

Presentation Tool

Interactive Platform for Class Presentations

Real-time Evaluation & Grading System

Documentation v1.0 - May 2025

Overview

The Presentation Tool is a React-based application designed to manage class presentation sessions with seamless integration between presentation display and real-time student evaluation. It provides automated timing, team management, and live polling results aggregation through MQTT communication with companion grading applications.

Features

Automated Session Management

Seamlessly transitions between countdown, presentation, and grading phases

Team Presentation Display

Embeds team demo applications via iframe with full-screen support

Real-time Evaluation

Students scan QR codes to submit evaluations through companion grading apps

Live Results Aggregation

Displays evaluation results with visual charts and feedback

MQTT Integration

Real-time communication with multiple grading applications

Flexible Timing

Configurable presentation and grading durations with warning alerts

Presentation Management

- **Multiple Teams Support:** Manages up to 6 teams with individual demo URLs
- **URL Management:** Upload and edit team demonstration URLs with persistent storage
- **Manual Controls:** Skip presentation phases or advance timing as needed
- **State Persistence:** Maintains team configurations and settings across sessions

Evaluation Features

- **QR Code Generation:** Automatic QR code generation for easy student access
- **Response Tracking:** Real-time counter showing evaluation submissions
- **Results Visualization:** Interactive charts for clarity, delivery, and confidence scores
- **Feedback Collection:** Displays anonymous student feedback and comments
- **Peer Connectivity:** Shows connected grading applications and response counts

Technology Stack

Component	Technology	Version
Frontend Framework	React	18.2.0
Styling	Custom CSS with CSS Variables	-
Real-time Communication	MQTT (WebSocket)	5.0.5
QR Generation	qrcode.react	3.1.0
Data Visualization	Recharts	2.7.2
Build Tool	Create React App	5.0.1

Installation

Prerequisites

- Node.js (version 14 or higher)
- npm or yarn package manager
- MQTT broker access (WebSocket support required)

Setup Steps

1. Clone the repository

```
git clone [repository-url] cd presentation-tool
```

2. Install dependencies

```
npm install
```

3. Configure MQTT broker

Edit `src/App.jsx` to set your MQTT broker URL:

```
const client = new MQTTClient('wss://your-broker-url:port/mqtt');
```

4. Start the development server

```
npm start
```

5. Build for production

```
npm run build
```

Configuration

Demo vs Live Mode

The application supports two operating modes:

Demo Mode (for testing)

```
const demoMode = true; // Uses current time with short durations //  
Presentation: 30 seconds // Grading: 40 seconds // Warning: 10 seconds
```

Live Mode (for actual presentations)

```
const demoMode = false; // Uses scheduled date: May 28, 2025 at 2:00 PM  
// Presentation: 12 minutes // Grading: 3 minutes // Warning: 1 minute
```

Event Configuration

Modify the `eventConfig` object in `src/App.jsx`:

```
const eventConfig = { teamCount: 6, // Number of teams presentationTime: 12 *  
60 * 1000, // 12 minutes gradingTime: 3 * 60 * 1000, // 3 minutes  
warningTime: 60 * 1000 // 1 minute warning };
```

Team Setup

Teams can be configured in the `teams` state or through the UI:

```
const teams = [ { id: 1, name: 'Team 1', demoUrl: 'https://team1-demo.com' },  
  { id: 2, name: 'Team 2', demoUrl: 'https://team2-demo.com' }, // ...  
  additional teams ];
```

Usage

Starting a Presentation Session

1. Pre-Event Setup

- Configure team demo URLs using the "Upload Team URL" button
- Verify MQTT broker connectivity
- Ensure grading applications are connected

2. Countdown Phase

- Displays countdown until presentation start time
- Shows connected grading applications
- Allows team URL management

3. Presentation Phase

- Automatically displays team demo in iframe
- Shows presentation timer with warnings
- Option to manually skip to grading phase

4. Grading Phase

- Displays QR code for student evaluation access
- Shows real-time response counter
- Automatically displays results when minimum responses reached
- Shows evaluation scores and student feedback

Manual Controls

- **Skip to Grading:** Click "Skip to Grading Portion" during presentations

- **Show Results:** Force display of results in demo mode
- **Team URL Management:** Upload or modify team demonstration URLs
- **Toggle Team URLs:** Show/hide team URL list during countdown

MQTT Integration

Topic Structure

The application subscribes to and publishes on various MQTT topics:

Topic	Purpose
scores/#	Individual evaluation scores
summary/#	Aggregated evaluation summaries
presence	Grading application connectivity
scores/broadcast	Broadcast score distribution
team_reset	Team change notifications

Message Format

Individual Score

```
{ "studentId": "student123", "clarity": 8.5, "delivery": 9.0, "confidence": 7.5, "feedback": "Great presentation!", "timestamp": "2025-05-28T14:15:30.000Z" }
```

Summary Update

```
{ "averages": { "clarity": 8.2, "delivery": 8.7, "confidence": 8.0 }, "submittedCount": 15, "submissions": [...] }
```

File Structure

```
src/ ├── components/ | | ├── CountdownDisplay.jsx # Long countdown display | | ├── Timer.jsx # Standard timer component | | ├── PresentationDisplay.jsx # Team demo  
iframe | | ├── PollingDisplay.jsx # Evaluation results | | ├── QRCodeDisplay.jsx #  
QR code generation | | └── UrlUploadModal.jsx # Team URL management |  
services/ | └── MQTTClient.js # MQTT communication | └── utils/ | | └──  
timeUtils.js # Time formatting utilities | └── styles/ | | └── App.css #  
Application styling | └── App.jsx # Main application logic
```

Companion Applications

This tool is designed to work with companion grading applications that:

- Connect via MQTT to the same broker
- Provide student evaluation interfaces
- Submit scores in the expected format
- Support broadcast and peer-to-peer communication

Troubleshooting

Common Issues

MQTT Connection Failed

- Verify broker URL and WebSocket support
- Check firewall and network connectivity
- Ensure broker allows anonymous connections

Team URLs Not Loading

- Verify URLs include proper protocol (https://)

- Check if target sites allow iframe embedding
- Test URLs independently in browser

Grading Responses Not Appearing

- Confirm grading applications are connected
- Check MQTT topic subscriptions
- Verify time synchronization between applications

Timer Issues

- Check system time accuracy
- Verify demo vs live mode configuration
- Ensure proper date format in event configuration

Debug Information

Enable demo mode for additional debugging:

- Status bar shows current state and timing
- Console logs detailed MQTT message handling
- Response counters and connection status visible

Browser Compatibility

- Chrome/Chromium (recommended)
- Firefox
- Safari
- Edge

Note: Audio warning features require user interaction to enable on some browsers.

Production Deployment

1. Set `demoMode = false` in `App.jsx`
2. Configure production MQTT broker
3. Build the application: `npm run build`
4. Deploy the `build` folder to your web server
5. Ensure WebSocket support for MQTT connectivity

License

This project is private and intended for educational use.