



# SCORESPOT

Created by:  
Lucas Patenaude  
Vishal Vunnam  
Aviral Mittal  
Dip Thapa  
Dominic Jakubauskas

# Project Description

---



Soccer fans want to know the performance of their favorite team as well as the performance of other teams in the league.

Solution? ScoreSpot, a platform dedicated to real-time football match standings with a fan-oriented approach that allows users to favorite their preferred clubs.

Up-to-date information on teams' players, information are included from the world's 6 major leagues!



# Tools Used (Software Development)



Postman : Used for testing API routes before implementing in back-end (5 stars: made API calls more efficient, easy to track. Our application had lots of calls so this was really helpful)



Football-data API: Open API used for populating page with scores, stats, and league information (4 stars: provides necessary information, but paywall decreases information, speed, and call requests.)

SourceTree: Software used to simplify git repository functions (4 stars; used to easily check in and out branches)



Handlebars, NodeJS, PostgreSQL: Software Development tools we used for building the page and connecting back-end and front-end . (4 star, the flexibility of these tools allowed to to really get and modify data from the API; Not without issues)



Docker: Database (3 stars: multiple problems with setting up and running docker image)

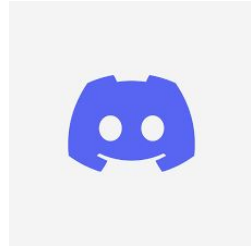


VCS: Github: Version Control used for writing, pushing and sharing code between branches. (5 stars: extremely useful in the process of building and testing our software)

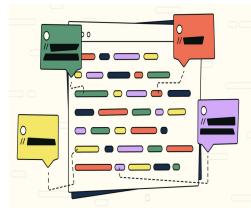
# Tools Used (Communication and Productivity)



Project Task Board for tracking on GitHub: Used to track project tasks and make sure deadline and requirements were met (3 stars: useful at first, but communication and discord often overwrote what was planned)

