# Khel-Connect: Requirements Document

## 1. Introduction

### 1.1. Project Purpose

This document outlines the functional and non-functional requirements for the "Khel-Connect" platform, a mobile and web-based application designed to democratize sports talent identification in India. The platform's primary goal is to leverage AI to provide a fair, objective, and scalable method for assessing sporting talent, creating a direct pathway for athletes from any background to be discovered by official scouts and sports authorities.

### 1.2. Scope

The initial scope for the Minimum Viable Product (MVP) of Khel-Connect is focused on a single sport—**cricket**—and a specific action within that sport—**the bowling action**. This focused approach is designed to validate the core technology and user experience, and to provide a strong, demonstrable prototype for the SIH 2025 competition.

## 2. Functional Requirements

### 2.1. Athlete Mobile App (khel-connect-app)

**User Stories:**

- **As an aspiring athlete (Priya)**, I want to create a digital profile so that I can showcase my skills and track my progress.
- **As an athlete**, I want to upload a short video clip of my bowling action so that the AI can analyze it.
- **As an athlete**, I want to receive an objective "Skill Score" for my uploaded video so that I can get unbiased feedback on my performance.
- **As an athlete**, I want to see my scores on a leaderboard so that I can compare my performance with other players.

**Feature-Specific Requirements:**

- **User Registration & Profile Creation:**
  - The app must allow users to register using their name, email, and password.
  - The app must allow users to create a basic profile including their sport (cricket), position (bowler), and location.
- **Video Upload:**
  - The app must provide a simple interface for users to select and upload a video clip.
  - The video format must be compatible with Firebase Storage and the backend processing pipeline.
  - The app must show a loading indicator while the video is being uploaded and

    - processed.
- **AI Analysis & Skill Score Display:**
    - The app must display the "Skill Score" received from the AI model in a clear and prominent manner.
    - The app should provide a basic breakdown of the score, showing key metrics (e.g., arm angle, swing speed, body posture) that were analyzed.
- **Leaderboard:**
    - The app must display a dynamic leaderboard that shows the top-ranked players based on their Skill Scores.
    - The leaderboard should allow filtering by location (e.g., district, state) to highlight regional talent.

### 2.2. Scout Dashboard (khel-connect-scout)

**User Stories:**

- **As a sports authority**, I want to log in to a secure dashboard to view top-ranked talent.
- **As a sports authority**, I want to see a leaderboard of the top players based on their Skill Scores.
- **As a sports authority**, I want to be able to view a player's profile and see their analyzed video clips.
- **As a sports authority**, I want to see the key metrics and analysis generated by the AI for each video.

**Feature-Specific Requirements:**

- **Secure Authentication:**
    - The dashboard must have a secure login system accessible only to pre-approved personnel from sports authorities.
    - Access must be granted on a user-by-user basis, ensuring data security.
- **Data Visualization:**
    - The dashboard must display the player leaderboards in a clean and easy-to-read format.
    - Player profiles should be easily navigable, displaying all uploaded videos and their corresponding Skill Scores.

## 3. Technical Requirements

- **Architecture:**
    - The application must use a serverless, cloud-based architecture.
    - Firebase must be used for authentication, Firestore (database), and Storage (video files).
- **AI/Machine Learning:**
    - The AI model must be implemented within a Firebase Cloud Function to handle video processing.
    - The model must use **MediaPipe** for pose estimation to identify key joint coordinates.

    - The model's custom logic must be written to analyze these coordinates to determine the "Skill Score."
    - The model's output (Skill Score, metrics, etc.) must be written back to the Firestore database.
- **Data Flow:**
    - User uploads video → Video saved to Firebase Storage → A Firebase Cloud Function is triggered → Cloud Function processes the video using MediaPipe/OpenCV and custom logic → Skill Score and metrics are calculated → Results are saved to Firestore → App reads the new score from Firestore and displays it.
- **Scalability:**
    - The architecture must be designed to handle a large number of concurrent video uploads and processing jobs.
- **Performance:**
    - The video analysis and Skill Score generation process should be completed within a reasonable time frame (e.g., under 60 seconds for a 10-second clip).
- **Security:**
    - Data stored in Firestore and Firebase Storage must be secured using Firebase security rules to prevent unauthorized access.
    - The Scout Dashboard must be protected by a secure authentication mechanism.

## 4. Non-Functional Requirements

- **Usability:** The app and dashboard must be intuitive and easy to use for their respective target audiences.
- **Reliability:** The system should be robust and handle common errors (e.g., failed uploads, processing errors) gracefully.
- **Maintainability:** The codebase should be well-commented and structured to allow for easy future development and feature expansion.
- **Accessibility:** The mobile app should be designed with accessibility in mind, ensuring it can be used by a wide range of users.