Original Prompt given to AI:
I want to create a battleship game. I want the architecture to be HTML, CSS, Javascript, with each of these 3 being kept in separate files. The grid should be 10x10, and there should be 3 ships. Each coordinate in the grid should be a clickable box. When I click the box, it should turn red if a ship is hit, or green if it's a miss. One ship's length is 5, the other is 3, and the last is 2. The ships can be placed horizontally or vertically, and they must be placed randomly.

I gave this prompt because I wanted to get the architecture of the program set up, as well as get the game up and running. It gave me a single 10x10 battleship game that worked. I ended up keeping the set up of the board, but I made major changes to it through later iterations.

2st Prompt given to AI:
I want to make it so that there's two grids. One of the user which is on the left, and one of the ai which is on the right. Each turn alternates between the user and the machine unless one of them gets a hit on a bus, then they can go consecutive turns until they miss. The game ends after someone wins. The user should always start and that should be clearly stated before the game starts.

I asked this prompt because I wanted to create another grid that the machine will use to play against the user. It ended up giving me two 10x10 grids, one for the machine and one for the user. I kept the structure of both birds, but I updated the way the AI played.

3rd Prompt given to AI:
Replace the random computer firing logic with a hunt/target AI:
Hunt mode: fire randomly using a checkerboard pattern.
Target mode: after a hit, prioritize adjacent cells. If a second hit confirms orientation, continue in that direction. Maintain internal AI memory of hits and misses.

I prompted this because while the machine played correctly, it didn't play smart. It would hit the target, but after that it would pick a random place on the grid instead of hitting the rest of the ship. This made the machine much smarter, and I ended up keeping these changes.

4th Prompt given to AI:
Persist the entire game state using localStorage.
Restore the game state automatically on page reload.

I asked this prompt because I wanted to make sure that I can maintain the state of the game across refreshes. This ended up working perfectly, and I ended up keeping it.

5th Prompt given to AI:
Include board state, ships, hits, misses, turn, and game status.
Add a "Clear Saved Game" option.

I asked this prompt because I like statistics, and I wanted to view the accuracy of the user and the machine. I also wanted to make sure that the user can clear the state of the game and refresh whenever they would like. I ended up keeping these changes.