Project 3 - Battleship Game Write-up

Timmons Xu and Fan Zhang

1. Challenges Faced

One of the initial challenges we faced was that although the backend move logic was functioning correctly, the frontend did not respond when a move was made. After debugging and testing, we discovered that the frontend was not correctly calling the `/move` API, and the MongoDB database was not updating game state data in real time. Fixing this required correcting the API integration on the frontend to align with the backend route and ensuring that database updates were handled properly after each move.

Another difficulty was ensuring consistent authentication behavior across all pages — especially in handling edge cases where logged-out users attempted to access game detail pages. We had to fine-tune the backend to selectively expose information: hiding sensitive data such as ship positions while still allowing game viewers to see public game progress.

It was also challenging to manage the rendering of the game board in real time — particularly ensuring that the board updated immediately after the opponent made a move. This involved handling turn updates and board synchronization accurately to reflect the current game state for both players.

2. Future Improvements

Given more time, we would like to:

- Allow players to **manually place their ships** before the game starts.
- Build an **AI opponent** so users can also play solo when no one else is available.
- Improve mobile responsiveness and accessibility with better layout and color contrast.

3. Assumptions Made

- We assumed that players would take turns properly and that the backend would enforce turn validation.

- All ships are placed randomly at the start of the game, and this is acceptable instead of manual ship placement.

- We assumed a two-player max model, with no spectator interactions other than read-only views.

4. Time Spent

The assignment took approximately **40–45 hours** to complete, which included:

- ~3h for game design
- ~10 hours for backend setup and user authentication

- ~7 hours for game logic and board management

- ~6 hours for frontend rendering and routing

- ~5 hours for styling and UI enhancements

- ~10 hours for debugging edge cases and polishing the demo video