***Assignment:***Real-time Polling Application with Chat Feature

**Objective:** Develop a comprehensive real-time polling application that integrates a voting system and a chat feature. This assignment focuses on combining these two features to create an engaging and interactive application using WebSockets and Node.js.

**Requirements:**

**1. Project Setup:**

- Create a new Node.js project with necessary dependencies (Express and Socket.IO).

- Set up an Express server to serve static files and handle WebSocket connections.

- Integrate Socket.IO to facilitate real-time communication.

**2. Client-side Interface:**

- Design an interactive HTML interface for the polling and chat application.

- Display the poll options, their current vote counts, and the chat interface.

- Implement user interface elements for voting, sending chat messages, and displaying chat history.

**3. Real-time Voting System:**

- Enable users to vote on various topics or options.

- Implement a real-time update mechanism for poll results, notifying all connected users.

- Utilize broadcasting to ensure all users receive updated poll data.

**4. Real-time Chat Feature:**

- Integrate a real-time chat feature alongside the polling interface

- Typing Indicator (Shows a dot animation when a user is typing)

- Allow users to send and receive chat messages in real-time.

- Ensure that chat messages are visible to all connected users.

**5. Data Management:**

- Maintain data structures on the server to store poll options, chat messages, and user information.

- Update poll data and store chat messages as users interact.

**6. Authentication and User Management:**

- Develop a basic user authentication system to uniquely identify users.

- Associate user names with chat messages.

**7. Styling and User Experience:**

- Apply CSS styling to create an attractive and user-friendly interface.

- Differentiate the sections for polling, chat, and voting buttons.

***8. Bonus Challenges (Optional):***

- Allow users to edit or delete their chat messages.

- Implement persistent user profiles and message history.

- Provide options to mute or disable chat notifications.

**Deliverables:**

1. A well-organized Node.js project comprising both server-side and client-side code.

2. A fully functional real-time polling and chat application meeting the outlined requirements.

3. A README file containing clear instructions for setting up and running the application, along with any required dependencies.

4. Documentation or a report explaining your technical implementation, addressing challenges faced, and presenting solutions.

5. Deploy the code and share us the link.

**Evaluation Criteria:**

*Your assignment will be evaluated based on:*

- Functionality: Does the application seamlessly integrate real-time polling and chat features?

- Code Quality: Is the code well-structured, easy to read, and well-documented? Does it follow industry best practices?

- Real-time Communication: Is real-time communication effectively implemented using WebSockets and Socket.IO?

- User Interface: Is the user interface visually appealing, responsive, and intuitive?

- Authentication and Bonus Challenges: Were authentication and any bonus challenges attempted, and how well were they executed?

Submit your assignment within the stipulated timeframe, and be ready to explain your code and showcase functionality during evaluation. Best of luck in creating your comprehensive real-time polling application with chat features!