

# Fullstack Battleship

Fullstack Development

## Overview

For the final iteration on this project, you will be building a full stack version of Battleship so that users can play with others in the world. For the sake of simplicity and focus on achievable academic goals, this app will NOT dynamically update and users will need to refresh regularly to see if it is their turn. Please keep this in mind as it can make the risk for race conditions quite high. Additionally, I urge you not to worry about things like performance: the userpool and complexity is quite small, so focus on completing the tasks.

For this assignment, you may work with up to two additional students, to make a group of 3. You can also tackle this work by yourself as well.

Also, I recommend you use the code I provide in class as a basis for this project as much of the frustrating setup is handled there. We may not be able to assist you if you follow other patterns or use other technologies.

## Rubric

- Core Functionality - 30%
- Github Repo - 5%
- Working, Deployed App - 20%
- Good Pages and Styling - 5%
- RESTful APIs - 20%
- MongoDB and Mongoose - 10%
- Well Written JavaScript - 5%
- Writeup - 5%
- Bonus Points

## Core Functionality and Pages

This app should have ALL the same functionality of project 2, with changes explained in the following sections. Please do not hesitate to reach out to the staff if there is any confusion OR make sure to leave any comments in the writeup that discuss any decisions you made.

### Login/Register Pages

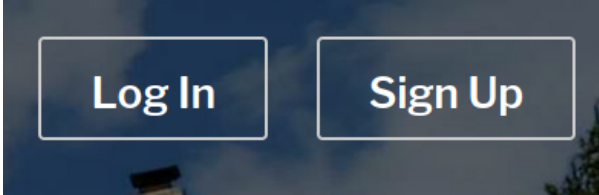
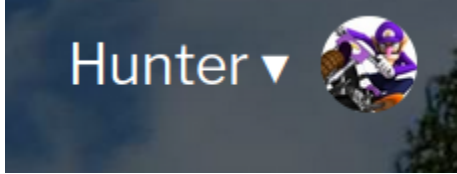
- URL structure - /login and /register

The newest change is that users will now be able to login, create a new account or logout. Being logged in vs logged out can have an impact on the behavior of the page, so I will comment on this for each page. We will be using cookies to track the user behavior, as shown in class. When a user logs in with an existing account or registers with a new account, they should be redirected to the homepage. On the register page, users should be required to validate their password (ie, type it in twice to compare if they are the same.) You can request any user information you wish, but we are only looking for unique usernames and password fields. If a logged in user goes to the register or login page, they should be automatically redirected to the home page.

A logged in user is considered the “active” user.

## Navbar

For the nav bar, you will add a few features. First, if a user is logged out, they should be able to see the login and register buttons. If they are logged in, they should be able to see their username and a sign out button.

	
Logged out	Logged in will show my username and a sign out button (not shown)

Additionally, you will have a button for the new all games page as well as a new button to create a new game. (You may have multiple ways to create a game, but at least have a new game button in the navbar). Clicking the sign out button will clear the user cookies (down in the backend) and redirect them to the homepage.

## New Game and Game Page

URL structure - /game/{game\_id}

When a user clicks on the new game button, your app should create a brand new game with a unique ID. This will create a new entry in the “All Games Page”, and when you create a new game, the user will need to wait for a second player to join (see below). If there is only one user in the game, we would consider the game “Open”; if there are two players in the game, but the game is not over, this is called “Active”; if the game is completed, we call this “Completed”. When a player starts or joins a game, the pieces of their board should be randomized. The logged in player’s board should also be the second game board shown, no matter if they were

the player to create the game or the player to join the game. Otherwise the rules should stay the same and the players should not be able to see where their opponents pieces are.

Once a player has won, display "{Username} Wins" at the top of the page.

There will be no free play mode anymore (you can include it if you wish, but it is not necessary. Any be careful about any unintended interactions.) You can also remove the timer and the reset button.

If a logged out user visits this page, there should be NO interaction (even mouse hover interactions.) Additionally, only show the "hit" spots on both player boards: we don't any cheating!

## All Games Page

URL structure - /games

We need a way for players to manage and see all their games. On the all games page, you will need to show 5 sections pieces of information:

- Open Games: a list of all games that have been started by OTHER players. Include a link to the start page, a button to join (and then redirect) to the game page
- My Open Games: a list of all the games that the logged in player has started but no one else has joined. Make sure to show the start time and a link to the game page.
- My Active Games: a list of all the games that have a second player. Include the name of the opponent, a link
- My Completed Games: include a list of all the games that the current user ONLY has participated in. Include a link to the game page, the name of the opponent, the start/end time and whether the active player has won or not.
- Other Games: include all active games or completed games that the active is NOT part of. Include a link to that game page, the start time and the 2 active users. If the game is completed, show the end time and indicate who won the game.

If a logged in user goes to this page, they should just see all the games in 2 sections:

- Active games: games that are currently ongoing and have 2 players. Please list the players participating in the game and the time/date that the game started.
- Completed games: games that are now over. Please display the participating players, the winner of the game, the start time and the end time.

