

1.Team name:
Sports NOW .

2.Summary:

The concept of our application is to create a web page for users to view their favorite sports and the current stats of the players and teams the user wants to follow. Furthermore, we will have a section for searching the database for specific filters in regards to teams, sports, and athletes. Ultimately, we want to provide the user with a personalized sports webpage where they can see their favorite sports teams and athletes and how they're performing during the season.

3.Description:

The main goal of our project is to create a personalized experience for users interested in sports. The user will be able to customize their experience by providing their favorite sports, teams and players, and stats that they are interested in following. Different datasets and APIs will be the source of our data that we'll provide to the user.

We will also create a feature for the user to be able to search a database containing information about sports, teams, players and statistics. This will enable the user to dive into data deeper if they are interested in something specific and want to find out more about it. The project aims to solve the problem of being shown irrelevant information on sports sites and not being able to access the information that the user needs. Additionally, we will have a section that will compute various stats for your teams/players to see how they're performing on any given week or possibly for the season.

4. Usefulness:

Our project is useful because it provides users with customized sports data that they can see at a glance. It makes it easier for the users to see data relevant to them instead of being distracted by irrelevant information. The user can see sports, teams or players that they are interested in and thus saves time and get better information.

Similar things that already exist are sites such as ESPN.com or nfl.com, but those sites do not deliver the same custom experience that we aim to provide whether it be customizability or combining the exact sports the user is interested in. That's why our site is unique and will be useful for potential users.

5. Realness: For Sports NOW, we will be using multiple APIs to get datasets for each sport. Since each sport has a different API that provides us with the necessary data, we will have to handle each sport differently. This also means that the dashboard, which is a cumulative of the user's choices, will have collections of different data that the user has specified as "favorites". We will get the different APIs from <https://rapidapi.com/category/Sports> which is a collection of multiple APIs across many sports. We aim to use three different APIs for three different sports.

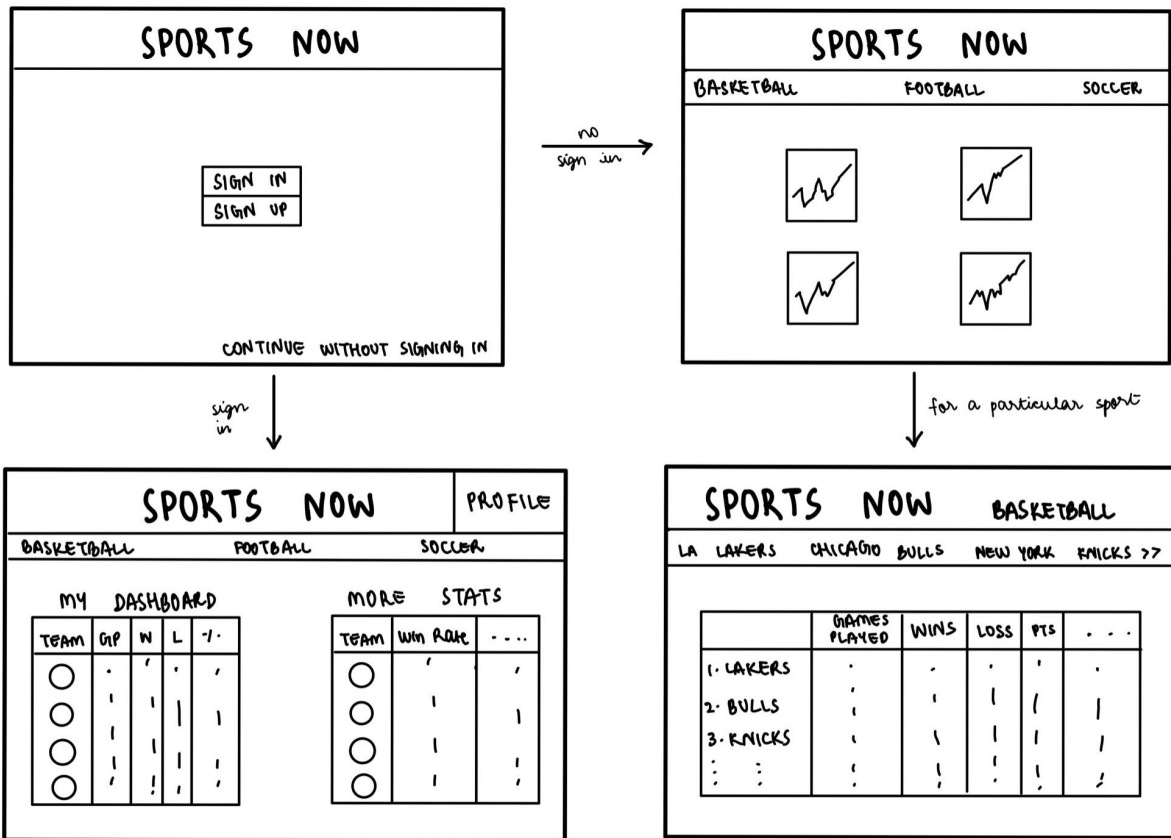
6. Functionality:

The data will be taken from multiple apis listing statistics for players, teams, and sports. For the purposes of this class, we will be focusing on Basketball, Soccer, and Football. We will be using the following data

1. Sport
 - a. Soccer
 - b. Football
 - c. Basketball
2. Teams
 - a. The statistics for a team given a certain sport
 - b. List of players on the team
 - c. Arbitrary info (location, founding date, etc. etc)
 - d. Games (previous,current, future)
3. Players
 - a. Statistics
 - b. Arbitrary info (height, weight, etc.)

The basic functions of our application will include a section to customize your dashboard with certain information that you want to see about your teams, a lookup bar to look for certain stats or players or teams for the sports. Furthermore, we will have sections that calculate stats for your specific teams and players for each sport. Also on the main dashboard will be weekly stats, season stats, and all-time stats for players/teams. Something cool that we could implement would be displaying the betting odds for specific games, this could get complex however and we would most likely have to scrape a betting odds website or use an API.

7.



8. Work Distribution:

Hamza: Back-End and Data Scrapping

Colin: DataBase Design and Back-End

Snigdha: Back-End and API endpoint design

Alex: Front-End