Requirements Stack

ID	Requirement Description	Story Points	Priority	Sprint Number
R1	Implement RESTful FastAPI backend to serve sports data (teams, players, stats, odds) using PostgreSQL and Redis caching.	13	1	1
R2	Integrate live data feeds from APIs for football, basketball, baseball, and UFC. Normalize and store in unified schema.	13	1	1
R3	Build WebSocket interface for real-time game and stat updates to connected clients.	8	1	1
R4	Create front-end interface (React Native or PWA) to display teams, players, and stats.	8	1	1
R5	Implement search and filtering by sport, team, or player across datasets.	3	1	1
R6	Implement user-defined alerts for news mentions and injury updates of subscribed teams.	8	2	2
R7	Implement scheduled background jobs (APScheduler) to poll for new data, injuries, and odds updates.	5	2	2
R8	Integrate betting odds API and display odds for upcoming events with bookmaker info.	5	2	2
R9	Store and display recent injury reports per team/player.	3	2	2
R10	Optimize DB queries and introduce Redis caching for high-traffic endpoints.	3	3	3
R11	Add automated tests (unit, integration) for backend services.	5	3	3
R12	Containerize services with Docker and deploy via Docker Compose.	3	3	3
R13	Add mobile notifications for live game updates and subscribed team news.	5	4	4
R14	Create admin dashboard to manage API keys, data sources, and logs.	5	4	4
R15	Implement data deduplication and validation pipeline for ingest jobs.	5	4	4
R16	Design responsive UI supporting both desktop and mobile clients.	5	4	5
R17	Add authentication (JWT) for user accounts and subscriptions to teams/players.	5	5	5
R18	Build Discord bot/webhook integration for sending alerts and live updates.	3	5	5

Requirements Stack

ID	Requirement Description	Story Points	Priority	Sprint Number
R19	Ensure compliance with betting data policies and user age verification.	2	5	5
R20	Add analytics module tracking user engagement and alert metrics.	3	5	5