AI-Powered Real-Time Presentation System for Seminars

Presented by: Navya Narayani Singh (2201220130078)
Samriddhi Srivastava (2201220130092)

Under the guidance of: Er. Priyanka

Department of Information Technology
Shri Ramswaroop College of Engineering and Management

PROJECT OBJECTIVE

- To develop an AI-driven presentation system that enhances interactivity and engagement in real time.
- To implement NLP-based Q&A handling for dynamic audience interaction.
- To integrate sentiment analysis for tracking audience engagement
- To provide real-time content recommendations and multilingual support.
- To generate analytics and feedback for improving future presentations.

PROJECT OVERVIEW

- The project aims to build an **AI-powered web application** that enhances real-time presentations through interactivity and audience engagement.
- It uses **NLP for Q&A handling** and **sentiment analysis** to assess audience reactions and engagement levels.
- The system provides **real-time feedback, content recommendations, and multilingual support** for dynamic presentations.
- It integrates **Node.js, Express.js, MongoDB, and React.js** to ensure efficient, low-latency, and scalable performance.

PROJECT PROGRESS

- The project is currently in the **development phase**, with the backend setup and initial NLP functionalities completed.
- Around **25% of the overall work** has been accomplished, including core module design and backend integration.
- The **frontend development** using React.js for presenter and audience interfaces is in progress.
- The team is focusing on **optimizing real-time communication** and ensuring **low-latency performance** during live sessions.

PROJECT FRONTEND

- The frontend of the system is developed using **React.js**, ensuring a fast, responsive, and user-friendly interface.
- It consists of **two main dashboards** — one for the **Presenter** and another for the **Audience**, enabling real-time interaction and engagement.
- The **Presenter Dashboard** displays live audience feedback, sentiment analysis results, and AI-generated recommendations during presentations.
- The **Audience Dashboard** allows participants to submit questions, express reactions, and view translated or recommended content in real time.
- The frontend communicates with the backend through **RESTful APIs and WebSocket connections** for seamless real-time updates.

WORK COMPLETED

- Conducted requirement analysis and finalized project architecture and system design.
- Designed core modules — Presenter Support, Audience Interaction, Recommendation Engine, and Analytics.
- Implemented basic NLP model for audience query understanding and speech-to-text conversion prototype.
- Established backend framework using Node.js, Express.js, and MongoDB for real-time communication.

CODE FILES

HOW TO RUN FRONTEND

1. **Install Prerequisites**: Make sure **Node.js** (v16 or above) and **npm** (Node Package Manager) are installed on your system.
   You can verify using: node -v npm -v
2. **Navigate to the Frontend Directory**: Open your terminal or command prompt.
   Move into the frontend project folder:
3. **Install Dependencies**: Install all the required React packages and libraries:
   npm install
4. **Start the Development Server** : Run the application locally.
   The frontend will automatically open in your default browser at:
   http://localhost:3000
5. **Connect to Backend** : Ensure that the **backend server** (Node.js + Express) is running so the frontend can APIs and WebSocket.
   You can update the backend URL in the .env file or in the configuration file if needed

NEXT STEP

- The **frontend and backend integration** will enable real-time interaction between the presenter and audience dashboards.
- The **sentiment analysis model** will be trained and refined using larger datasets to improve accuracy in detecting audience engagement and emotions
- Additional features such as **multilingual translation**, **data visualization dashboards**, and **automated feedback generation** will be implemented.
- Finally, the system will be prepared for **deployment** and **performance evaluation** under real-time presentation scenarios.

THANK YOU