

### AI Prompts:

1. Generate a very simple Battleship web game using (XAMPP) for the server and basic HTML/CSS/JavaScript for the client.
  - a. **WHY:** I asked for this prompt to get the base battleship web game. I made sure not to ask for anything advanced just because we want a base starting game. Also, I made sure to inform the AI that I will be using XAMPP for this.
  - b. **Accepted / Rejected:** The AI generated one HTML file (index.html). This file had the CSS styling inside of it as well. I created a new prompt and informed the AI to split the file into CSS and html for better readability. I then added the files into my htdocs folder inside XAMPP and opened the base battleship game.
2. Modify the Battleship game to move ship placement to PHP using XAMPP, while keeping the frontend and gameplay simple. I also want to make sure the game does not restart when the player refreshes the page. I want the game to only restart when the player clicks the "restart game" button.
  - a. **WHY:** To introduce server-side logic while keeping the gameplay simple. Also, to preserve game state and improve usability.
  - b. **Accepted / Rejected:** I accepted using PHP sessions to store ship positions. I also accepted a program that now has a restart button. This program also does not restart on refresh.
3. Improve the visual design of this Battleship game to look more professional and polished using only HTML and CSS. Keep the layout simple, do not change any PHP logic or game behavior, and focus on spacing, colors, buttons, and overall presentation
  - a. **WHY:** After the game logic was complete, I wanted to improve the overall appearance so the game looked more polished and professional while keeping the functionality unchanged.
  - b. **Accepted / Rejected:** I accepted improvements to layout, colors, spacing, and button styling. I rejected any changes to PHP logic, gameplay behavior, or the addition of advanced visual effects.
4. Extend my Battleship game to include a very simple computer turn. After the player clicks a cell, the computer automatically fires at a random unguessed cell on the player's hidden board. Keep everything in PHP sessions so refresh does not reset. Do not add advanced strategies—random shots only. Provide updated files.
  - a. **WHY:** I wanted to add an opponent so the game includes turn-based gameplay instead of only the player guessing. **This is also a major iteration!** I kept the computer logic simple (random shots), and I required PHP sessions so the game state wouldn't reset on page refresh.
  - b. **Accepted / Rejected:** I accepted adding a basic computer turn, random shot selection from unguessed cells, and storing all game state in PHP sessions with updated project files. I rejected advanced AI strategies, extra features (like multiplayer, databases, or accounts), and any changes that would cause the game to reset on refresh instead of only resetting with the restart button.

5. Add selectable Easy, Medium, and Hard AI difficulty levels to the Battleship game. Easy uses random shots, Medium checks adjacent cells after a hit, and Hard uses a basic strategy to efficiently find ships. Keep the logic simple and session-based, and do not add databases or frameworks.
  - a. **WHY:** I wanted to improve the computer opponent by adding different difficulty levels so the game feels more interactive and shows progression, while still keeping the AI logic simple and easy to understand.
  - b. **Accepted / Rejected:** I accepted three difficulty modes (Easy, Medium, Hard) with gradually smarter shot selection. I rejected advanced AI techniques and complex algorithms.