      padding: 0 20px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      position: fixed;
      top: 0;
      left: 0;
      z-index: 1000;
    }
    .logo-container {
```

```
    display: flex;
    align-items: center;
    gap: 10px;
}
.logo {
    width: 40px;
    height: 40px;
    background: #007bff;
    color: white;
    font-size: 20px;
    font-weight: bold;
    display: flex;
    align-items: center;
    justify-content: center;
    border-radius: 50%;
}
.title {
    font-size: 22px;
    font-weight: bold;
    color: #007bff;
}
.container {
    flex: 1;
    display: flex;
    flex-wrap: wrap;
    width: 85%;
    max-width: 1200px;
    align-items: center;
    justify-content: space-between;
    margin-top: 80px;
}
.hero-section {
    width: 50%;
    padding: 40px;
    text-align: left;
}
.hero-section h1 {
    font-size: 36px;
    color: #0056d2;
    margin-bottom: 10px;
}
.description {
    font-size: 18px;
    margin-bottom: 20px;
    color: #333;
}
.features {
```

```
    list-style: none;
}
.features li {
    font-size: 18px;
    color: #222;
    padding: 6px 0;
}
.features li::before {
    content: "✓";
    color: #007bff;
    font-weight: bold;
    margin-right: 8px;
}
.login-container {
    width: 50%;
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 40px;
}
.login-box {
    width: 100%;
    max-width: 400px;
    padding: 35px;
    background: rgba(255, 255, 255, 0.9);
    backdrop-filter: blur(12px);
    border-radius: 14px;
    box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.2);
}
.login-box h2 {
    text-align: center;
    color: #222;
    margin-bottom: 20px;
}
.input-group {
    position: relative;
    margin-bottom: 15px;
}
.input-group input {
    width: 100%;
    padding: 12px;
    border: 1px solid #ccc;
    border-radius: 8px;
    font-size: 14px;
}
.eye-icon {
    position: absolute;
```

```
            right: 12px;
            top: 50%;
            transform: translateY(-50%);
            cursor: pointer;
            font-size: 14px;
            color: #007bff;
            font-weight: bold;
            user-select: none;
        }
        .login-btn {
            width: 100%;
            padding: 12px;
            background: linear-gradient(135deg, #007bff, #0056d2);
            color: white;
            font-size: 16px;
            border: none;
            border-radius: 8px;
            cursor: pointer;
            transition: 0.3s;
        }
        .login-btn:hover {
            background: linear-gradient(135deg, #0056d2, #0040a3);
            transform: scale(1.02);
        }
        .footer {
            width: 100%;
            text-align: center;
            padding: 10px;
            background: rgba(0, 0, 0, 0.05);
            position: absolute;
            bottom: 0;
            font-size: 14px;
            color: #333;
        }
        .error-message {
            color: red;
            font-size: 14px;
            margin-top: 10px;
            text-align: center;
        }
    </style>
</head>
<body>
    <!-- Top Bar -->
    <div class="top-bar">
        <div class="logo-container">
            <div class="logo">SC</div>
```

```
        <span class="title">SmartClass</span>
      </div>
    </div>

    <div class="container">
      <!-- Hero Section -->
      <div class="hero-section">
        <h1>Stay Accurate with SmartClass</h1>
        <p class="description">A secure and efficient way to manage attendance with:</p>
        <ul class="features">
          <li>QR Code-based attendance</li>
          <li>Face recognition verification</li>
          <li>Manual attendance for reliability</li>
        </ul>
      </div>
      <!-- Login Section -->
      <div class="login-container">
        <div class="login-box">
          <h2>Login to SmartClass</h2>
          <form action="/login" method="POST">
            <div class="input-group">
              <input type="text" name="username" id="username"
placeholder="Username" required>
            </div>
            <div class="input-group">
              <input type="password" name="password" id="password"
placeholder="Password" required>
              <span class="eye-icon" id="eyeText" onclick="togglePassword()">Show</span>
            </div>
            {% if error %}
              <div class="error-message">{{ error }}</div>
            {% endif %}
            <button class="login-btn" type="submit">Login</button>
          </form>
        </div>
      </div>
    </div>

    <!-- Footer -->
    <div class="footer">
      &copy; 2025 SmartClass. All rights reserved.
    </div>

    <script>
      function togglePassword() {
        let passwordField = document.getElementById("password");
        let eyeText = document.getElementById("eyeText");
```

