**Design Sporty Website.**

Design a subscription based sports website which can display scores, game status, history for any game.

Requirements:

1. User login/signup with authentication based on subscription .
2. Retrieving a particular sports data from the past.
3. Retrieving a particular sports data live.

Components:

1. Users
2. Load Balancer
3. Web Servers
4. CDN (Content Delivery Network)
5. Database

Capacity:

1. No.of users: 1 Billion
2. No.of active users/event: 10 million
3. User metadata/user = 1 KB
4. Total user metadata space required per replication = 1TB
5. Sports Metadata (Depends on no. of sports being handled)
   1. Considering we are handling 20 different types of sports across the globe.
   2. Considering each sports type has 10 different events / year. So total events / year= 20\*10 = 200 events/year.
   3. Considering each event has 10 matches. Total matches/year = 2000.
   4. Considering one match details of any sports require 10 MB of metadata storage on average, it will take 10GB/year for storing all the match details.
6. Considering all 10 different sports having a live event at the same time at peak.
   1. Each sport has 10 million active users watching an event in a day. So a total of 100 million users requesting to server in a day at peak if 10 different sports are live at same time.
   2. So the system needs to handle 100 million requests at the same time at peak.

Load Balancing at peak time.

1. To handle 100 million requests, Load balancers should be distributing load using least live network connection (no.of live request a server is handling).
2. Database should be using sharding based on sport type and geolocation.
3. CDN (Content Delivery Network) should be used before querying directly to web servers.

**High Level Block Diagram**

API’s To Be Used:

* API to fetch the user data for login and sign up.
* API to list down the plans available for subscription.
* API to give on going live matches/events based on parameter ‘sport type’ or ‘event\_id.
* API to get details of sport events that occurred in the past based on sport type.
* API to unsubscribe from a plan.

Database Design:

1. Users: *(user\_id, user\_name, password, active\_plan, plan\_expiry)*
2. Plans: *(plan\_id, plan\_name, plan\_cost)*
3. Sports Metadata: *(sport\_id, ongoing\_flag, event\_id, sport\_type, teamA, teamB, sport\_duration, location)*
4. Players: *(player\_id, player\_name, team\_name, sport\_id, age, country)*
5. Teams: *(Team\_id, player\_id)*
6. Scores: *(date, event\_id, player\_id, score, ongoing\_flag)*
7. Live Events DB: *(date, event\_name, event\_id, sport\_id, Score, teamA, teamB).*
8. Past Sports: *(sport\_id, event\_id, wining\_team, date, event\_start\_time, event\_end\_time, teamA, teamB, teamA\_score, teamB\_score).*