IT643 Software Design and Testing

Date: December 05, 2025

Project:- PollSync Report

(Real-time voting system)

Live Link: https://instant-pulse.vercel.app/

Group Details:-

Jaymin Dave - 202412016

Kashish Jain - 202412025

Harsh Mehta - 202412048

Hiten Thakkar - 202412115

Instructor: Prof. Ankush Chander

Introduction

PollSync is a dynamic and intuitive web application designed to revolutionize real-time classroom interactivity. In fast-paced learning environments, keeping the audience engaged and gathering instant feedback is crucial. PollSync empowers educators, presenters, and event organizers to effortlessly create engaging polls and collect real-time responses from their audience.

The system is built as a robust full-stack solution, featuring a secure **Node.js/Express backend** for managing data and a responsive **React/TypeScript frontend** for a seamless user experience. By utilizing WebSockets, the platform ensures that vote updates are broadcasted instantly, fostering dynamic discussions and immediate insights.

Objectives of the Project

The primary objectives of the TeachingAid system are:

- **Enhance Engagement:** To provide a tool that breaks the monotony of traditional lectures by introducing interactive elements.
- **Real-Time Feedback:** To enable instructors/Organizers to gauge students/audience understanding instantly through live polls.
- **Ease of Access:** To remove barriers to participation by allowing students/audience to vote via a simple link or QR code without needing to register or log in.
- **Secure Management:** To provide a secure environment for professors to manage their content (polls, quizzes, and folders) using authentication.
- **Data Organization:** To allow users to categorize polls into folders and quizzes for better content management.

Tech Stack

The project utilizes a modern, scalable technology stack:

Frontend (Client-Side)

- **Framework:** React.js (with TypeScript)
- **Build Tool:** Vite
- **Styling:** Tailwind CSS & Shadcn UI
- **State Management:** React Query (@tanstack/react-query)
- **Real-Time Communication:** Socket.IO Client
- **Visualization:** Recharts (for analytics charts)

Backend (Server-Side)

- **Runtime:** Node.js
- **Framework:** Express.js
- **Database:** MongoDB (with Mongoose ODM)
- **Authentication:** JSON Web Tokens (JWT) & bcryptjs
- **WebSocket Engine:** Socket.IO

System Functionalities

For Poll Creators

- **User Authentication:** Secure registration and login functionality.
- **Dashboard:** A central hub to view active polls, past history, and folders.
- **Poll Creation:** Ability to create single-choice or multiple-choice questions.
- **Folder Management:** Organize related polls into specific folders (e.g., "Math 101", "Physics").
- **Quiz Generation:** Create multi-question quizzes that generate a sequence of polls.
- **Live Controls:** Start, Stop, and Reset polls in real-time.
- **Visual Analytics:** View live bar charts updating as students vote.

For Poll Participants

- **No-Login Access:** Join polls instantly using a unique 6-digit code or QR code.
- **Real-Time Voting:** Submit answers via a mobile-responsive interface.
- **Duplicate Prevention:** Browser fingerprinting ensures a student cannot vote multiple times on the same poll.