## Score Page

URL structure - /high-scores

The score page will now work properly. You should list out all the players, the name of wins and the number of losses. The scores should be sorted by number of wins (if wins are equal among

several pages, sort by fewest losses; if that is also tied, sort alphabetically by username). Logged in and logged out users can see this page equally, but if a user is logged in, their username should be bold or in a unique color.

## Landing Page

Url structure - /

This will stay unchanged.

## Working Github and Deployed App Link

For this assignment, I recommend you use Render to host your code, but you are welcome to use any web hosting service you are comfortable with. Please follow the instructions from the lectures or contact any of the teaching staff to get this set up if you need help. Please be sure to add the TA's as collaborators to your Github repos. Remember that it is better to have a partially working game with an active website, rather than a 100% working game that is not deployed.

## Correct Pages and Good Styling

I have listed out all the URLs you will be looking for in the above sections. Additionally, you are required to have a navbar with the specification listed above.

As always, you should have a unique and consistent style across the different pages. There are no specific styling requirements, but make sure that your website looks good on mobile as well as desktop. You are welcome to use any 3rd party styling libraries, such as Tailwind, React Bootstrap, Material UI, etc. Ensure that this is something you would be proud to show an employer.

Finally, if these views are on different pages, consider sensible and good URL design.

## Well Written JavaScript

Now that we're writing logic, you must start considering the quality of the code you're writing. Functions should be simple, easy to read and avoid repetition. Make sure to make helper functions to simplify code in the backend and ensure that your React components are as simple as possible. We are not expecting you to use any "advanced" JavaScript functionality, but you should be writing code that you would be happy to show to a potential employer. You are welcome to use or not use any library of your choice (this means that you are not expected to use Redux, for instance.)

## RESTful APIs

Since this is a full stack app, you are expected to write backend Express APIs using each of the proper RESTful verbs that we learn in this course: POST, PUT, DELETE and GET. It is important that your code respects the promise made with these verbs so that there are no unexpected side effects. You may design as many APIs as you need.

## MongoDB, Mongoose and Security Implementation

You should be correctly connecting to MongoDB and Mongoose with your app. You should have at least 2 collections (i.e., MongoDB tables), but may create as many as you wish.. Additionally, your data should be secure in the ways we show in class. Requests from invalid users should be rejected (for instance, if an API request comes to delete another user's review, that should not succeed.)

## Bonus Points

These tasks are OPTIONAL but will be good experiences if you're interested in exploring further into some harder ideas in web development, but remember that the teaching staff will not help you with this.

### Submit Early - 3pts

Submit this assignment 48 hours before the submission time to receive extra points.

### Password Encryption - 2pts

Ensure that your user passwords are encrypted in the database to add an extra level of security. The walkthrough for this can be found in the slides.

### Click and Drag Setup - 5pts

Just in project 2, the user will be prompted to add their ships manually to the board. Show all the ships to the side and the user will be able to click and drag the ships onto the board to decide their placement. Please be careful about this, as the API design can be quite tricky. Once both players have placed all their pieces, the game can now be played.

### AI Opponents - 5pts

When the active user is in one of their open games (ie, while waiting for the opponent), there should be a button that says "challenge AI". This will create a game that is against the AI, and no other players can join. All the data for this game must be stored in the database and the

logic for the AI behavior (same from project 2) should only be in the backend. Games against the AI should not appear on the high score page.

## Writeup

With your submission, you must include a writeup that touches on the following points. You may discuss any other ideas that you deem salient to this work:

- What were some challenges you faced while making this app?
- Given more time, what additional features, functional or design changes would you make?
- What assumptions did you make while working on this assignment?
- How long did this assignment take to complete?

## Deliverables

All members of a group must submit their projects individually.

Include all the following in your submission on Canvas:

1. A link to your Github repo. If you are working with a partner, you may submit a link to the same repo (for grading purposes, the TA's will likely only look at a single repo so make sure they are identical.) Please note that your Github repo should be named: {firstname}-{lastname}-project3, and if you're working with other people, all names should appear on the repo.
1. A video (uploaded or hosted anywhere you wish) of you and/or your collaborators walking through the project, talking about each page, anything you're particularly proud of, and playing a few rounds of the game. If you are working with a collaborator, please show an example of the game being played online. Make sure to show the following features: logging in, registering, making a game, joining an existing, the high score page, and completing the game. Please also show all the pages when the user is logged out. There is no time limit to this video but should typically be less than 10 minutes. If you prefer not to talk, please add subtitles to describe what you're seeing.
2. A link to your deployed app. As above, if you are working with one or more collaborators, please submit the same link.
3. Your writeup. If you are working with one or more collaborators, you may each write this together or individually, but please indicate this with your submission.
4. The name of your collaborator(s), if any

## Academic Integrity

As always, all assignments are expected to uphold and follow [NEU policies regarding academic integrity](#). Any students that violate these policies will be reported to OSCCR immediately.