```
            if (passwordField.type === "password") {
                passwordField.type = "text";
                eyeText.textContent = "Hide";
            } else {
                passwordField.type = "password";
                eyeText.textContent = "Show";
            }
        }
    </script>
</body>
</html>
```

6.1.2 teacher_dashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Teacher Dashboard - SmartClass</title>
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=swap"
rel="stylesheet">
    <script src="https://cdn.rawgit.com/davidshimjs/qrcodejs/gh-
pages/qrcode.min.js"></script> <!-- QR Code JS library -->
    <style>
        /* Sidebar Styles */
        .sidebar {
            position: fixed;
            top: 0;
            left: -250px; /* Initially hidden */
            width: 250px;
            height: 100%;
            background-color: #2c3e50;
            color: white;
            transition: all 0.3s ease;
            padding-top: 20px;
        }

        .sidebar-header {
            text-align: center;
            padding: 15px;
            background-color: #34495e;
        }

        .sidebar-menu {
```

```
    list-style-type: none;
    padding: 0;
}

.sidebar-menu li {
    padding: 15px;
    border-bottom: 1px solid #7f8c8d;
}

.sidebar-menu li a {
    color: white;
    text-decoration: none;
    display: block;
}

.sidebar-menu li a:hover {
    background-color: #16a085;
}

/* Main Content */
.main-content {
    margin-left: 0; /* Initially no sidebar visible */
    transition: all 0.3s ease;
}

.top-nav {
    background-color: #2980b9;
    padding: 10px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    color: white;
}

.menu-toggle {
    background: none;
    border: none;
    font-size: 30px;
    color: white;
    cursor: pointer;
}

.top-nav-right {
    display: flex;
    align-items: center;
}
```

```
.top-nav-right span {
    margin-right: 15px;
}

.logout-btn {
    background-color: #e74c3c;
    color: white;
    border: none;
    padding: 10px;
    cursor: pointer;
}

.card {
    border: 1px solid #ddd;
    padding: 20px;
    margin: 10px;
    border-radius: 5px;
    background-color: #ecf0f1;
    box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.1);
    text-align: center;
}

.card h3 {
    margin-bottom: 20px;
}

button {
    padding: 10px;
    background-color: #3498db;
    color: white;
    border: none;
    cursor: pointer;
    margin: 10px 0;
}

button:hover {
    background-color: #2980b9;
}

#qr-box {
    margin: 15px 0;
    display: flex;
    justify-content: center;
    align-items: center;
}

#qrcode {
```

```
    display: inline-block;
}

/* Footer */
footer {
    text-align: center;
    padding: 15px;
    background-color: #34495e;
    color: white;
    position: fixed;
    bottom: 0;
    width: 100%;
}

/* Styles for Attendance Report */
.attendance-report {
    max-height: 400px; /* Set max height for scrollable area */
    overflow-y: auto; /* Allow scrolling */
    padding: 10px;
    margin-top: 20px;
    background-color: #ecf0f1;
    border-radius: 5px;
    box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.1); /* Optional shadow for better visibility */
}

table {
    width: 100%;
    border-collapse: collapse;
}

table, th, td {
    border: 1px solid #ddd;
}

th, td {
    padding: 10px;
    text-align: left;
}

th {
    background-color: #2980b9;
    color: white;
}

/* Ensuring the body or main content scrolls properly */
body, html {
    height: 100%;
```

```
            margin: 0;
            overflow-x: hidden; /* Prevent horizontal scrolling */
        }

        /* Ensure main content can scroll */
        .main-content {
            padding-bottom: 50px; /* Add space for footer */
            overflow-y: auto; /* Allow vertical scrolling */
            height: 100vh; /* Full height */
        }
    </style>
</head>
<body>
    <!-- Sidebar -->
    <div id="sidebar" class="sidebar">
        <div class="sidebar-header">
            <h2>SmartClass</h2>
        </div>
        <ul class="sidebar-menu">
            <!-- Link for 'Mark Manual Attendance' -->
            <li><a href="/mark_manual_attendance">Mark Manual Attendance</a></li>
            <!-- Link for 'View Attendance Report' -->
            <li><a href="#" id="view-attendance">View Attendance Report</a></li>
            <!-- Link for 'Attendance visualization' -->
            <li><a href="#" id="visualize-attendance">Visualize Attendance</a></li>
        </ul>
    </div>

    <!-- Main Content -->
    <div class="main-content">
        <!-- Top Navigation -->
        <nav class="top-nav">
            <button id="menu-toggle" class="menu-toggle">☰</button>
            <div class="top-nav-right">
                <span>Welcome, {{ teacher_name }}</span>
                <button class="logout-btn"
onclick="window.location.href='/logout'">Logout</button>
            </div>
        </nav>

        <!-- Dashboard Content -->
        <div class="content">
            <div class="row">
                <!-- QR Code Box -->
                <div class="card">
                    <h3>Scan to mark your attendance</h3>
                    <div id="qr-box">
```

```
            <div id="qrcode"></div>
          </div>
          <button id="generate-qr" onclick="startQRRefresh()">Generate QR</button>
        </div>

        <!-- Attendance Report Download Box -->
        <div class="card">
          <h3>Download Attendance Report</h3>
          <button class="download-btn" onclick="downloadReport('pdf')">Download
PDF</button>
          <button class="download-btn" onclick="downloadReport('excel')">Download
Excel</button>
        </div>

        <!-- Attendance Report -->
        <div id="attendance-report" class="card attendance-report" style="display:none;">
          <h3>Attendance Report</h3>
          <div id="attendance-table"></div>
        </div>
      </div>
    </div>
  </div>

  <!-- Footer -->
  <footer>
    <p>SmartClass - A Secure Attendance System &copy; 2025</p>
  </footer>

  <!-- JS -->
  <script>
  let qrInterval = null;
  let countdownInterval = null;
  let qrExpiryTime = 10; // seconds

  // Sidebar Toggle
  document.getElementById('menu-toggle').addEventListener('click', function () {
    var sidebar = document.getElementById('sidebar');
    var mainContent = document.querySelector('.main-content');

    if (sidebar.style.left === '-250px') {
      sidebar.style.left = '0';
      mainContent.style.marginLeft = '250px';
    } else {
      sidebar.style.left = '-250px';
      mainContent.style.marginLeft = '0';
    }
  });
```

```
// Attendance Report Load
document.getElementById('view-attendance').addEventListener('click', function () {
    const students = [
        { name: 'Ravi Kumar', rollNumber: '123', attendance: 'Present' },
        { name: 'Sita Rani', rollNumber: '124', attendance: 'Absent' },
        { name: 'Vijay Kumar', rollNumber: '125', attendance: 'Present' },
        { name: 'Priya Rao', rollNumber: '126', attendance: 'Present' },
        { name: 'Irfhan', rollNumber: '127', attendance: 'Absent' },
        { name: 'Jaithra', rollNumber: '128', attendance: 'Present' },
        { name: 'Siddhartha', rollNumber: '129', attendance: 'Absent' },
        { name: 'Nikitha Chowdary', rollNumber: '130', attendance: 'Present' },
        { name: 'Pranav', rollNumber: '131', attendance: 'Absent' },
        { name: 'Ramya', rollNumber: '132', attendance: 'Present' }
    ];

    const reportContainer = document.getElementById('attendance-report');
    const tableContainer = document.getElementById('attendance-table');

    let tableHTML = '<table><tr><th>Student Name</th><th>Roll
Number</th><th>Attendance</th></tr>';
    students.forEach(student => {
        tableHTML +=
<tr><td>${student.name}</td><td>${student.rollNumber}</td><td>${student.attendance}</
td></tr>;
    });
    tableHTML += '</table>';

    tableContainer.innerHTML = tableHTML;
    reportContainer.style.display = 'block';
});

// Visualize
document.getElementById('visualize-attendance').addEventListener('click', function () {
    alert("Coming Soon....");
});

// Dynamic QR Generator
function startQRRefresh() {
    if (qrInterval) clearInterval(qrInterval);
    if (countdownInterval) clearInterval(countdownInterval);

    generateDynamicQR();
    qrInterval = setInterval(generateDynamicQR, qrExpiryTime * 1000);
}

function generateDynamicQR() {
```

```
      const qrBox = document.getElementById('qrcode');
      qrBox.innerHTML = ''; // Clear old QR

      const randomToken = Math.random().toString(36).substring(2, 10); // Simple random
token
      const timestamp = Date.now(); // Current time

      const qrText = SmartClass-${randomToken}-${timestamp};

      new QRCode(qrBox, {
        text: qrText,
        width: 200,
        height: 200,
        colorDark: "#000000",
        colorLight: "#ffffff",
        correctLevel: QRCode.CorrectLevel.H
      });

      // Show expiry countdown
      showCountdown(qrExpiryTime);
    }
    function showCountdown(seconds) {
      const btn = document.getElementById('generate-qr');
      let remaining = seconds;
      btn.innerText = QR valid for ${remaining}s;

      countdownInterval = setInterval(() => {
        remaining--;
        if (remaining <= 0) {
          clearInterval(countdownInterval);
          btn.innerText = "Generating new QR...";
        } else {
          btn.innerText = QR valid for ${remaining}s;
        }
      }, 1000);
    }

    // Report Download
    function downloadReport(format) {
      alert('Downloading ' + format.toUpperCase() + ' report...');
    }
</script>
</body>
</html>
```

6.1.3 mark_manual_attendance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mark Manual Attendance</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    /* General Styling */
    body {
      background: linear-gradient(to right, aqua, skyblue);
      font-family: 'Arial', sans-serif;
      height: 100vh;
      color: white;
      margin: 0;
      padding: 0;
    }

    h2 {
      color: cornflowerblue;
      text-align: center;
      margin-bottom: 30px;
      font-weight: bold;
      font-size: 32px;
      text-transform: uppercase;
      letter-spacing: 3px;
    }