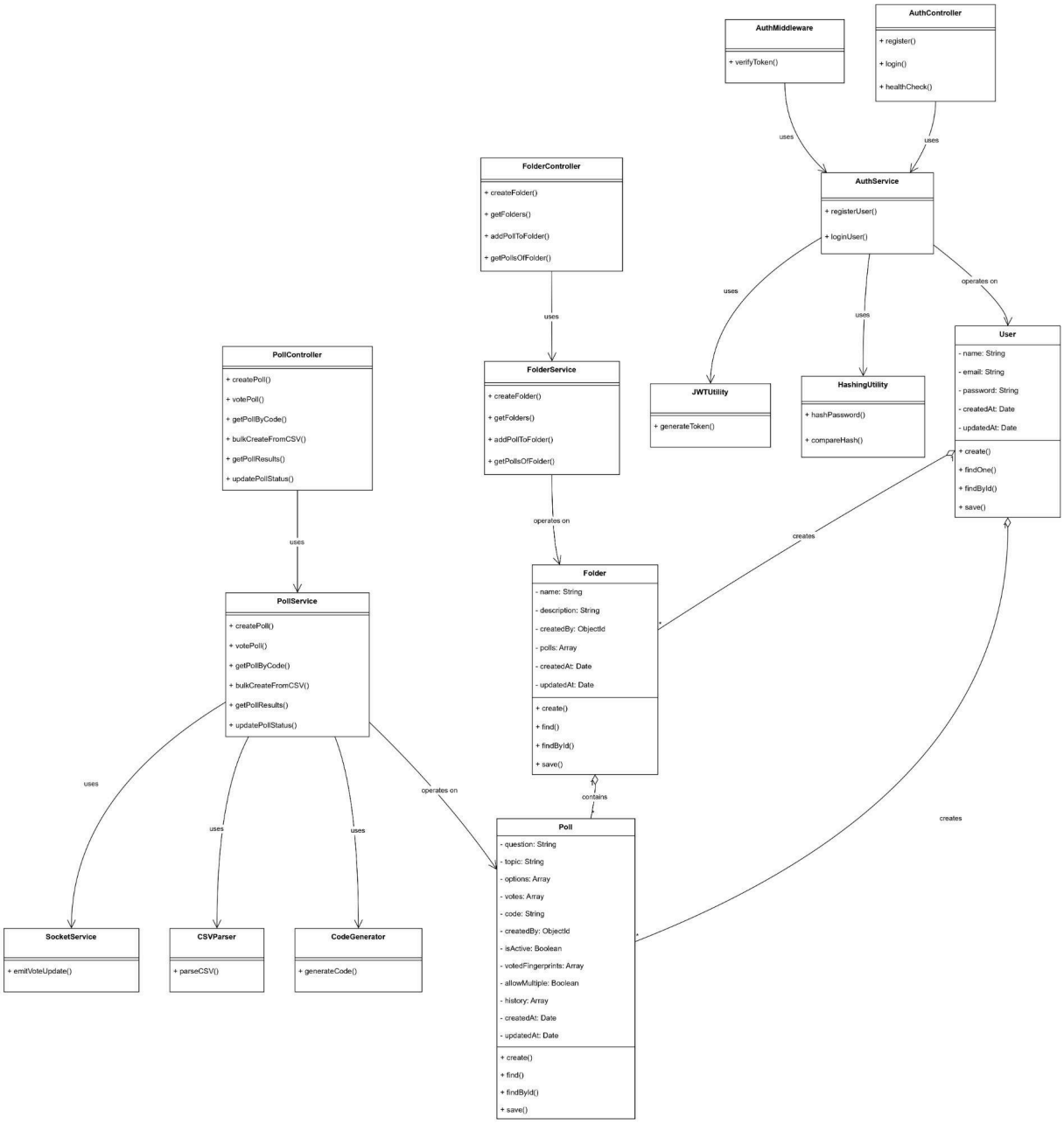
Conclusion

TeachingAid successfully bridges the gap between presenters and audiences by leveraging modern web technologies. By simplifying the feedback loop, it transforms passive listening into active participation. The modular architecture ensures the system is maintainable and ready for future scaling, making it a valuable tool for any educational setting.

