

Final Project– Trevan Reese and Jason Stewart

Semester Project Features Revised from HW9

Websites:

[Scoreboard.jasonjstewart.com](https://scoreboard.jasonjstewart.com)

[Games.jasonjstewart.com](https://games.jasonjstewart.com) (Mobile, looks bad on the computer but on phone looks ok)

<https://github.com/trevanreese41/scoreboard>

Our project stems from an idea we shared together last semester and has a scope that is quite large for just one person. We are developing a ReactJS live scoreboard for sports we enjoy and follow throughout the year. Unfortunately, no live sports are played at the moment due to COVID-19, so real-time data will not be observable until play resumes. We retrieve live data from an ESPN endpoint and call the API using React fetch statements.

Below is a thorough list of features that are included in the project:

- ReactJS website displaying daily sports matchups in a “floating grid” format
 - Matchups include the correct team names, team logos, team records, live score, live game status, and/or live game result (FINAL, FINAL/OT, FINAL/3OT, FINAL/10 etc.)
 - Postgame scores will indicate the winner of each game and update team records in each matchup
- Each matchup will display a blue link which indicates the game location (arena, city, state/country)
 - Location will have an onClick event that opens a new tab and queries the arena to return the location on a Google Map
- Each team of the scoreboard is clickable and routes the user to the team’s current schedule, displaying team logos, colors, and records
- Live data is retrieved using React fetch statements and parses JSON from several live ESPN endpoints at a designated interval
 - During live games, scores will update every 30-60 seconds
 - Outside live games, scores will not update/rerender
 - Data is categorized by league, sport, rankings, personal teams
- A navigation bar will allow the user to change between different tabs and view live score information for different leagues and sports
- A tab option will allow for a user to create a custom list of teams they follow and organize the data into a custom live scoreboard across different sports/leagues
 - **add and remove teams from the frontend will be explored but extended out of this project’s scope**
- A tab option will list the current top 25 ranked teams for college sports (football and basketball)

Jason Stewart:

- ✓ Team.js, this includes the teams and other stats about teams being pulled for specific teams from the API
- ✓ Get API to work and populate the scoreboard correctly
- ✓ CSS Grid/Flexbox for Scoreboard and Teams
- ✓ Rankings page (pulling from the API)
 - ✓ React Hooks
 - Refactored to use Hooks and Function rather than components for Scoreboard.js
 - (Still in Development) Mobile Version
 - Still in development but created a version which will look good on a mobile phone which will be the main use of this in our lives
 - (Still in Development) Material UI
 - Implemented Material UI on the mobile version

Trevan Reese:

- ✓ ScoreboardTable.js, this includes the scores, teams, and schedules being pulled from the API
- ✓ Updating the scores, refreshing scores every 10-30 seconds (UpdateScore.js)
 - **this functionality is in place but disabled until live sports resume after COVID-19
- ✓ Create the NavBar and the tabs (NavBar.js and NavBar.css)
- ✓ Link the arenas to the Google Maps API with onClick event (location handling in ScoreboardTable.js)
- ✓ Team names on the scoreboard are clickable and navigate to the team's schedule
 - ✓ Each schedule loads unique schedule, name, and color data from the API (ScheduleTable.js)
- ✓ Transitions throughout the project (fading, growing, etc.) (all .css files)
- After digging deeper into the API, we found that a separate BYU page was unnecessary because the personal scoreboard already houses sports that the user wants to follow, and BYU only exists on two of ESPN's endpoints (basically the BYU page already exists on the Personal Scoreboard Tab (Team.js))

A lot of the development and excitement for the project was impacted by the COVID-19 crisis as sports were canceled, not allowing us to see it in full action. This hindered us in showing off and demoing a website that would show off its intended use. While the website currently shows the games for the postponed or canceled games, we are depending on the ESPN API which if they were to stop promoting data through its endpoint would make our project stop working. So we worked with what we have, hoping to one day be able to use this scoreboard when COVID-19 has passed and our beloved sports start up again. We started off the semester creating the React project and building class based Components for this project. We learned about React Hooks and Functional Components and saw the benefit of changing the already existing code

over to more of this format. This took a lot of time and effort but we were able to see how this was the better and easier way in the long term to build the project, while not all of the project has been refactored the main parts have, giving us better functionality. We did not get to the functionality of being able to add teams or remove them from a teams list yet but that will be coming in the next few months as we continue to work on this project.