

Project Planning Phase

Technology Stack (Architecture & Stack)

Date	25-11-2023
Team Id	Team-592914
Project Name	T20 Totalitarian: Mastering Score Predictions
Maximum Marks	4 Marks

Technical Architecture:

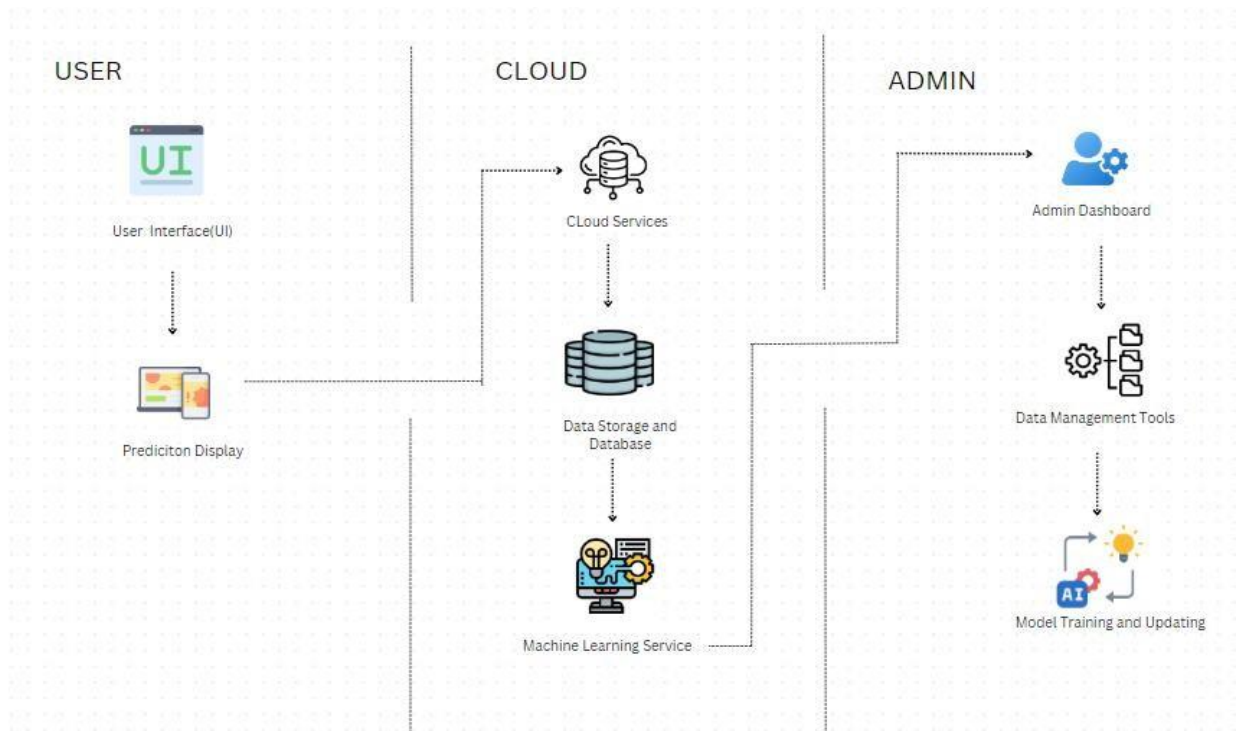


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface (UI)	Users interact through a web UI for predictions, a dedicated mobile app with push notifications, chatbots for natural language interaction, and APIs for programmatic access. Realtime notifications are delivered via email, SMS, or in-app pushes, and voice interfaces may be supported. Social media integration enhances community engagement, providing diverse channels for user interaction and accessibility.	React, Angular, or Vue.js
2.	Authentication and Authorization	Authentication verifies user identity, securing T20 score prediction accounts. Authorization controls access, ensuring users only interact with relevant functionalities and data, enhancing overall security and privacy	OAuth, JWT, or IBM Cloud App ID
3.	Prediction Engine	The Prediction Engine in a T20 score prediction system utilizes machine learning and statistical models to analyse historical data, player performance, and match conditions. It enhances user engagement by providing accurate predictions based on evolving data, offering valuable insights for informed decisionmaking during T20 matches.	Machine learning models (built using scikit-learn, TensorFlow, or PyTorch)