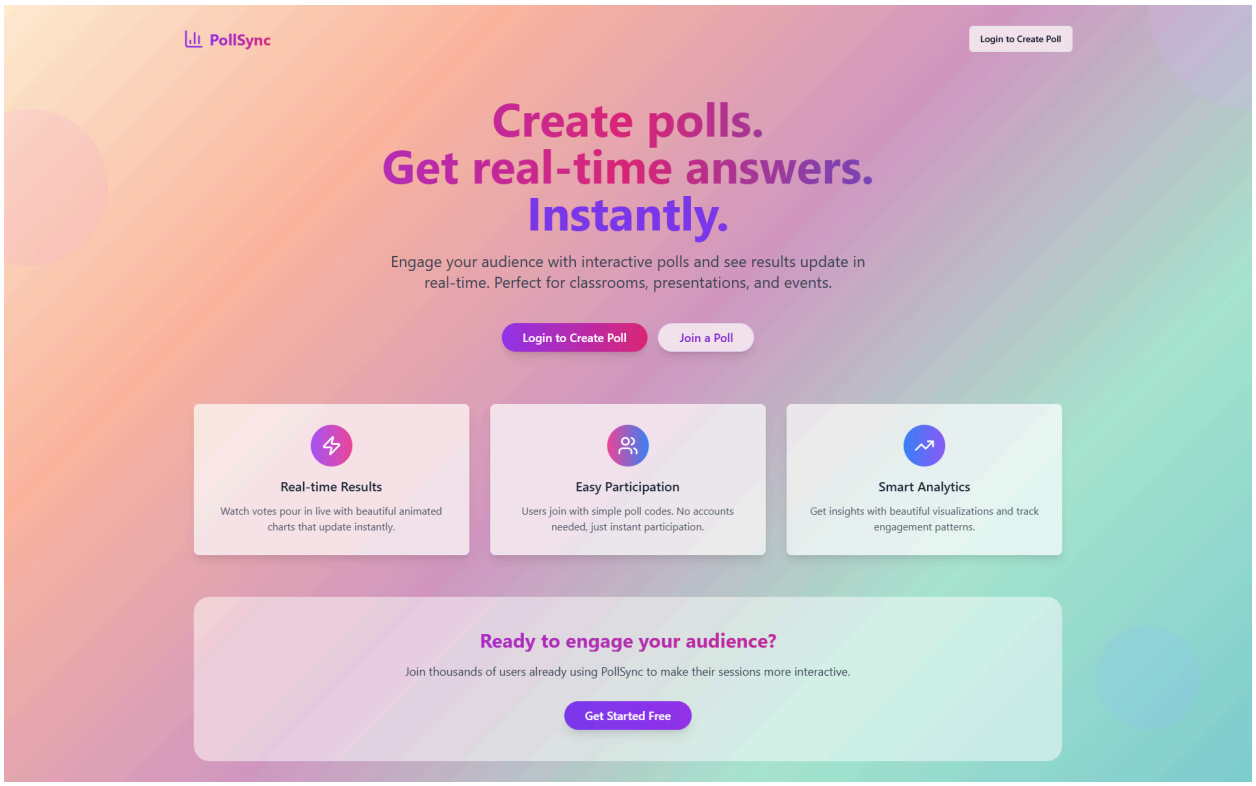
Links & Repositories

- **Backend Repository:** https://github.com/19harshmehta/TeachingAid-Backend
- **Frontend Repository:** https://github.com/19harshmehta/TeachingAid-FrontEnd
- **Live Link:** https://instant-pulse.vercel.app/