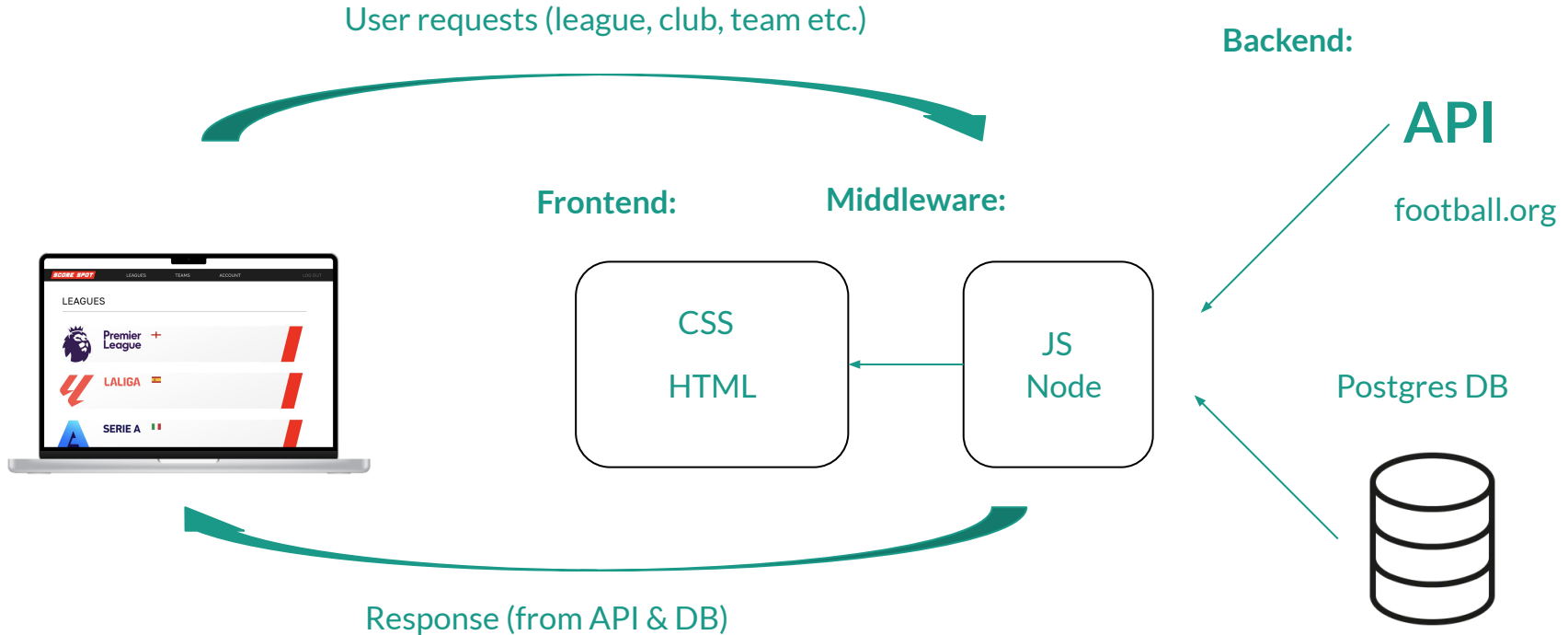


Discord: Used for communication and coordinating meeting times (4 stars: extremely useful for reaching teammates, organized conversations with the use of channels)

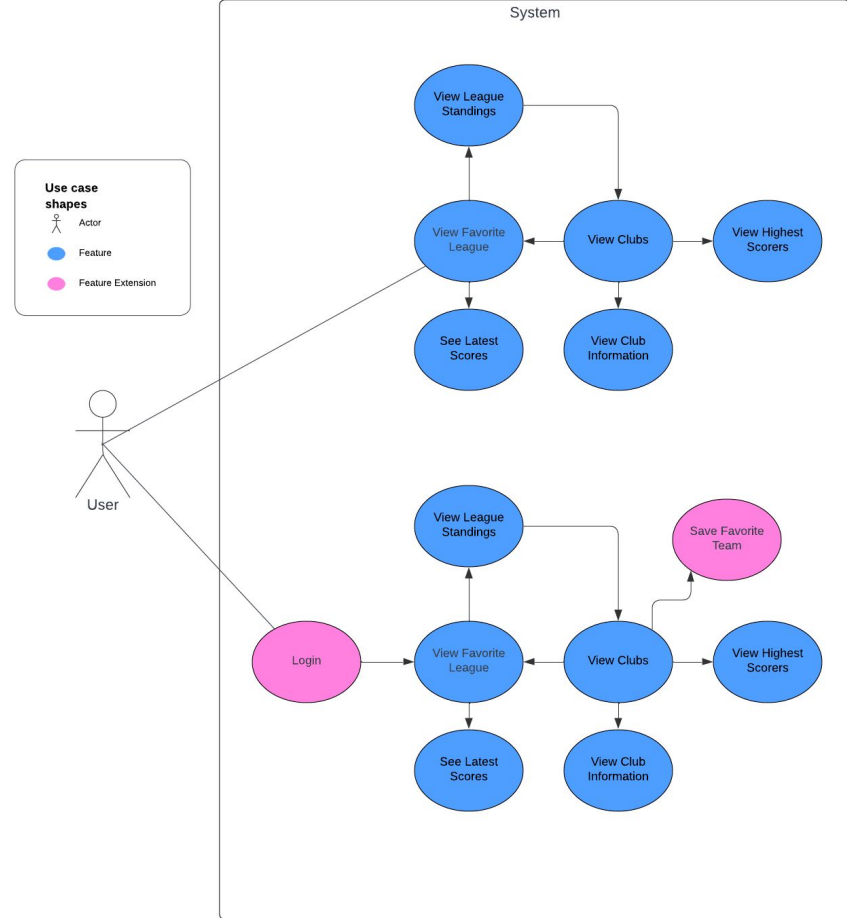


Commenting: Done in order for the team to understand what the code was doing so when making edits or pushing it on to main, everyone understood what the code accomplishes. (5 stars; was extremely helpful and necessary for completion of the project)

# Architecture Diagram



# Use Case Diagram



# Challenges Faced

---

## Coding and Implementation:

- Delayed and limited API requests
  - Inefficient page rendering
  - Frequent API calls leading to un-populated pages
- Connecting and implementing database functions for personal user information
  - Linking the favorite teams table with the user table
  - Fusing back-end and front-end, streamlining user experience



## Team and Communication:

- Proper work distribution
- Meetings and Communication
  - productive meetings
  - punctual and consistent meeting times waning as project progresses

# Future Improvements/Additions

- More information...
  - More leagues, clubs, player information
  - More information on live and current stats
  - Implementing more sports?
    - valid option, or we can stay focused on football, thriving in meaningful specification
- Payment for the API for faster calls, and more reliable page population
- More in-depth relationships between user information and the front-end
- A more user-friendly approach to the design of the page, implementing more modern features
- An ability to keep track of favorite players alongside teams (have an option to track a player, club or both)
- Show historical information for leagues and clubs such as past games, goal scorers, etc
- Mobile app, a quick method to keep up to date with favorite club and current games going on

