Component Descriptions for Real-Time Quiz Application

# 1. Client (Web/Mobile)

**Description**:  
The Client component represents the user interface of the application. It can be accessed via web browsers or mobile devices. Users interact with this component to participate in quizzes, view leaderboards, and check results.

**Responsibilities**:

* Display quiz questions and options in real-time.
* Send user inputs (e.g., answers, quiz join requests) to the backend.
* Display real-time updates like leaderboard changes.

**Interfaces**:

* Connects to the **WebSocket Server** for real-time updates.
* Interacts with the **API Server** for HTTP-based requests and responses.

# 2. WebSocket Server

**Description**:  
The WebSocket Server manages real-time communication between the Client and the backend. It ensures that quiz updates, such as new questions or leaderboard updates, are delivered instantly.

**Responsibilities**:

* Establish and maintain WebSocket connections with multiple clients.
* Push real-time updates to the clients.
* Manage quiz session states and broadcast relevant updates.

**Interfaces**:

* Receives data from the **Client** (via WebSocket).
* Publishes and subscribes to messages from the **Cache (Redis)** for efficient real-time updates.

# 3. API Server

**Description**:  
The API Server handles all non-real-time interactions between the Client and the system. It manages user authentication, data retrieval, and persistent storage operations.

**Responsibilities**:

* Process HTTP requests from the Client (e.g., user registration, quiz submissions).
* Handle user authentication requests via the **Authentication Service**.
* Manage data storage and retrieval from the **Database**.
* Interact with the **Cache** for temporary data storage.

**Interfaces**:

* Communicates with the **Client** for HTTP requests.
* Interacts with the **Database** for persistent storage.
* Collaborates with the **Authentication Service** for user verification.
* Reads from and writes to the **Cache** for temporary data.

# 4. Database (PostgreSQL/MySQL)

**Description**:  
The Database stores all persistent data for the application, including user profiles, quiz questions, and historical quiz results.

**Responsibilities**:

* Store and manage structured data for quizzes, users, and scores.
* Ensure data integrity and availability.

**Interfaces**:

* Receives queries from the **API Server** for reading and writing data.

# 5. Cache (Redis)

**Description**:  
The Cache component provides a fast, temporary storage solution for frequently accessed data, such as active quiz sessions and leaderboard updates.

**Responsibilities**:

* Temporarily store data for quick access.
* Support the publish/subscribe mechanism for real-time leaderboard updates.

**Interfaces**:

* Interacts with the **WebSocket Server** for real-time data push.
* Communicates with the **API Server** for temporary data storage.

# 6. Authentication Service

**Description**:  
The Authentication Service manages user authentication and authorization. It ensures secure access to the application by validating user credentials.

**Responsibilities**:

* Verify user credentials during login.
* Manage user sessions and tokens.
* Provide authentication status to the **API Server**.

**Interfaces**:

* Communicates with the **API Server** to authenticate users.
* Returns authentication results to the **Client** via the API Server.