Class Diagram



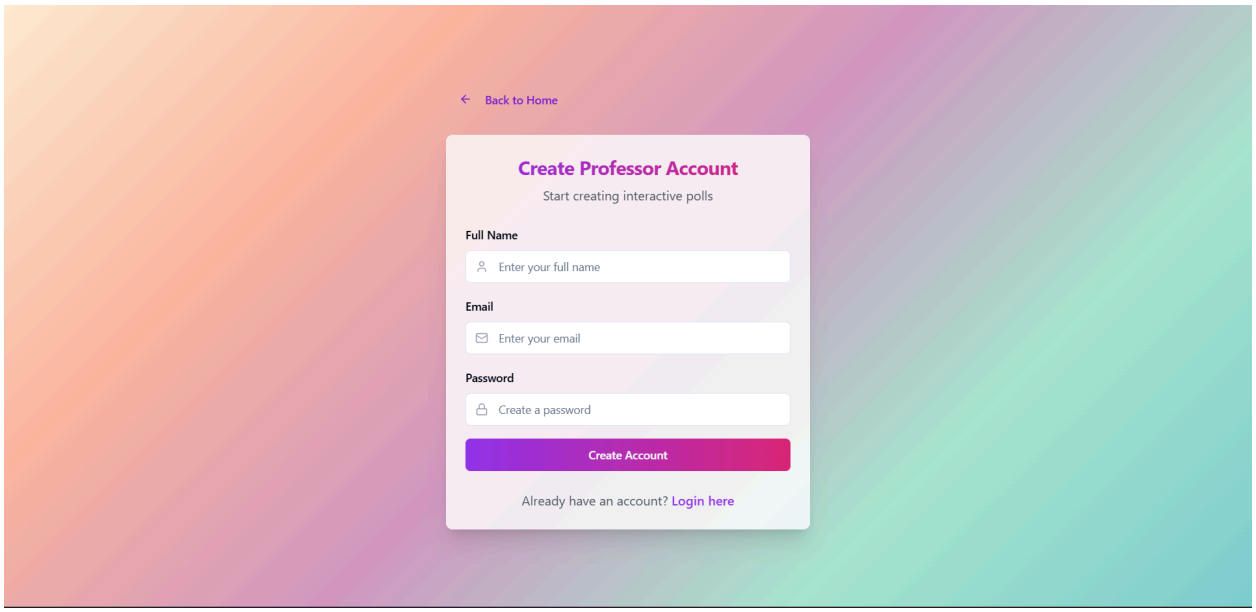
Screenshots



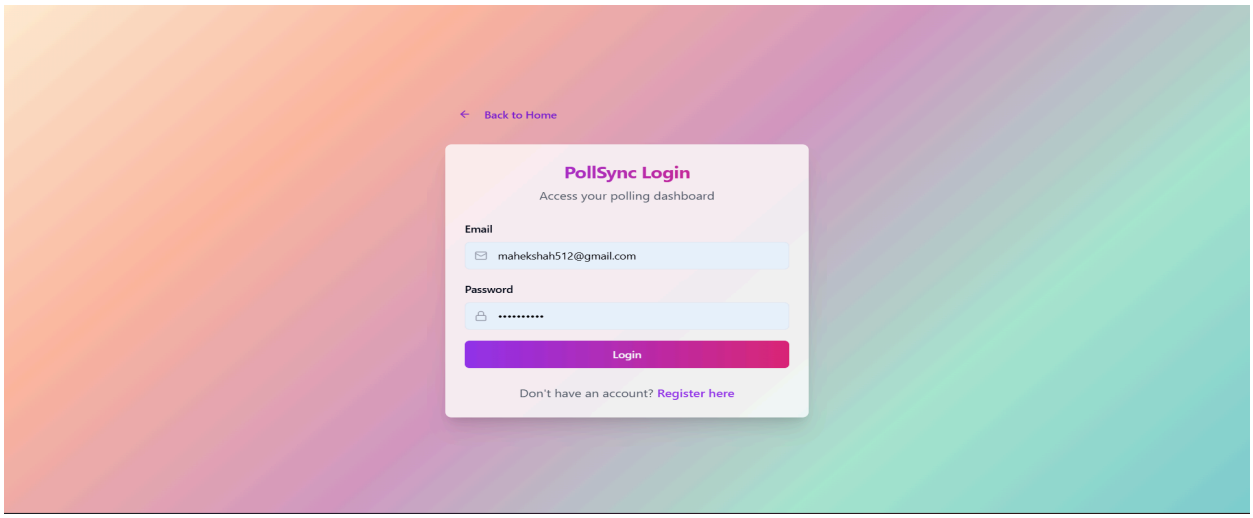
1. User Authentication (Login & Registration Screen)

The application implements full user authentication.

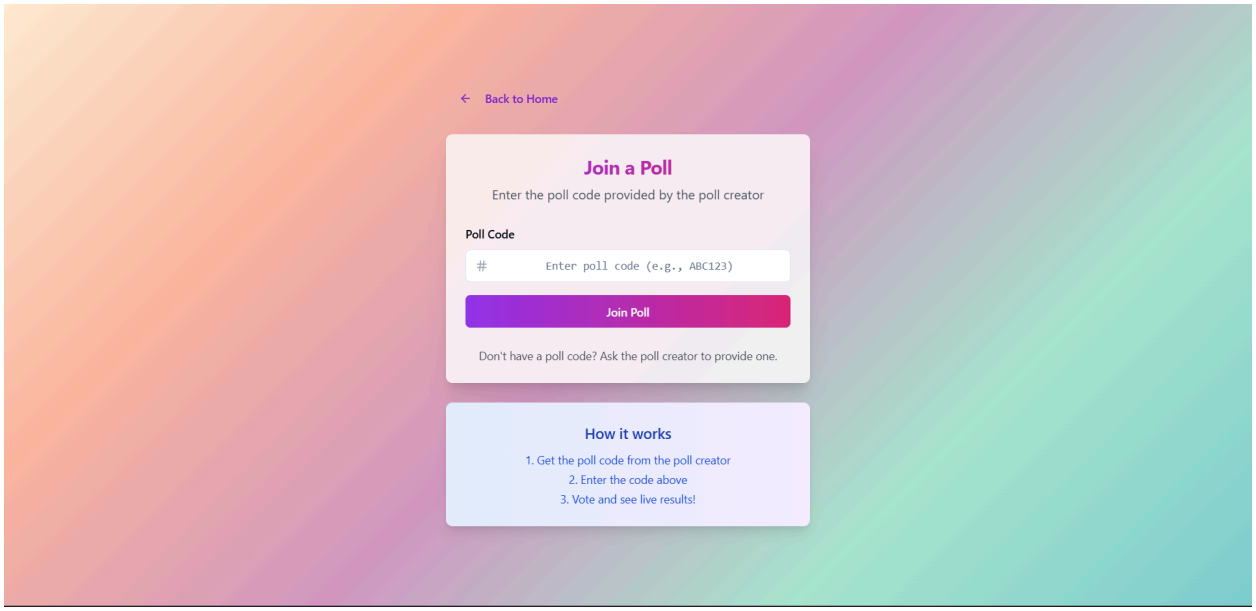
- Registration Screen: Allows a new user to create an account by providing credentials (Name, Email, Password).



- Login Screen: Authenticates existing users via their registered Email and Password, issuing a JWT for secure access.



