## What were some challenges you faced while making this app?

One major hurdle was ensuring that both the player’s and the AI’s boards stayed in sync to maintain correct turn order. This required carefully orchestrated event handling and game logic. Another challenge was implementing drag-and-drop placement for ships, especially when dealing with rotation and collision detection—precise calculations and DOM manipulation were necessary to align ships correctly. On top of that, creating the AI’s targeting system demanded a fair balance between randomness and predictable patterns. Lastly, achieving a responsive layout for various screen sizes meant fine-tuning the interface for both mobile and desktop users.

## Given more time, what additional features, functional or design changes would you make?

I would add an onboarding tutorial to guide newcomers through Battleship’s mechanics. Visually, incorporating dynamic animations for ship destruction and smoother transitions would heighten the user experience. Functionally, I would refine the AI to include more advanced targeting (for example, a “search and destroy” mode) and introduce local storage to allow games to be saved and resumed.##What assumptions did you make while working on this assignment?

I assumed that players were already familiar with the basic rules of Battleship, which meant I could focus on core gameplay rather than tutorial creation. The AI’s behavior was deliberately kept simple—random valid attacks were enough for this scope. I opted for a straightforward design instead of elaborate visuals, and I did not integrate local storage by default, prioritizing real-time interaction over persistence.

## How long did this assignment take to complete?

Overall, it took about 30 hours. Most of this time went into building and polishing core mechanics—like ship placement, turn progression, and AI logic—plus implementing drag-and-drop functionality and ensuring responsive grid layouts. Additional time was spent troubleshooting state synchronization problems and handling edge cases to keep gameplay smooth.