**Project Proposal**

Use the “Free NBA API” from “<https://rapidapi.com/theapiguy/api/free-nba>” to build a website that can:

1. Search a player’s stats in a particular season.
2. Players’ career stats
3. Franchise stats like the highest points scored within this franchise
4. Game info like when is the last time the Lakers beats the Thunders by more than 20 points.
5. Player stats correlation(like if this guy scored more than 10 points in the previous game, what is the chance of him scoring 20 points the following game).

The data passed back through the API would be looking like:

{

"data":[

{

"id":237,

"first\_name":"LeBron",

"last\_name":"James",

"position":"F",

"team":{

"id":14,

"abbreviation":"LAL",

"city":"Los Angeles",

"conference":"West",

"division":"Pacific",

"full\_name":"Los Angeles Lakers",

"name":"Lakers"

}

}

...

],

"meta": {

"total\_pages": 50,

"current\_page": 1,

"next\_page": 2,

"per\_page": 25,

"total\_count": 9999

}

}

Which I can then parse and store into self-built data base.

**User flow and app capability:**

The home page should simply show the top 10 scorer for the current season. A nav bar with the categories of rebounds, assists, etc. By clicking the categories with the “limit” and “season” button to see the top players of the selected category of the selected season.

Also there would be a search bar that users can use to search a player’s stats, along with some filtering criteria like the season(s), average or total, etc.

Finally there would be a button to go to the correlation page. In this page, users can do some advance research on the player performance correlation with other data. For example, if a player scored more than 10 points in the previous game, what is the chance of him scoring 20 points the following game.

**Login and security requirement:**

This app would require login to use the advanced “performance correlation” page. The store the user password safely, we would use the course taught bcrypt.

**Database:**

The only thing we are going to store in the user information. So we need only one table to store all the user’s profile info.