    .attendance-container {
      width: 95%;
      margin: 0 auto;
      padding: 30px;
      background: #fff;
      box-sizing: border-box;
      box-shadow: 0px 10px 30px rgba(0, 0, 0, 0.1);
      border-radius: 10px;
      margin-top: 50px;
      overflow-x: auto;
      max-height: 90vh;
    }

    table {
      width: 100%;
      border-collapse: collapse;
    }
```

```
th, td {
    padding: 15px;
    text-align: center;
    border: 1px solid #ddd;
    font-size: 16px;
    cursor: pointer;
}

th {
    background-color: #2575fc;
    color: white;
    text-transform: uppercase;
}

tbody tr:nth-child(odd) {
    background-color: #f9f9f9;
}

tbody tr:nth-child(even) {
    background-color: #f1f5f9;
}

td:hover {
    background-color: #e2f0ff;
    color: #2575fc;
}

/* Submit and Back Buttons Styling */
.button-container {
    text-align: center;
    margin-top: 20px;
}

.btn-submit, .btn-back {
    background: linear-gradient(45deg, #2575fc, #6a11cb);
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 8px;
    font-size: 16px;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.2s ease;
    display: inline-block;
    margin: 0 10px;
}

.btn-submit:hover, .btn-back:hover {
```

```
    background: linear-gradient(45deg, #6a11cb, #2575fc);
    transform: scale(1.05);
}

.btn-back {
    background-color: #ffc107;
    font-size: 18px;
}

.btn-back::before {
    content: '\2190';  /* Unicode for the left arrow */
    margin-right: 8px;
    font-size: 18px;
}

/* Notification Popup */
.notification-popup {
    position: fixed;
    top: -100px; /* Start off-screen */
    left: 50%;
    transform: translateX(-50%);
    background-color: #dc3545;
    color: white;
    padding: 15px 30px;
    border-radius: 8px;
    box-shadow: 0px 10px 20px rgba(0, 0, 0, 0.2);
    font-size: 18px;
    z-index: 1000;
    opacity: 0;
    visibility: hidden;
    transition: top 0.5s ease, opacity 0.5s ease, visibility 0s 0.5s;
}

.notification-popup.show {
    top: 20px;  /* Position it at the top of the page */
    opacity: 1;
    visibility: visible;
    transition: top 0.5s ease, opacity 0.5s ease;
}

.popup-btn-close {
    background-color: #dc3545;
    color: white;
    padding: 5px 10px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
```

```
        font-size: 14px;
        float: right;
      }

      .popup-btn-close:hover {
        background-color: #c82333;
      }

      /* Responsive Design */
      @media (max-width: 768px) {
        th, td {
          padding: 10px;
        }

        .btn-submit, .btn-back {
          font-size: 16px;
          padding: 10px;
        }
      }

    </style>
</head>
<body>

    <div class="attendance-container">
      <h2>Mark Manual Attendance</h2>

      <!-- Attendance Form -->
      <form id="attendanceForm">
        <div style="overflow-y: auto; max-height: 70vh;">
          <table class="table table-bordered">
            <thead>
              <tr>
                <th>Name</th>
                <th>Roll Number</th>
                <th>Present</th>
              </tr>
            </thead>
            <tbody>
              {% for student in students %}
              <tr class="attendance-row">
                <td class="student-name" aria-label="Student Name">{{ student[1] }}</td>
<!-- Student Name -->
                <td>{{ student[2] }}</td> <!-- Roll Number -->
                <td>
                  <input type="checkbox" name="attendance" data-student-id="{{
student[0] }}" class="attendance-checkbox" aria-label="Attendance Checkbox">
```

```
                    </td>
                  </tr>
                  {% endfor %}
                </tbody>
              </table>
            </div>

            <!-- Submit and Back Buttons -->
            <div class="button-container">
              <button type="button" class="btn-back" onclick="goBack()">← Back</button>
              <button type="submit" class="btn-submit" disabled>Submit Attendance</button>
            </div>
          </form>
        </div>

        <!-- Notification Popup -->
        <div id="notification" class="notification-popup">
          <button class="popup-btn-close" onclick="closePopup()">X</button>
          <span id="notification-message">Please mark at least one student's
attendance.</span>
        </div>

        <!-- Scripts -->
        <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
        <script>
          $(document).ready(function() {
            // Handle form submission
            $('#attendanceForm').on('submit', function(event) {
              event.preventDefault();

              var attendanceData = [];
              var anyChecked = false;

              // Collect attendance data
              $('.attendance-checkbox').each(function() {
                var studentId = $(this).data('student-id');
                var isPresent = $(this).prop('checked');

                if (isPresent) {
                  anyChecked = true;
                }

                attendanceData.push({
                  'student_id': studentId,
                  'present': isPresent
                });
              });
```

```
            if (!anyChecked) {
                // Show the "No attendance marked" notification
                $('#notification-message').text('Please mark at least one student\'s attendance.');
                $('#notification').addClass('show');

                // Hide the notification after 3 seconds
                setTimeout(function() {
                    $('#notification').removeClass('show');
                }, 3000);

            } else {
                // Process and submit attendance if at least one checkbox is marked
                alert('Attendance Submitted!');
            }
        });

        // Handle checkbox selection
        $('.attendance-checkbox').on('change', function() {
            $(this).closest('tr').toggleClass('selected-row');
            toggleSubmitButton();
        });

        // Enable Submit button if attendance has been marked
        function toggleSubmitButton() {
            var anyChecked = $('.attendance-checkbox:checked').length > 0;
            $('.btn-submit').prop('disabled', !anyChecked);
        }
    });

    // Close the notification popup manually
    function closePopup() {
        $('#notification').removeClass('show');
    }

    // Redirect to Teacher Dashboard
    function goBack() {
        window.location.href = '/teacher_dashboard';  // Replace with the actual Teacher
Dashboard URL
    }
  </script>