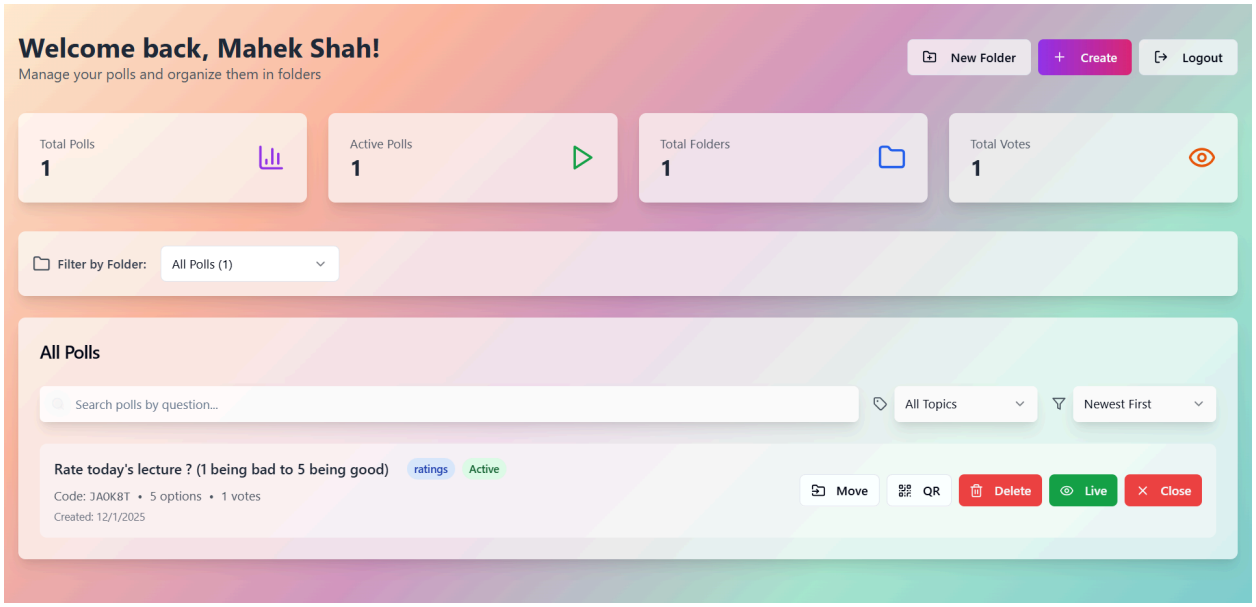
- Access: Join polls using a unique code without requiring registration, ensuring quick, anonymous participation.



2. Dashboard

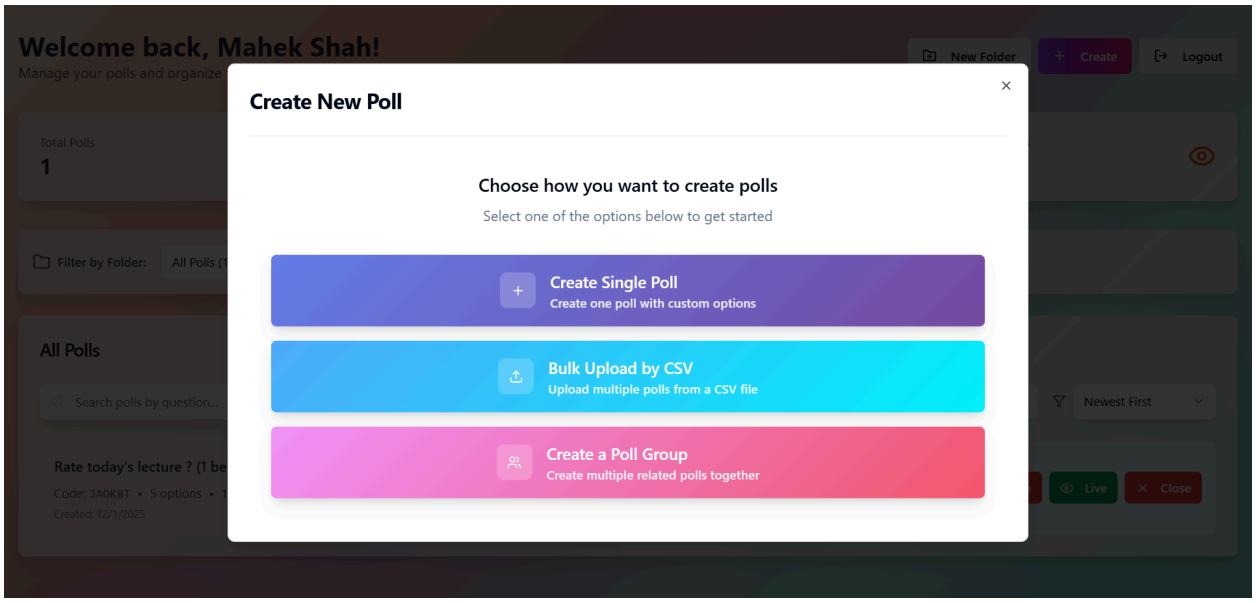
This serves as the central hub for authenticated users ( ) to manage their content.

- Content Display: Displays a comprehensive list of all polls created.

