🎯 Presentation Tool

Interactive Platform for Class Presentations

Technology: React.js | **Integration:** MQTT Real-time Communication | **Platform:** Web Application

📅 Event Details

Date

May 28, 2025

Start Time

2:00 PM

End Time

3:30 PM

Duration

90 Minutes

Teams

6 Teams

Time per Team

15 Minutes

🎯 Core Requirements

🛑 Critical Requirements

MANDATORY

- **Timer Functionality:** Automated countdown and session timing
- **MQTT Communication:** Real-time event broadcasting to grading applications
- **Iframe Integration:** Embedded team demo presentations
- **Team Management:** Support for 6 teams with individual demo URLs

🔍 Technical Specifications

HIGH PRIORITY

- **MQTT Port:** Port 1883 for CS library communication
- **JavaScript Library:** mqtt.js for WebSocket connections
- **Secure Context:** HTTPS/WSS protocol implementation
- **Framework:** React.js for component-based architecture

⏱️ Session Timeline

📋 Presentation Flow

Phase 1: Pre-Event Countdown

Display countdown timer until 2:00 PM start time. Show "3 minutes until Team 1 presents", "2 minutes until Team 1 presents", etc.

Phase 2: Team Presentation (12 minutes)

Automatically load Team X's demo in iframe. Display presentation timer with warning alerts before time expires.

Phase 3: Grading Period (3 minutes)

Transition to grading interface. Send MQTT notification to grading apps. Display QR code for student evaluation access.

Phase 4: Team Transition

Automatically advance to next team. Repeat cycle for all 6 teams (90 minutes total).

🚀 Feature Requirements

✅ Demo Upload System

CORE FEATURE

Teams can specify and upload their demo URLs. Each demo must be embedded within the timer wrapper using iframe technology.

🕒 Warning System

MANDATORY

Audio/visual warnings to alert teams before their presentation time expires. Provides smooth transitions between phases.

🟦 QR Code Integration

ENHANCEMENT

Generate QR codes to connect classmates to the active polling feature

+ Real-time Polling Display

INTEGRATION

Receive and display polling data from the grading application team. Show live during grading periods.

results and feedback during grading periods.

🔗 System Integration

The presentation tool integrates with a companion grading application developed by another team. Polling data is transmitted via MQTT and displayed in real-time during the 3-minute grading windows.

⚙ Technical Specifications

Framework

React.js with functional components and hooks

Communication

MQTT.js library for real-time messaging

Protocol

WebSocket Secure (WSS) for production

Port Configuration

Port 1883 for MQTT broker communication

Deployment

Web-based platform with mobile responsiveness

Integration

Cross-team MQTT communication protocol

📊 Presentation Requirements

📊 Final Presentation Topics

DELIVERABLE

The final presentation must cover three essential topics:

1. **Demo:** Live demonstration of the presentation tool functionality
2. **Technology Used:** Technical architecture and implementation details
3. **AI Involvement:** How artificial intelligence was utilized in the development process

📂 Repository Information

GitHub Repository: https://github.com/AGXeno/t3PresentationTool
Development Environment: https://t1.uvucs.org/
MQTT Testing Tool: MQTTX application for connection testing and debugging

🎯 Success Criteria

Project Acceptance Requirements

- ✅ Automated timer functionality with 15-minute team cycles
- ✅ Seamless iframe integration for team demos
- ✅ MQTT communication with grading application
- ✅ QR code generation for student access
- ✅ Real-time polling data display
- ✅ Warning system for time management
- ✅ Support for 6 teams with individual URLs
- ✅ Responsive design for classroom deployment