Battleship Game Architecture Snapshot

Client Responsibilities (HTML / CSS / JavaScript)

- Render player and enemy grids
- Handle ship placement input
- Send player actions to server using `fetch()`
- Display hit/miss results
- Update visual state of the board
- Show game status messages

Server Responsibilities (PHP)

- Store full game state in session
- Randomly place CPU ships
- Validate player shots
- Determine hit or miss
- Generate CPU counter-shots
- Detect game-over condition
- Return results as JSON

Where Game State Lives

- All game state is stored **on the server** using PHP sessions.

- This includes:
  - Player ship locations
  - CPU ship locations
  - Shots taken
  - Hit records
- The client only displays what the server reports.

State Transitions

1. **Page loads**
   - Client displays empty grids
   - Server prepares CPU ships in session
2. **Player places ships**
   - Client sends placement data to server
   - Server stores player ships in session
   - Game becomes ready
3. **Player fires**
   - Client sends chosen cell to server
   - Server checks hit/miss
   - Server generates CPU shot
   - Server returns results
4. **Client updates UI**

   - Marks hit (red) or miss (green)
   - Displays CPU shot result

5. **Game over**
   - Server detects all CPU ships sunk
   - Server sends gameOver = true
   - Client displays win message