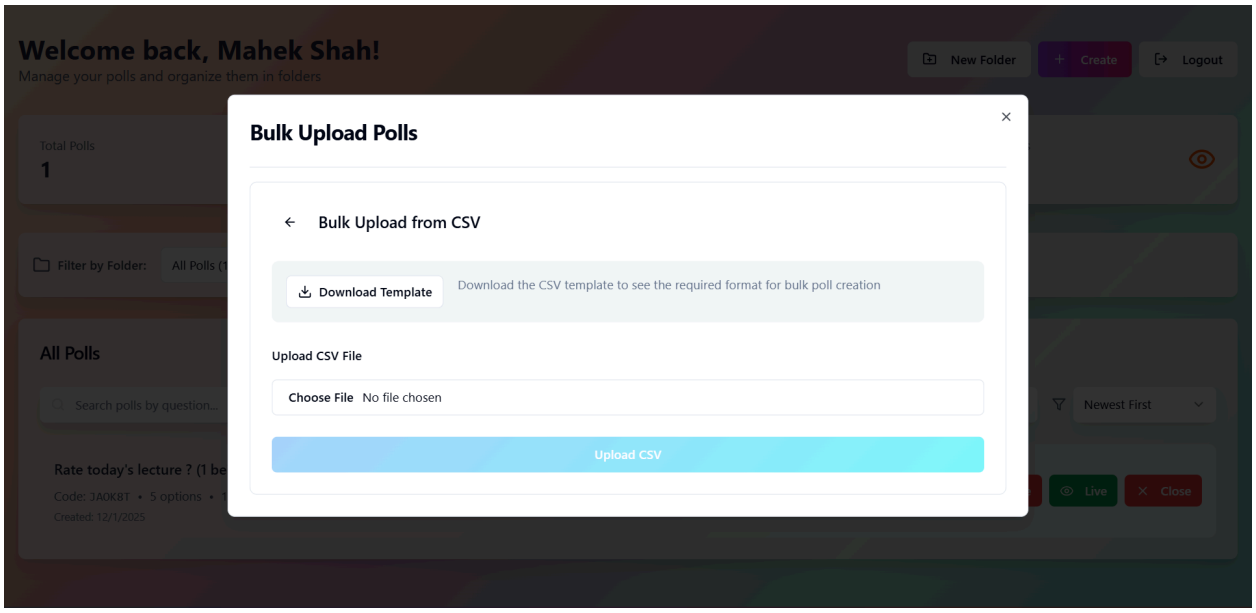


- Actionable Items: Provides immediate access to create a new poll.

