**Business Requirements Document (BRD) (v1.2)**

**1. Introduction**

**1.1 Purpose**

This document outlines the business requirements for an athlete management and event monitoring system. It covers features for managing events, athlete and coach profiles, registrations, and performances.

Note: This project was done individually

**1.2 Scope**

The system includes:

* Event creation and management
* Athlete and coach profile management
* Event registration and performance tracking
* News publication

**1.3 Audience**

* Project stakeholders
* Development team
* Quality assurance team
* System administrators

**2. Functional Requirements**

**2.1 Event Management**

* **Create Event**: Allow users to create events with details such as name, description, photo URL, date, and venue.
* **Update Event**: Enable updates to event details.
* **Get Event Details**: Retrieve detailed information about a specific event.
* **List Upcoming Events**: List all events that are scheduled for the future.
* **List Past Events**: List all events that have already occurred.
* **Add Event Item**: Add items to an event for which athletes can compete.
* **Get Event Registrations**: Retrieve registrations for a specific event.
* **Record Performance**: Record performances of athletes in specific event items.
* **Get Event Results**: Retrieve results for all items in a specific event.

**2.2 User Profile Management**

* **Create Athlete Profile**: Allow users to create athlete profiles with personal details and photo.
* **Create Coach Profile**: Allow users to create coach profiles with personal details, achievements, and photo.
* **Update Athlete Profile**: Enable updates to athlete profiles.
* **Update Coach Profile**: Enable updates to coach profiles.
* **Get Athlete Profile**: Retrieve details of a specific athlete profile.
* **Get Coach Profile**: Retrieve details of a specific coach profile.
* **List All Athletes**: List all registered athletes.
* **List All Coaches**: List all registered coaches.
* **List Coaches Accepting Requests**: List coaches who are open to receiving coaching requests.
* **Toggle Accepting Requests**: Allow coaches to toggle their availability for accepting new requests.

**2.3 Event Registration**

* **Register for Event**: Allow athletes to register for events.
* **Approve Event Registration**: Admin can approve event registrations.
* **Reject Event Registration**: Admin can reject event registrations.
* **Get Athlete Registrations**: Retrieve registrations made by a specific athlete.

**2.4 Coaching Requests**

* **Send Coaching Request**: Allow athletes to send coaching requests to coaches.
* **Respond to Coaching Request**: Coaches can accept or reject coaching requests.
* **Get Coach Athletes**: Retrieve the list of athletes associated with a specific coach.
* **Create Coaching Relationship**: Establish a coaching relationship when a request is accepted.

**2.5 News**

* **Publish News**: Allow the publication of news related to events, athletes, and coaching.
* **View News**: Public users, athletes, coaches, and admins can view news.

**3. Non-Functional Requirements**

**3.1 Performance**

* The system must handle a high number of concurrent users and requests efficiently.

**3.2 Scalability**

* The system should be scalable to accommodate an increasing number of events, users, and data.

**3.3 Security**

* Ensure secure handling of user data and authentication using JWT tokens.
* Implement access control for different user roles (e.g., admin, coach, athlete).

**3.4 Availability**

* The system should have high availability with minimal downtime.

**3.5 Usability**

* The user interface must be intuitive and easy to use for both administrators and regular users.

**4. System Architecture**

**4.1 Microservices**

* **Auth Service**: Manages user authentication and registration.
* **Event Service**: Handles event creation, updates, and performance recording.
* **User Service**: Manages user profiles, coaching requests, and relationships.

**4.2 Communication**

* **Feign Client**: Used for inter-service communication.
* **Spring Cloud Gateway**: Routes and authenticates requests.

**4.3 Data Storage**

* **Database**: Stores user, event, and performance data.
* **AWS S3**: Stores event images and user profile images.

**4.4 Monitoring (Optional)**

* **Prometheus**: Collects metrics from services.
* **Grafana**: Visualizes metrics and creates dashboards.
* **AWS CloudWatch**: Monitors logs and alerts.

**4.5 CI/CD (Optional)**

* **GitHub Actions**: Automates the deployment process to EC2 instances.

**5. User Interface**

**5.1 Frontend**

* **Angular**: Used for building the frontend application.

**6. Conclusion**

This document outlines the core functionalities and requirements for the athlete management and event monitoring system. It serves as a reference for development, testing, and deployment phases to ensure alignment with business objectives.