</body>
</html>
```

6.1.4 student_dashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
 <title>Student Dashboard - SmartClass</title>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/lottie-web/5.9.6/lottie.min.js"></script>
 <script src="https://cdn.jsdelivr.net/npm/jsqr/dist/jsQR.js"></script>
 <script src="https://cdn.jsdelivr.net/npm/chart.js"></script> <!-- Add Chart.js -->
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css"/>
 <style>
  body {
   margin: 0;
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
   background: linear-gradient(to right, #87CEFA, #FFFFFF);
   overflow: hidden;
  }
  .sidebar {
   width: 250px;
   background: #2D3E50;
   color: white;
   padding: 20px;
   position: fixed;
   height: 100%;
   left: -300px;
   transition: left 0.4s ease;
   box-shadow: 5px 0px 15px rgba(0, 0, 0, 0.2);
   z-index: 999;
  }
  .sidebar h2 {
   text-align: center;
   font-size: 22px;
   margin-bottom: 30px;
  }
  .menu-item {
   background: #34495E;
   padding: 12px 15px;
   margin-bottom: 15px;
   border-radius: 8px;
   cursor: pointer;
   display: flex;
   align-items: center;
   gap: 12px;
   transition: 0.3s;
  }
  .menu-item:hover {
```

```
  background: #1ABC9C;
 }
 .toggle-btn {
  position: fixed;
  left: 15px;
  top: 15px;
  font-size: 22px;
  background: #2C3E50;
  color: white;
  border: none;
  padding: 8px 12px;
  cursor: pointer;
  border-radius: 6px;
  z-index: 1000;
 }
 .main-content {
  padding: 40px;
  margin-left: 50px;
  transition: margin-left 0.4s ease;
 }
 .welcome-text {
  font-size: 26px;
  font-weight: bold;
  margin-bottom: 20px;
 }
 #qr-reader-container {
  position: fixed;
  top: 0; left: 0;
  width: 100vw; height: 100vh;
  display: none;
  align-items: center;
  justify-content: center;
  background: rgba(0, 0, 0, 0.6);
  z-index: 2000;
 }
 #qr-reader {
  height: 300px;
  width: 300px;
  border: 3px solid white;
  border-radius: 10px;
  overflow: hidden;
  position: relative;
  background: black;
 }
 #qr-reader .scanner-line {
  position: absolute;
  top: 0;
```

```
      left: 0;
      width: 100%;
      height: 2px;
      background-color: rgba(0, 255, 0, 0.8);
      animation: moveLine 4s linear infinite;
    }
    video {
      width: 100%;
      height: 100%;
      object-fit: cover;
    }
    @keyframes moveLine {
      0% { top: 0; }
      50% { top: 50%; }
      100% { top: 100%; }
    }
    .percentage-display {
      font-size: 24px;
      font-weight: bold;
      color: #2C3E50;
      margin-top: 20px;
      display: none;
      text-align: center;
    }
    .percentage-value {
      color: #1ABC9C;
      font-size: 32px;
    }
    #attendanceChart {
      width: 400px;
      height: 400px;
      margin: 20px auto;
    }
  </style>
</head>
<body>
<!-- Sidebar -->
  <div class="sidebar" id="sidebar">
    <h2>Student Dashboard</h2>
    <div class="menu-item" id="scanQR">
      <i class="fa-solid fa-camera"></i> Scan QR Code
    </div>
    <div class="menu-item" id="requestManualAttendance">
      <i class="fa-solid fa-hand-paper"></i> Request Manual Attendance
    </div>
    <div class="menu-item" id="checkAttendancePercentage">
      <i class="fa-solid fa-chart-line"></i> Check Attendance Percentage
```

```
    </div>
    <div class="menu-item" id="logout">
      <i class="fa-solid fa-sign-out-alt"></i> Logout
    </div>
  </div>

  <!-- Toggle Sidebar -->
  <button class="toggle-btn" id="sidebarToggle">
    <i class="fas fa-bars"></i>
  </button>

  <!-- QR Reader Container -->
  <div id="qr-reader-container">
    <div id="qr-reader">
      <div class="scanner-line"></div>
      <video id="video" autoplay></video>
    </div>
  </div>

  <!-- Main Content -->
  <div class="main-content" id="mainContent">
    <h1 class="welcome-text">Welcome, Student</h1>
    <div class="percentage-display" id="attendanceDisplay">
      <p>Your Attendance Percentage: <span class="percentage-value"
id="attendanceValue">85%</span></p>
      <!-- Doughnut Chart for Attendance Percentage -->
      <canvas id="attendanceChart"></canvas>
    </div>
  </div>

  <script>
    // Sidebar toggle
    document.getElementById("sidebarToggle").addEventListener("click", function () {
      const sidebar = document.getElementById("sidebar");
      const mainContent = document.getElementById("mainContent");

      if (sidebar.style.left === "0px") {
        sidebar.style.left = "-300px";
        mainContent.style.marginLeft = "50px";
      } else {
        sidebar.style.left = "0px";
        mainContent.style.marginLeft = "300px";
      }
    });

    // QR Scanner logic
    let videoElement = document.getElementById('video');
```

```
let qrContainer = document.getElementById('qr-reader-container');
let canvas = document.createElement('canvas');
let context = canvas.getContext('2d');

function startQrScanner() {
  qrContainer.style.display = 'flex';

  navigator.mediaDevices.getUserMedia({ video: { facingMode: "environment" } })
    .then((stream) => {
      videoElement.srcObject = stream;
      videoElement.play();
      requestAnimationFrame(scanQRCode);

      // auto close after 10s
      setTimeout(() => {
        stopQrScanner();
      }, 10000);
    })
    .catch((error) => {
      alert("❌ Camera error: " + error);
    });
}
function scanQRCode() {
  if (videoElement.readyState === videoElement.HAVE_ENOUGH_DATA) {
    canvas.width = videoElement.videoWidth;
    canvas.height = videoElement.videoHeight;
    context.drawImage(videoElement, 0, 0, canvas.width, canvas.height);
    let imageData = context.getImageData(0, 0, canvas.width, canvas.height);
    let code = jsQR(imageData.data, canvas.width, canvas.height);
    if (code) {
      alert("✅ QR Scanned: " + code.data);
      stopQrScanner();
      return;
    }
  }
  requestAnimationFrame(scanQRCode);
}
function stopQrScanner() {
  qrContainer.style.display = 'none';
  let stream = videoElement.srcObject;
  if (stream) {
    stream.getTracks().forEach(track => track.stop());
    videoElement.srcObject = null;
  }
}

document.getElementById("scanQR").addEventListener("click", startQrScanner);
```

```
   document.getElementById("checkAttendancePercentage").addEventListener("click", () =>
{
    document.getElementById("attendanceDisplay").style.display = "block";
    let ctx = document.getElementById('attendanceChart').getContext('2d');
    let attendancePercentage = 85; // Example attendance percentage, replace with dynamic
value
    new Chart(ctx, {
     type: 'doughnut',
     data: {
      labels: ['Attended', 'Not Attended'],
      datasets: [{
       data: [attendancePercentage, 100 - attendancePercentage],
       backgroundColor: ['#1ABC9C', '#e74c3c'],
       borderColor: ['#ffffff', '#ffffff'],
       borderWidth: 2
      }]
     },
     options: {
      responsive: true,
      cutoutPercentage: 70, // Makes the chart look like a doughnut
      plugins: {
       legend: {
        position: 'bottom',
       },
       tooltip: {
        callbacks: {
         label: function(tooltipItem) {
          return tooltipItem.label + ': ' + tooltipItem.raw + '%';
         }
        }
       }
      }
     }
    });
   });

   document.getElementById("requestManualAttendance").addEventListener("click", () => {
    alert("✉ Manual Attendance Request Sent.");
   });
   document.getElementById("logout").addEventListener("click", () => {
    window.location.href = "/logout";
   });
  </script>
</body>
</html>
```

6.1.5 face_recognition.html

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>SmartClass - Face Recognition</title>
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.0/css/all.min.css">
 <style>
  * {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
  }

  body {
   background: linear-gradient(to right, skyblue, white);
   font-family: 'Segoe UI', sans-serif;
   display: flex;
   flex-direction: column;
   align-items: center;
   min-height: 100vh;
   padding-top: 80px;
   color: #333;
  }

