INTERNSHIP PROJECT DOCUMENT

ON

ONLINE POLLING SYSTEM

Submitted by

(Intern)

Alwala Satvika Reddy

Submitted to

Founder - Kanduri Abhinay

CTO - Saniya Begum

INTRODUCTION:

The Online Polling System is a web-based application designed to allow users to create, participate in, and manage polls. It provides a dynamic platform where users can cast their votes, view live results, and delete polls once they are no longer needed. By utilizing modern web technologies like Node.js, Express, and a simple front-end interface with HTML, CSS, and JavaScript, the system ensures smooth interaction and seamless voting. It aims to simplify the process of conducting polls, whether for events, surveys, or decision-making. This project is ideal for students, organizations, and community groups who need a quick and accessible solution for online polling.

SOFTWARE REQUIREMENTS:

Frontend Technologies:

- HTML5 & CSS3: To structure and style the web page, ensuring an appealing and functional user interface.
- JavaScript (ES6+): To implement dynamic features and interactivity, such as voting and live preview.
- React.js: For creating a responsive and interactive user interface that allows real-time updates.

Backend Technologies:

- Node.js: A JavaScript runtime to handle server-side operations.
- Express.js: A lightweight framework for setting up the web server and handling HTTP requests.
- MongoDB: For storing poll data and user responses.
- Mongoose: An ODM (Object Data Modeling) library to interact with MongoDB.

Tools and Platforms:

- VS Code: The primary development environment for coding.
- GitHub: For version control and collaborative development.

PROCEDURE AND METHODS USED:

Step 1: Planning

- Define core functionalities:
    - Ability to create a poll with multiple options.
    - User participation by casting votes.
    - Display live voting results.
    - Option to delete polls after voting.
- Ensure real-time data updates and synchronization.

Step 2: Design

- Create React components:
    - **Poll Creation Component**: For inputting poll questions and options.
    - **Poll Display Component**: For showing available polls with voting options.
    - **Results Component**: For showing voting results after participation.
- Use **styled-components** for styling, ensuring a modern and clean user interface.

Step 3: Development

- **Frontend**:
    - Build the poll creation, display, and voting interfaces with React.js.
    - Implement dynamic updates using React state and hooks.
    - Add responsive CSS for mobile-friendly display.
- **Backend**:
    - Set up a **Node.js** server with **Express.js** for handling poll creation, voting, and deletion requests.
    - Use **MongoDB** and **Mongoose** to store poll data and votes.
    - Implement API routes for creating, fetching, voting, and deleting polls.

Step 4: Testing

- Test the application on multiple browsers (Chrome, Firefox, Edge) to ensure compatibility.
- Verify user interactions like voting and displaying results work correctly.
- Test delete functionality and ensure polls are removed from the database and UI.

Step 5: Deployment

- Share the URL for user feedback and testing.

FUTURE SCOPE:

- Real-time Polling: Allow users to see live updates of votes without refreshing the page.
- Multiple Poll Options: Extend support for more than two options per poll.
- Analytics: Add features to analyze voting patterns and display detailed statistics.

FUTURE ENHANCEMENTS:

1. User Authentication: Implement a login system for users to create and manage their polls.
2. Multi-language Support: Offer the platform in multiple languages to cater to a broader audience.
3. Mobile App: Develop a mobile version of the application for better accessibility on smartphones.

ADVANTAGES:

- User-Friendly Interface: The simple, intuitive UI makes it easy for users to create and vote in polls.
- Real-Time Voting: Participants can instantly see voting results after casting their votes.
- Responsive Design: The application works smoothly across different devices and screen sizes.
- Poll Deletion: Polls can be deleted after completion, keeping the platform clutter-free.

REFERENCES:

- Node.js Documentation: https://nodejs.org/en/docs/
- Express.js Documentation: https://expressjs.com/en/starter/installing.html
- MongoDB: https://www.mongodb.com/docs/