School of Computing Sciences
Pak-Austria Fachhochschule: Institute of Applied Sciences & Technology
Fall 2025

Class Pulse

Project Report

By

Hassan Ali
Naveed Raza

Under the supervision of

Mr. Ahsan Khan

Chapter 1: Project Vision

Problem Statement

Traditional classroom participation methods lack real-time interaction and engagement. Teachers face challenges in receiving instant feedback from students, and students often hesitate to participate verbally. ClassPulse addresses this by enabling digital interaction, polls, and feedback during lectures.

Business Opportunity

ClassPulse offers a modern classroom communication tool that bridges the gap between teachers and students. It has potential for adoption in universities and schools to enhance student engagement and improve learning outcomes.

Objectives

- Enable real-time student participation using QR-based session joining.
- Allow teachers to collect instant feedback and attendance.
- Provide an easy-to-use interface for both students and teachers.
- Ensure scalability for use in large classrooms.

Project Scope

ClassPulse will focus on providing real-time feedback, live Q&A, and attendance features via web sockets. It will not include advanced analytics or AI grading at this stage.

Constraints

The project is limited by time (one semester), available hardware (laptops only), and team size (two developers).

Feasibility

The project is technically and resource feasible with the planned scope and available time frame.

Chapter 2: Software Requirement Specification

List of Features

- Real-time connection using Socket.IO
- QR code-based session joining
- Live polls and feedback
- Attendance system

Functional Requirements

The system should allow teachers to create sessions, students to join via QR code, and exchange real-time messages or votes.

Non-Functional Requirements

The system must be responsive, reliable, and secure. It should handle multiple simultaneous users without delay.

Chapter 3: System Overview

Architectural Design

The system follows a client-server model. Flask handles backend logic, and Socket.IO provides real-time communication between users.

Data Design

Data is stored using a relational database for session management, student lists, and feedback records.

Chapter 4: System Implementation

The system is implemented using Flask (backend), Socket.IO (real-time communication), and HTML/CSS/JavaScript (frontend). The QR code generation uses Python’s qrcode library. All components are integrated in a lightweight server environment.

Chapter 5: System Testing & Deployment

Testing was performed manually by simulating multiple users joining sessions and sending feedback simultaneously. Deployment was done using a local Flask server, later hosted on a cloud environment for testing.

Chapter 6: Results and Discussion

The system successfully enables real-time participation and feedback collection. Users reported improved engagement and ease of use during testing.

Chapter 7: Conclusion and Future Work

ClassPulse provides a simple, effective tool for improving classroom communication. Future work will include adding analytics, AI-based insights, and mobile app support.

Detailed Description

System Overview

ClassPulse is a real-time classroom feedback system designed to bridge the gap between teachers and students. The system uses modern web technologies to provide a seamless interaction experience. Teachers can initiate a session, generate a QR code for students to join, and receive instant feedback through polls or open responses. Students, on the other hand, can easily connect using their mobile devices without creating accounts, ensuring accessibility and simplicity.

Technology Stack

ClassPulse is built on a lightweight stack to ensure performance and scalability:

- Backend: Flask (Python)
- Real-Time Communication: Flask-SocketIO
- Frontend: HTML, CSS, JavaScript
- Database: SQLite (development), PostgreSQL (deployment)
- Libraries: qrcode, Pillow for QR generation
- Hosting: Render / Heroku for cloud deployment

Architecture and Modules

The system architecture follows a modular client-server approach:
1. Frontend Module: Handles UI and user interactions using responsive web design.
2. Backend Module: Manages business logic, session handling, and socket communication.
3. Database Module: Stores session data, user responses, and feedback securely.
4. QR Module: Generates QR codes for session joining.

5. Analytics Module (future): Will analyze feedback trends and participation metrics.

Testing and Validation

Testing was performed throughout the development process using manual and simulated multi-user scenarios. We conducted stress tests to ensure the server could handle multiple socket connections without lag. Functional testing verified the accuracy of attendance logging and message delivery. Additionally, user feedback from test sessions highlighted that the interface was intuitive and effective for classroom use.

Deployment Environment

The application was deployed on a cloud-based platform using Flask's built-in web server initially and later Dockerized for production. The deployment pipeline included GitHub for version control and automated pushes to Render. Environment variables were used for configuration management, ensuring security and flexibility.

Future Enhancements

While ClassPulse effectively meets its initial objectives, several enhancements can further improve its functionality:

- Adding AI-powered analytics to summarize class engagement levels.
- Implementing push notifications and reminders.
- Introducing mobile applications for Android and iOS.
- Integrating with LMS systems like Moodle or Google Classroom.
- Allowing anonymous question submission for shy students.

Conclusion:

The project successfully demonstrates how real-time feedback can transform the classroom experience. Through ClassPulse, teachers gain actionable insights instantly, and students feel more involved in learning. This system sets a foundation for future research and development in smart classroom technologies.