  .navbar {
   position: fixed;
   top: 0;
   width: 100%;
   background: #1e3c72;
   padding: 15px 30px;
   color: #fff;
   font-size: 22px;
   font-weight: 600;
   display: flex;
   align-items: center;
   justify-content: center;
   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
   z-index: 999;
  }

  .navbar i {
   margin-right: 10px;
   color: #00f7ff;
  }
```

```
#face-recognition-panel {
  background: hide;
  padding: 30px;
  border-radius: 20px;
  text-align: center;
  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.15);
  width: 370px;
  position: relative;
  animation: slideIn 0.6s ease;
}

@keyframes slideIn {
  from { transform: translateY(30px); opacity: 0; }
  to { transform: translateY(0); opacity: 1; }
}

.menu-item {
  background-color: #1e90ff;
  border: none;
  padding: 12px 22px;
  color: white;
  border-radius: 12px;
  font-size: 16px;
  cursor: pointer;
  transition: background-color 0.3s ease, transform 0.2s;
}

.menu-item i {
  margin-right: 8px;
}

.menu-item:hover {
  background-color: #0d6efd;
  transform: scale(1.05);
  box-shadow: 0 0 10px #0d6efd;
}

#camera-container {
  margin-top: 25px;
  border: 3px solid #1e90ff;
  padding: 18px;

  border-radius: 18px;
  background-color: #f8f9fa;
  width: 280px;
  height: 280px;
  position: relative;
```

```
  overflow: hidden;
  box-shadow: 0 0 15px #1e90ff;
  animation: glow 2s infinite alternate;
}

@keyframes glow {
  from { box-shadow: 0 0 10px #1e90ff; }
  to   { box-shadow: 0 0 25px #0d6efd; }
}

#camera
{
  width: 100%;
  height: 100%;
  border-radius: 12px;
  object-fit: cover;
}

#face-box {
  position: absolute;
  top: 60px;
  left: 60px;
  width: 160px;
  height: 160px;
  border: 3px dashed #1e90ff;
  border-radius: 10px;
  display: none;
  z-index: 10;
}

.hidden {
  display: none;
}

#recognition-text {
  margin-top: 16px;
  font-weight: 600;
  font-size: 16px;
  color: #1e3c72;
}

.popup {
  color: #fff;
  padding: 12px 20px;
  border-radius: 10px;
  position: absolute;
  bottom: -60px;
```

```
  left: 50%;
  transform: translateX(-50%);
  animation: slideUpFade 1.8s ease forwards;
  font-weight: 600;
  box-shadow: 0 5px 15px rgba(0,0,0,0.2);
 }

 @keyframes slideUpFade {
  0% { bottom: -60px; opacity: 0; }
  40% { bottom: 10px; opacity: 1; }
  80% { bottom: 10px; opacity: 1; }
  100% { bottom: -60px; opacity: 0; }
 }

 #cancel-button {
  position: absolute;
  top: 10px;
  right: 10px;
  background: transparent;
  border: none;
  font-size: 18px;
  color: #888;
  cursor: pointer;
 }

 #cancel-button:hover {
  color: #e63946;
 }

 #sound-toggle {
  margin-top: 15px;
  font-size: 18px;
  cursor: pointer;
  color: #1e90ff;
 }

 @media (max-width: 500px) {
  #face-recognition-panel {
   width: 90%;
   padding: 20px;
  }

  #camera-container {
   width: 100%;
   height: auto;
  }
 }
```

```
 </style>
</head>
<body>

 <div class="navbar">
  <i class="fa-solid fa-chalkboard-user"></i> SmartClass - Face Recognition
 </div>

 <div id="face-recognition-panel">
  <h2 style="color: #1e3c72;">Face Recognition Panel</h2>
  <button id="verify-face" class="menu-item">
   <i class="fa-solid fa-camera"></i> Verify Face
  </button>

  <div id="camera-container" class="hidden">
   <button id="cancel-button" title="Close">&#10006;</button>
   <video id="camera" autoplay></video>
   <div id="face-box"></div>
   <p id="recognition-text">Initializing camera...</p>
  </div>

  <div id="popup" class="popup hidden"></div>

  <div id="sound-toggle" title="Toggle sound">
   <i class="fa-solid fa-volume-up"></i> Toggle Sound
  </div>
 </div>

 <!-- Audio elements -->
 <audio id="success-sound" src="https://www.myinstants.com/media/sounds/success-
fanfare-trumpets.mp3"></audio>
 <audio id="fail-sound"
src="https://www.myinstants.com/media/sounds/error.mp3"></audio>

 <script>
  const verifyButton = document.getElementById("verify-face");
  const cameraContainer = document.getElementById("camera-container");
  const camera = document.getElementById("camera");
  const recognitionText = document.getElementById("recognition-text");
  const popup = document.getElementById("popup");
  const cancelBtn = document.getElementById("cancel-button");
  const faceBox = document.getElementById("face-box");
  const successSound = document.getElementById("success-sound");
  const failSound = document.getElementById("fail-sound");
  const soundToggle = document.getElementById("sound-toggle");

  let soundEnabled = true;
```

```
// Toggle sound on/off
soundToggle.addEventListener("click", () => {
  soundEnabled = !soundEnabled;
  soundToggle.innerHTML = soundEnabled ?
    '<i class="fa-solid fa-volume-up"></i> Toggle Sound' :
    '<i class="fa-solid fa-volume-mute"></i> Sound Muted';
});

// Text-to-speech function
function speak(message)
 {
  const speech = new SpeechSynthesisUtterance(message);
  speech.lang = "en-US";  // You can change the language here if needed
  window.speechSynthesis.speak(speech);
}

async function startCamera() {
  try
 {
    const stream = await navigator.mediaDevices.getUserMedia({ video: true });
    camera.srcObject = stream;
  }
  catch (err)
  {
    recognitionText.innerText = "Camera access denied ✕";
    console.error("Error accessing the camera:", err);
  }
}

function stopCamera() {
  const stream = camera.srcObject;
  if (stream)
 {
    stream.getTracks().forEach(track => track.stop());
    camera.srcObject = null;
  }
}

verifyButton.addEventListener("click", async () => {
  cameraContainer.classList.remove("hidden");
  recognitionText.innerText = "Initializing camera...";
  await startCamera();

  faceBox.style.display = "block";

  setTimeout(() => {
```

```
      recognitionText.innerText = "Detecting face...";
    }, 1500);

    setTimeout(() => {
      const isSuccess = Math.random() > 0.3;

      if (isSuccess) {
        recognitionText.innerText = "Face recognized successfully ✅";
        popup.innerText = "Attendance Marked Successfully ✅";
        popup.style.backgroundColor = "#28a745";
        if (soundEnabled) {
          // Use TTS to announce success
          speak("Your attendance has been marked successfully,Thank You");
        }
      }
      else
      {
        recognitionText.innerText = "Face not recognized ❌";
        popup.innerText = "Face Recognition Failed ❌";
        popup.style.backgroundColor = "#dc3545";
        if (soundEnabled) {
          // Use TTS to announce failure
          speak("Face verification failed, Please try again");
        }
      }

      popup.classList.remove("hidden");
      stopCamera();
      cameraContainer.classList.add("hidden");
      faceBox.style.display = "none";

      setTimeout(() => {
        popup.classList.add("hidden");
      }, 1800);
    }, 3200);
  });

  cancelBtn.addEventListener("click", () => {
    stopCamera();
    cameraContainer.classList.add("hidden");
  });
 </script>
</body>
</html>
```

6.2 Backend : Flask (python)

6.2.1 app.py

