Technology Justification

# 1. Frontend: Web/Mobile Client

* **Technologies**: React (Web), React Native (Mobile)

**Justification:**

* React and React Native remain ideal for building dynamic and responsive user interfaces efficiently.

# 2. WebSocket Server

* **Technology**: Node.js with Socket.IO

**Justification:**

* **Node.js** is highly suitable for real-time communication, and **Socket.IO** remains the best fit for WebSocket handling, providing robust real-time event broadcasting.

# 3. API Server

* **Technology**: Node.js with Express.js

**Justification:**

* **Express.js**, as a minimal and flexible Node.js web framework, is well-suited for building RESTful APIs.
* Node.js has non-blocking I/O and an event-driven architecture, which ensures high performance and scalability.
* The JavaScript ecosystem allows seamless integration between frontend and backend with shared libraries and types.

# 4. Authentication Service

* **Technology**: Passport.js with JWT or OAuth2.0

**Justification:**

* **Passport.js** is a versatile authentication middleware for Node.js supporting a wide range of authentication strategies (e.g., JWT, OAuth2.0).
* This ensures secure user authentication and easy integration with third-party login providers like Google or Facebook.

# 5. Cache

* **Technology**: Redis

**Justification:**

* **Redis** is ideal for caching temporary data and supporting pub/sub messaging for real-time leaderboard updates and notifications.

# 6. Database

* **Technology**: PostgreSQL or MySQL (via Sequelize ORM)

**Justification:**

* **Sequelize** is a Node.js ORM that simplifies database interactions and supports both **PostgreSQL** and **MySQL**.
* This allows for structured queries, migration management, and simplified database handling within the Node.js ecosystem.

# 7. Hosting and Deployment

* **Technology**: AWS (Amazon Web Services) or Google Cloud Platform (GCP)

**Justification:**

* Both **AWS** and **GCP** offer scalable and secure hosting. Services like **AWS Elastic Beanstalk** or **Google App Engine** can easily deploy Node.js applications.

# 8. CI/CD Pipeline

* **Technology**: GitHub Actions or Jenkins

**Justification:**

* Automating build, test, and deployment processes using **GitHub Actions** or **Jenkins** ensures faster, more reliable releases.

# 9. Monitoring and Logging

* **Technology**: Prometheus (Monitoring) and ELK Stack (Elasticsearch, Logstash, Kibana)

**Justification:**

* **Prometheus** and **ELK Stack** remain essential for monitoring and logging, ensuring system health and quick issue resolution.

# 10. Testing Framework

* **Technology**: Jest and Supertest

**Justification:**

* **Jest** is a comprehensive JavaScript testing framework for unit and integration tests.
* **Supertest** simplifies HTTP assertions and testing API endpoints.
* These tools ensure high code quality and reliability.