4.	Real-time Data Integration	Real-time data integration in T20 score prediction ensures instant updates on live match scores, player statistics, and events. This enhances prediction accuracy by utilizing the latest information, providing users with a dynamic and engaging experience.	RESTful APIs, WebSocket
5.	Cloud Infrastructure	Cloud infrastructure enhances the T20 score prediction system by providing scalable resources, robust storage solutions, and flexible databases. It ensures optimal performance during peak periods, supports data availability, and offers cost-effective scaling.	Cloud services (AWS, Google Cloud, or Azure)
6.	Database	The database is essential for storing and managing historical match data, user predictions, player statistics, and live scores. It enables real-time updates, personalized user experiences, and accurate predictions by efficiently organizing and retrieving data. The database is integral to the overall functionality and reliability of the T20 score prediction application.	MongoDB, PostgreSQL, or similar databases
7.	Admin Dashboard	The Admin Dashboard in a T20 score prediction system is crucial for user management, content moderation, and realtime monitoring. It enables administrators to oversee predictions, ensure fair play, and swiftly address issues, maintaining the platform's integrity and providing a seamless user experience.	React, Angular, or Vue.js for the admin UI.

8.	Infrastructure	Local deployment aids development and testing, allowing rapid iterations, while cloud deployment provides scalability and global accessibility during live T20 matches. Cloud platforms offer auto-scaling and optimized services, ensuring a resilient and high-performance T20 score prediction system.	Local, Cloud Foundry, Kubernetes, etc.
----	----------------	---	--

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	For a T20 score prediction system, open-source frameworks are used for machine learning tasks. Frontend interfaces are also built. We perform data manipulation, and also create high-performance API. Here we manage asynchronous task processing. The selection of frameworks is based on development preferences.	Scikit-learn, TensorFlow, PyTorch, Flask, Django, React, Angular, or Express.js. NumPy, Pandas, etc.
2.	Security Implementations	In the T20 score prediction system, security measures include user authentication (OAuth, JWT), encrypted communication (HTTPS), data encryption, firewalls, input validation, secure session management, API security, and security headers. Comprehensive monitoring, regular patching, user education, and an incident response plan are implemented for overall security and data integrity.	OAuth, Database encryption tools, HTTP security headers, Network firewalls, cloudbased firewalls, etc.

3.	Scalable Architecture	Scalability in a T20 score prediction system is essential during peak usage like live matches. A 3-tier architecture allows modular scaling with load balancing, while microservices offer granular, independent scaling for efficient resource allocation. The choice depends on factors like system complexity and the need for precise scalability in different functionalities.	Nginx, MySQL, PostgreSQL, Kubernetes, gRPC, etc.
4.	Availability	Ensuring high availability in a T20 score prediction app involves load balancing for even traffic distribution, redundancy and failover for continuous operation, cloud scalability for dynamic resource adjustments, and asynchronous processing for improved responsiveness. A high availability architecture with continuous monitoring further enhances system resilience, ensuring uninterrupted service during live matches and peak user demand.	HAProxy, Clustering solutions (e.g., Pacemaker, Corosync), Google Cloud Autoscaler, distributed databases (e.g., MongoDB, Cassandra), etc.

5.	Performance	Optimizing a T20 score prediction app involves a scalable architecture with load balancing, caching, and CDN usage for efficient handling of requests. Database indexing, asynchronous processing, and response compression enhance performance. Frontend optimization, connection pooling, and monitoring tools contribute to overall system efficiency, ensuring responsiveness during peak usage. Throttling and rate limiting mechanisms control request rates for stability.	Redis, Memcached, NoSQL databases (e.g., MongoDB), Microservices (e.g., Spring Boot, Flask), Cloudflare, etc.
----	-------------	---	---