```
from flask import Flask, render_template, request, redirect, url_for, session, jsonify
import mysql.connector
import random
import string
import time
from flask_cors import CORS

app = Flask(name)
app.secret_key = 'your_secret_key'

# Enable CORS for cross-origin requests (if needed)
CORS(app)

# MySQL Database Connection

db = mysql.connector.connect(
   host='localhost',
   user='root',
   password='12345',
   database='smartclass'
)
cursor = db.cursor()

# Home Route (Redirects to Login)

@app.route('/')
def home():
   return redirect(url_for('login'))

# Login Route
@app.route('/login', methods=['GET', 'POST'])
def login():
   if request.method == 'POST':
      username = request.form['username']
      password = request.form['password']

      cursor.execute("SELECT id, username, role FROM users WHERE username = %s AND
password = %s", (username, password))
      user = cursor.fetchone()

      if user:
         session['user_id'] = user[0]
         session['username'] = user[1]
         session['role'] = user[2]
```

```
            if user[2] == 'teacher':
                return redirect(url_for('teacher_dashboard'))
            elif user[2] == 'student':
                return redirect(url_for('student_dashboard'))
            else:
                return render_template('login.html', error="Invalid role assigned. Contact admin.")
        else:
            return render_template('login.html', error="Invalid credentials. Please try again.")

    return render_template('login.html')

# Teacher Dashboard Route
@app.route('/teacher_dashboard')
def teacher_dashboard():
    if 'user_id' in session and session['role'] == 'teacher':
        teacher_name = session['username']
        return render_template('teacher_dashboard.html', teacher_name=teacher_name)
    return redirect(url_for('login'))

# Student Dashboard Route
@app.route('/student_dashboard')
def student_dashboard():
    if 'user_id' in session and session['role'] == 'student':
        return render_template('student_dashboard.html')
    return redirect(url_for('login'))

# Logout Route
@app.route('/logout')
def logout():
    session.clear()
    return redirect(url_for('login'))

# Generate Unique QR Code (Updated every 10-15 seconds)
@app.route('/generate_qr')
def generate_qr():
    if 'user_id' in session and session['role'] == 'teacher':
        qr_length = 10
        qr_expiry_seconds = 15

        unique_code = ''.join(random.choices(string.ascii_letters + string.digits, k=qr_length))
        timestamp = int(time.time())

        try:
            cursor.execute("""
                REPLACE INTO qr_codes (teacher_id, qr_value, generated_at)
                VALUES (%s, %s, %s)
            """, (session['user_id'], unique_code, timestamp))
```

```
            db.commit()

            print(f"✅ [{time.strftime('%H:%M:%S')}] QR Code Generated: {unique_code}")

            return jsonify({
                'qr_code': unique_code,
                'expires_in': qr_expiry_seconds
            })
        except Exception as e:
            print(f"❌ [{time.strftime('%H:%M:%S')}] QR Generation Error: {str(e)}")
            return jsonify({'error': 'QR generation failed due to server error'}), 500

    return jsonify({'error': 'Unauthorized access'}), 403

# Validate Scanned QR Code

@app.route('/validate_qr', methods=['POST'])
def validate_qr():
    if 'user_id' in session and session['role'] == 'student':
        scanned_code = request.json.get('qr_code')
        current_time = int(time.time())

        try:
            cursor.execute("""
                SELECT teacher_id FROM qr_codes
                WHERE qr_value = %s AND generated_at >= %s
            """, (scanned_code, current_time - 15))
            result = cursor.fetchone()

            if result:
                print(f"✅ [{time.strftime('%H:%M:%S')}] QR Code Validated for Student ID:
{session['user_id']}")
                return jsonify({'status': 'valid'})
            else:
                print(f"❌ [{time.strftime('%H:%M:%S')}] Invalid/Expired QR Scanned:
{scanned_code}")
                return jsonify({'status': 'invalid'})
        except Exception as e:
            print(f"❌ [{time.strftime('%H:%M:%S')}] QR Validation Error: {str(e)}")
            return jsonify({'error': 'Validation error'}), 500

    return jsonify({'error': 'Unauthorized access'}), 403

# Route to fetch students and display them on the manual attendance page

@app.route('/mark_manual_attendance', methods=['GET', 'POST'])
def mark_manual_attendance():
```

```
   if 'user_id' in session and session['role'] == 'teacher':  # Check if the user is logged in and is
a teacher
      if request.method == 'POST':
         attendance_data = request.json.get('attendance')  # Expecting JSON data from
frontend

         try:
            # Mark attendance for each student
            for entry in attendance_data:
               student_id = entry['student_id']
               is_present = entry['present']

               # Insert attendance data into the database
               cursor.execute("""INSERT INTO attendance (student_id, teacher_id, is_present,
date)
                              VALUES (%s, %s, %s, NOW())""", (student_id, session['user_id'],
is_present))
               db.commit()

            return jsonify({'status': 'Attendance marked successfully'})
         except Exception as e:
            print(f"❌ Error marking attendance: {str(e)}")
            return jsonify({'error': f'Error marking attendance: {str(e)}'}), 500

      # If GET request, render manual attendance form
      cursor.execute("SELECT student_id, name, roll_no FROM students")  # Get all students
from the database
      students = cursor.fetchall()

      return render_template('mark_manual_attendance.html', students=students)  # Pass
students data to HTML page

   return redirect(url_for('login'))  # If the user is not logged in, redirect to the login page


