Individual Contribution Report

SwingAI - AI-Powered Cricket Performance Analysis Platform

Ashvin | 2022101015

Role & Responsibilities

As the **Bowling System & Team Management Lead**, I owned the complete bowling analysis module and the team roster management system. I was responsible for architecting the role-based player display system, ensuring that batsmen, bowlers, and all-rounders were supported with distinct, comprehensive statistics and performance metrics.

Key Contributions

1. Bowling Analysis Module

Component	File	Contribution
Bowling Analysis	`BowlingAnalysis.tsx`	Built the interface for video analysis with bowling-specific metrics.
Video Upload	`BowlingAnalysis.tsx`	Implemented drag-and-drop with file validation and 500MB limits.
Skeleton Overlay	`BowlingAnalysis.tsx`	Created SVG visualizations with bowling action tracking lines.
Classification	`BowlingAnalysis.tsx`	Implemented delivery detection (Yorker, Bouncer, Slower Ball, etc.).

2. Bowling Progress Tracking

- **Statistics Dashboard:** Created speed and accuracy trend charts using Recharts area/line components.
- **Workload Monitoring:** Implemented daily delivery and over tracking with interactive bar charts.
- **Gamification:** Developed 6 bowling-specific achievements including *Speed Demon* and *Maiden Master*.

3. Team Management System

- **Roster Interface:** Built the complete team interface featuring player cards, search, and status filters.
- **Data Architecture:** Extended the `Player` interface to include `bowlingStyle`, `economyRate`, and `wickets`.
- **Role-Based Logic:** Implemented dynamic rendering to switch stat displays based on player roles.

4. Player Squad Development

Player	Role	Key Statistics Added
Jasprit Bumrah	Bowler	95 km/h avg, 88% accuracy, 5.8 economy
Mohammed Siraj	Bowler	90 km/h avg, 85% accuracy, 6.2 economy
Hardik Pandya	All-Rounder	94 km/h bat speed, 88 km/h bowl speed
Ravindra Jadeja	All-Rounder	82 km/h bat speed, 90% bowling accuracy

Technologies Used

- **React 19 & TypeScript:** For type-safe bowling statistics and functional component architecture.
- **Framer Motion:** For card animations and staggered page transitions.
- **Recharts:** For visualizing delivery variety and economy rate trends.
- **MDI Icons:** Integrated `mdiSpeedometer` and `mdiTrophy` for bowling-specific UI.

Problem Solving

Challenge: Role-Based Stat Display
Players required different UI components based on their role (e.g., Bowlers shouldn't see Batting charts). I solved this by extending the `Player` interface with optional properties and using conditional rendering logic:

```
{player.role === 'all-rounder' && (<BattingStats /><BowlingStats />)}
```

Impact & Verification

- **Performance:** Search filtering results delivered in <50ms for the full roster.
- **Quality Control:** Verified with 100% type-safety via `tsc -b` and linting.
- **UX:** Delivered seamless navigation between batting and bowling view modes on the dashboard.