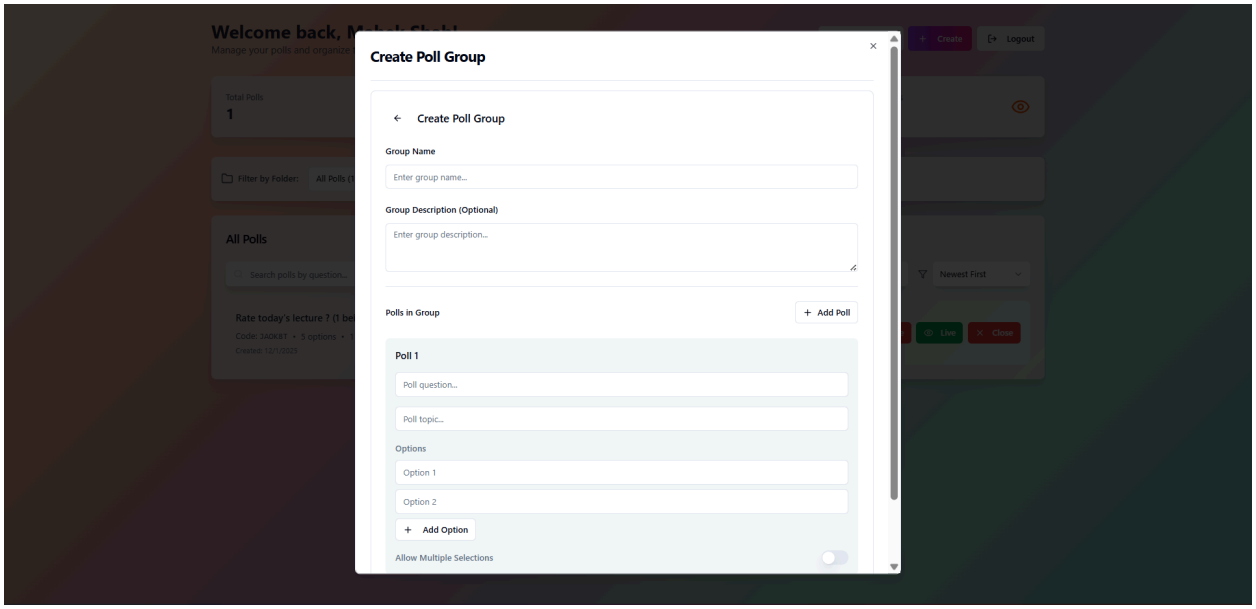
Single poll



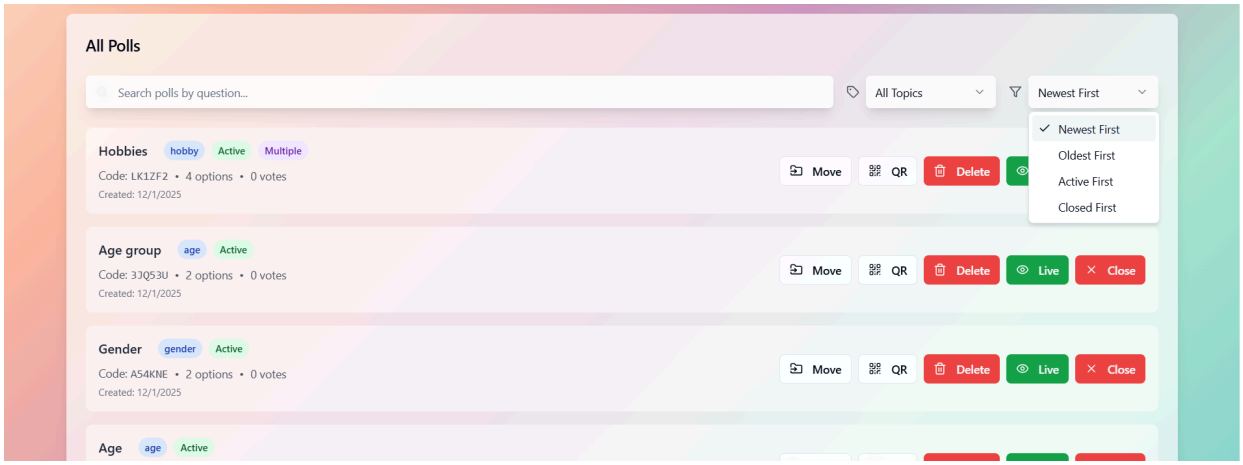
Bulk poll creation via CSV



Create poll group at once



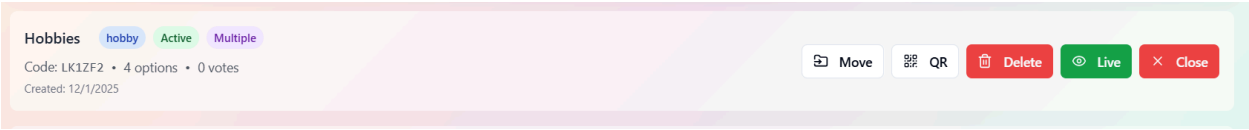
- Folder View: Allows filtering and viewing polls organized within specific folders for better content management.

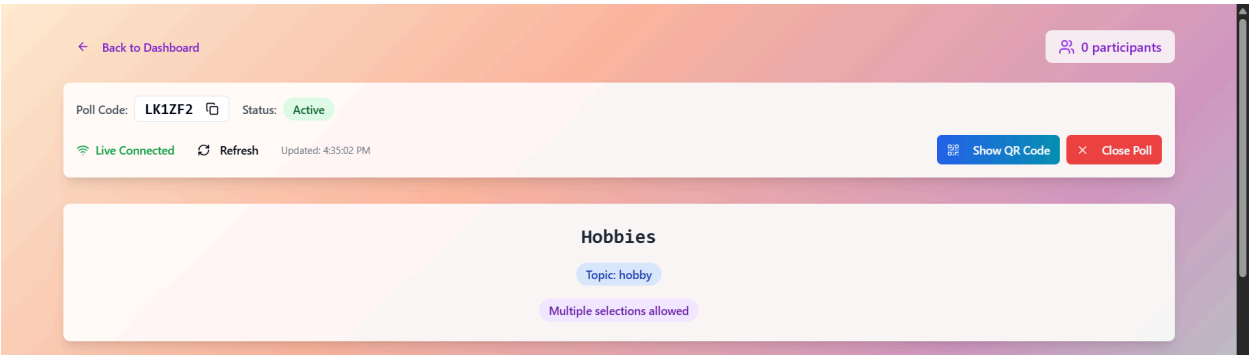


4. Poll Management & Live Control

Dashboard and Live View.

- **Status Toggle:** The user can instantly **Activate** or **Close** a poll, controlling when users are allowed to vote.
- **Relaunch:** The poll can be relaunched, optionally generating a **New Code** and **Resetting Votes** to reuse content for a new class session.
- **Code Sharing:** Displays a unique, short **Poll Code** and a scannable **QR Code** to facilitate quick, easy joining by students.