# Run the App
if name == 'main':
app.run(debug=True, port=5001)
```

The project was developed using Visual Studio Code, and the output is displayed in a web browser through Live Server. I used HTML, CSS, and JavaScript to create an interactive and user-friendly interface. For the backend, I have implemented Flask (Python) to manage server-side operations and business logic. Flask acts as a bridge between the frontend and the MySQL database, enabling smooth data communication. This combination ensures that all components of the system work together seamlessly. The overall setup provides a secure, responsive, and efficient attendance tracking experience.

7. SCREENSHOTS

The following are the screenshots of the outputs of different frontend codes linked using backend code.

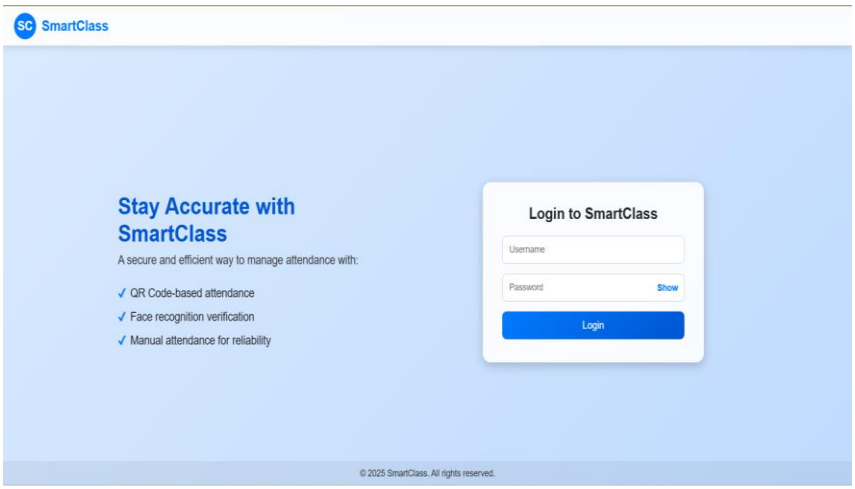


Figure 7.1 : Login page of SmartClass System

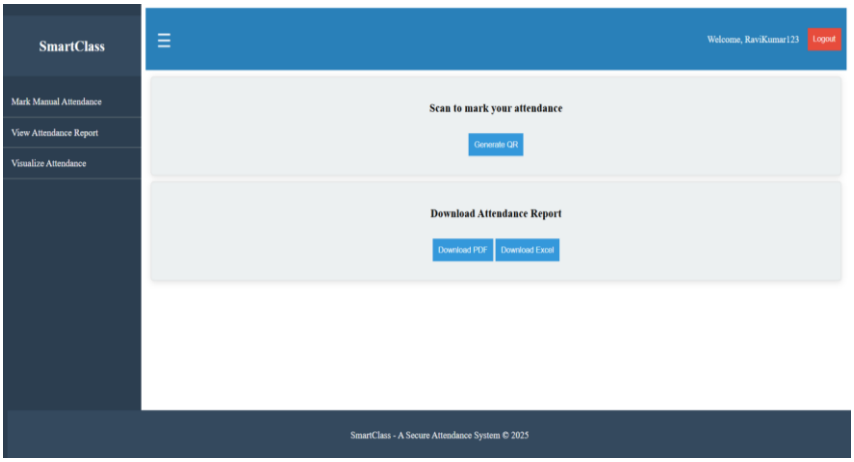


Figure 7.2 : Teacher Dashboard of SmartClass (Interface)

Figure 7.3 : Teacher Dashboard (Manual attendance)

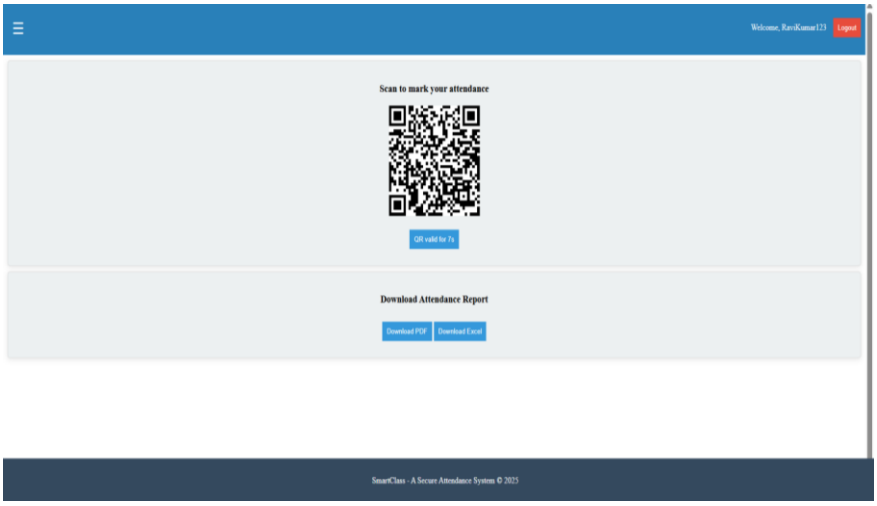


Figure 7.4 : Teacher Dashboard (Dynamic QR generated)

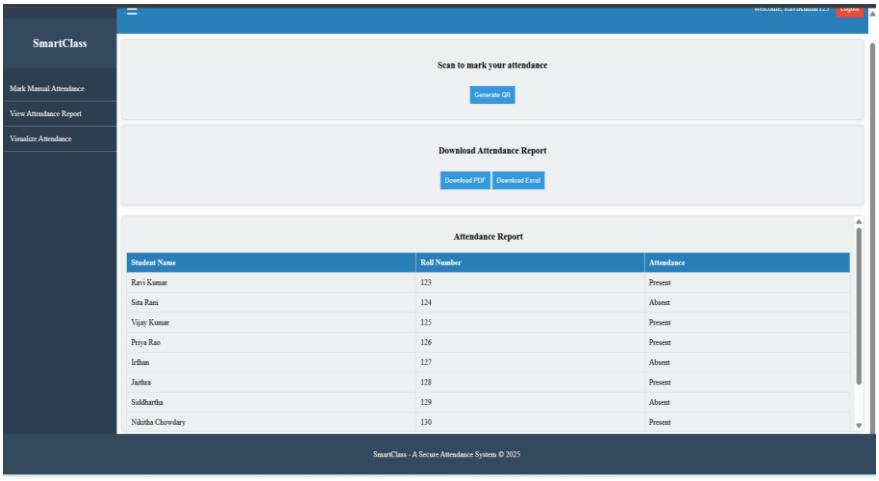


Figure 7.5 : Teacher Dashboard (Attendance Report Generated)

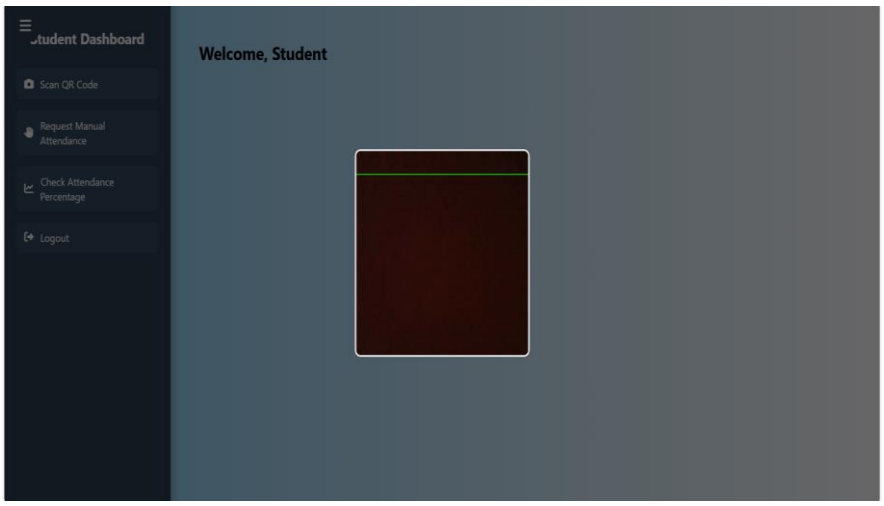


Figure 7.6 : Student Dashboard (QR reader opened)

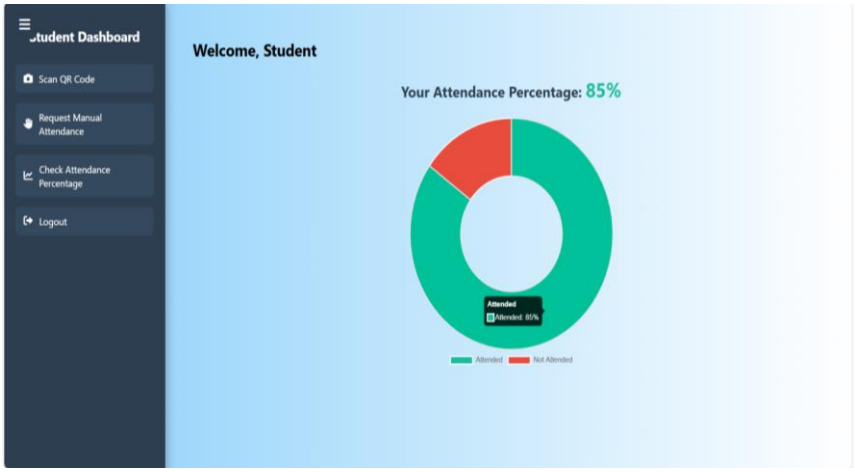


Figure 7.7 : Student Dashboard (Checking Attendance percentage)

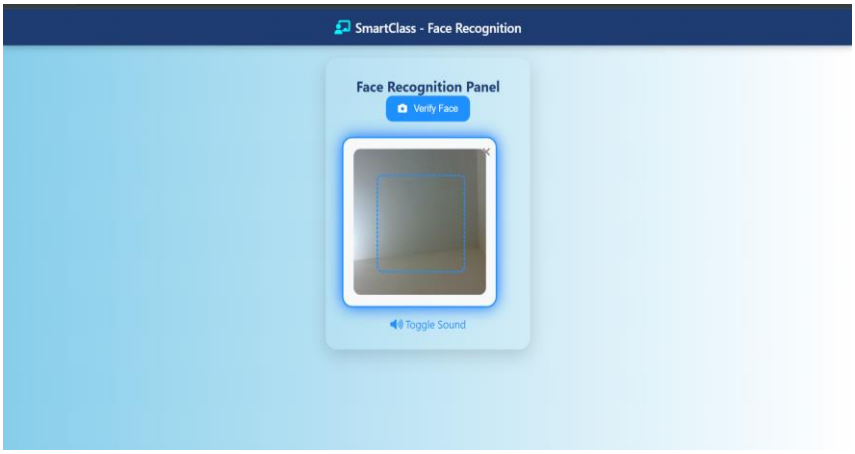


Figure 7.8 : Student Dashboard (Face Recognition)

8. TESTING AND VALIDATION

8.1 Introduction to Testing and Validation

Testing and validation are essential steps to make sure our software works well and does what it's supposed to. Testing helps us catch bugs by checking if the actual results match what we expect across all the features. Validation is about making sure the software actually meets the needs of users and the goals of the project — basically, that we built the right thing. Together, these processes help us deliver a reliable, high-quality, and easy-to-use system. You can see how we're doing in the test status table below.

8.2 Testing and Validation Status Overview

My test status table gives me a clear snapshot of where I currently stand in the project. Core features like login functionality and QR code generation have successfully passed testing, which means those parts are now stable. Some key components, like QR code scanning, are still in progress and under active development. More advanced features, such as face recognition, haven't been implemented yet—mainly due to their complexity and my current focus on higher-priority tasks. This table helps me track my progress and stay organized, ensuring I build a solid and reliable final product.

Test Case	**Expected Outcome**	**Status**
Login Flow	User logs in successfully and is redirected to the right page (Teacher or Student)	Passed
Invalid Login	Displays error message: "Invalid username or password"	Passed
Login UI Elements	All elements like username field, password field, and login button are properly aligned and work correctly	Passed
Buttons on Login Page	The login button submits the credentials and moves the user forward	Passed
QR Generation	A QR code appears with a countdown timer	Passed
QR Expiry	Once expired, the QR code shows an "Expired" message	Passed
Scanner Opening	The camera interface opens up to scan the QR code	Passed
QR Code Scanning	The system recognizes the QR code and navigates to face recognition page	In Progress
Face Recognition Page	The webcam or mobile camera activates to capture the student's face	Passed
Face Recognition Success	Attendance is marked if the face is correctly recognized	Not Implemented
