Codesandbox template:

Objective: Your task is to design and build a simplified version of the classic game Battleship. This will test your ability to work with data structures, and system design principles in a language-agnostic setting. You are free to choose any programming language you are comfortable with, or even pseudocode, to articulate your solution.

Game Overview:

- **Battleship** is a two-player game, where each player secretly places a number of ships on a grid. Each ship occupies a number of consecutive squares on the grid, arranged either horizontally or vertically. The number of ships and the size of the grid can be adjusted for complexity.
- After the ships have been positioned, players take turns guessing grid coordinates to "fire" at the opponent's ships. The opponent must respond with "hit" or "miss" based on whether a ship occupies that square. If all the squares that a ship occupies are hit, the ship is considered sunk.
- The game ends when all the ships of one player are sunk. The goal of the game is to sink all of the opponent's ships.

Requirements:

1. **Board Initialization:**
    - Define a grid for each player. The standard grid size is 10x10, but your design should allow for easy adjustments.
    - For this task, you can have **predetermined**, **fixed locations** for the ships, and skip over any functionality to manually place or randomly generate ship locations.
2. **Gameplay Mechanics:**
    - Implement a function to "fire" at an opposing player, and detect if the shot was hit or a miss.
    - Implement a turn-based mechanism allowing players to take turns guessing coordinates.
    - Keep track of score and determine when a ship is sunk.
    - Stretch goal: keep track of each player's guesses to prevent repeating the same coordinates.
3. **Winning Condition:**
    - Determine and the winning player once all ships of one player have been sunk.
4. **UI**
    - There should be an accompanying UI to display both board state for players, and buttons to trigger actions.

Please walk us through your design and implementation, highlighting key aspects of your solution, and demonstrate how it works with a few example scenarios.