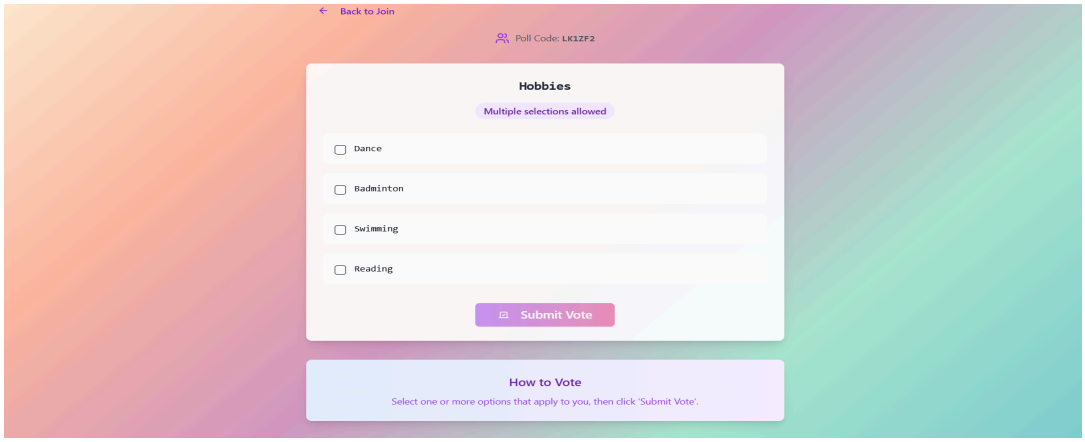




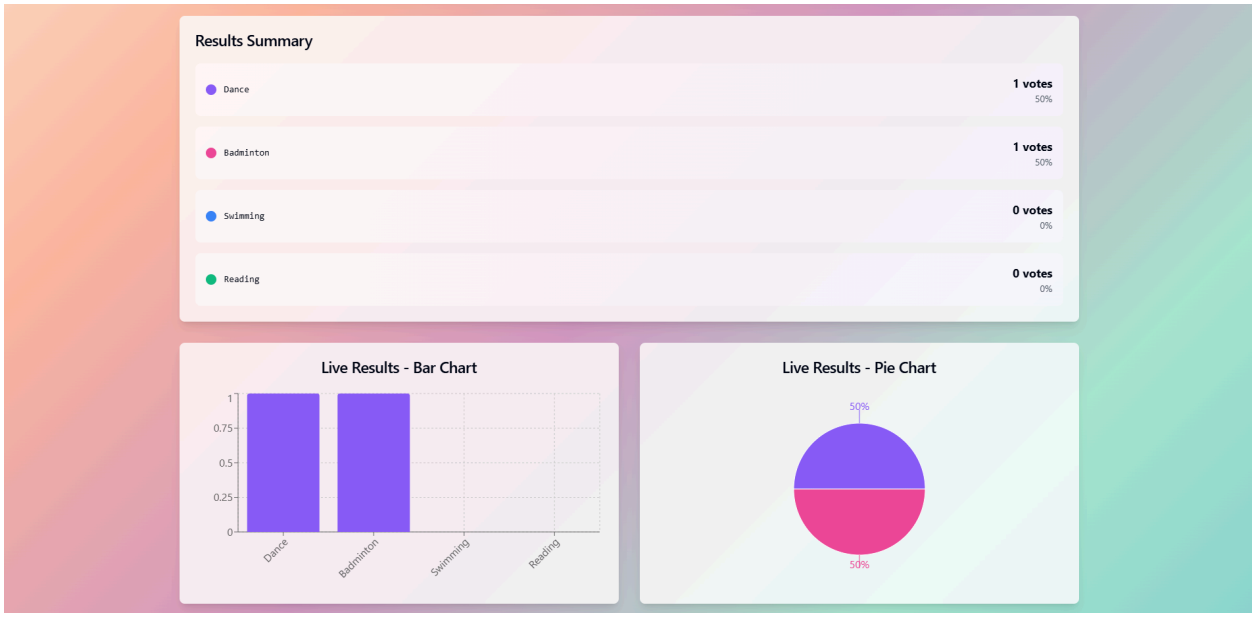
5. Polling Interface

This interface is designed for rapid and secure participation by students.
Join Poll: Students access the poll by entering the unique 6-character **code**

- **Voting Page:** Voting is secured by a client-side fingerprinting mechanism (using **FingerprintJS**) to ensure **one-vote-per-device** integrity.
- **Real-time Status:** Participants are immediately notified



6. Real-time Results and Past Results

The system provides live data visualization for immediate feedback and historical analysis.

- **Live Poll View:** In the Dashboard, can view poll results in **real-time** as votes are submitted, powered by **Socket.IO** integration.
- **Data Visualization:** Results are displayed clearly using **Chart.js** or similar visualization components to show the distribution of votes across options.
- **Past Results View:** Displays historical voting data, allowing the Professor to review the results of previous poll sessions.