Face Recognition Failure	Shows error: "Face not recognized, please try again"	Not Implemented

9. SMARTCLASS SYSTEM:PROJECT SUMMARY

9.1 Objective

To develop a secure and automated attendance system that leverages dynamic QR codes, facial recognition, and manual attendance options to ensure accurate, tamper-proof, and reliable attendance tracking for educational institutions.

9.2 Key Features

- User authentication system for both teachers and students.
- Dynamic generation of QR codes for attendance marking.
- QR code scanning functionality enabling students to record their attendance.
- Facial recognition technology for identity verification prior to attendance confirmation.
- Generation of comprehensive attendance reports in PDF and Excel formats.

9.3 Technology Stack

Python (Flask), MySQL, OpenCV, HTML/CSS/JavaScript, Pandas, and ReportLab.

9.4 Testing Status

Core functionalities are operational, while some advanced features remain under development.

10.CONCLUSION

SmartClass offers an innovative and secure automated attendance management system specifically designed for educational institutions. By seamlessly integrating dynamic QR code scanning with cutting-edge face recognition technology, it simplifies the attendance marking process while ensuring accuracy and preventing proxy attendance. The system is built with a user-friendly interface, enabling teachers and students to easily interact with features such as real-time QR code generation and instant identity verification. Additionally, SmartClass supports automated generation of detailed attendance reports in multiple formats, helping educators efficiently monitor student participation. Its modular architecture allows for easy scalability and future enhancements, making it suitable for institutions of all sizes. By prioritizing security, usability, and adaptability, SmartClass sets a new standard for attendance management in modern education.

11.FUTURE ENHANCEMENTS

1.AI Based Attendance Prediction

Incorporate machine learning algorithms to analyze attendance trends and proactively predict potential student absences, enabling early intervention and improved student engagement.

2.Mobile App Integration:

Develop a dedicated mobile application for both students and teachers, providing convenient access to attendance records, QR code generation, and real-time attendance management from anywhere.

3.Advanced Face Recognition:

Enhance the face recognition system to increase accuracy and processing speed, particularly in challenging scenarios such as low-light environments or varying facial angles, ensuring more dependable identity verification.

4.Cloud-Based Storage:

Implement cloud storage solutions to securely store attendance data, offering scalable, reliable, and accessible data management across multiple devices and locations.

12.REFERENCES

- Flask Documentation.
  Retrieved from https://flask.palletsprojects.com/

- OpenCV Documentation.
  Retrieved from https://opencv.org/

- MySQL Documentation.
  Retrieved from https://dev.mysql.com/doc/

- qrcode Python Package Documentation.
  Retrieved from https://pypi.org/project/qrcode/

- Flask-Login Documentation.
  Retrieved from https://flask-login.readthedocs.io/

- Pandas Documentation.
  Retrieved from https://pandas.pydata.org/

- ReportLab Documentation.
  Retrieved from https://www.reportlab.com/docs/

- In addition, a variety of scholarly research papers were thoroughly reviewed, covering topics such as QR code-based attendance systems, face recognition technologies, and biometric attendance solutions, providing valuable insights to strengthen the project's foundation.