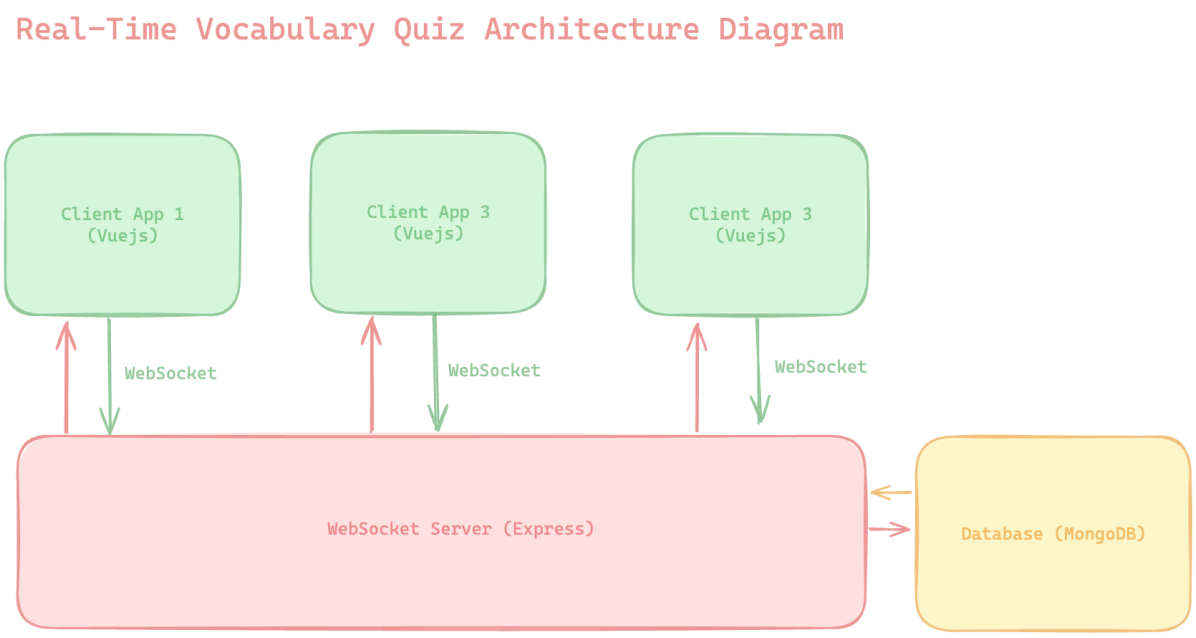
Real-Time Vocabulary Quiz



Component Descriptions

1. **Client Applications:** These are the front-end applications used by the users to interact with the quiz system. This is a web app built using **Vue.js**. It connect to the WebSocket server to receive real-time updates.
2. **WebSocket Server:** Using Node.js and Express to handle real-time communication between the client applications and the application server. It ensures that messages (like quiz questions, answers, and leaderboard updates) are broadcasted to all connected clients instantly. Technologies like Socket.IO or WebSocket API can be used here. This server handles the core logic of the quiz system, including user authentication, quiz management, score calculation, and leaderboard updates. It communicates with the WebSocket server to send real-time updates to clients.
3. **Database:** The database stores all persistent data, including user information, quiz questions, scores, and leaderboard data. A relational database like PostgreSQL is suitable for this purpose due to its robustness and support for complex queries.

Data Flow

1. **User Joins Quiz:**
   - The user enters their username and quiz ID in the client application.
   - The client sends this information to WebSocket server. The server verifies the quiz ID and adds the user to the quiz session.

- The server sends a confirmation back to the client, which updates the UI to show the quiz interface.

2. **Real-Time Score Updates:**
   - As the user answers questions, the client sends the answers to the application server via the WebSocket server.
   - The application server validates the answers, updates the user's score in the database, and sends the updated score back to the client.
   - The WebSocket server broadcasts the updated scores to all connected clients.
3. **Real-Time Leaderboard:**
   - The application server calculates the leaderboard based on the scores stored in the database.
   - The updated leaderboard is sent to the WebSocket server, which broadcasts it to all connected clients.
   - The client applications update their UI to reflect the new leaderboard standings.

Technology Choices

- **Client Applications:** Vue.js, TailwindCSS, Marked for building responsive and interactive user interfaces.
- **WebSocket Server:** Socket.IO or WebSocket API for real-time communication. Node.js with Express for handling server-side logic and communication.
- **Database:** MongoDB for reliable and efficient data storage and retrieval.

These technologies are chosen for their scalability, real-time capabilities, and ease of integration, ensuring a smooth and responsive user experience.