# GENERAL INSTRUCTIONS

## OBJECTIVE

Create a **Multisession** web game. The game web page should display the current game score and the game board.

The game must support **simultaneous access from multiple browser tabs or devices.** Eachconnected clients should display a **shared and synchronized game board.**

The game features a **3x6 grid** where each cell contains a randomly selected **shape** and **color**.

Players click on a cell to change **both** the shape and color in the cell. The new values must be **valid** - meaning it **cannot match the shape or color of any adjacent cells**.

Each valid click cell increases the score by 1.

The game continues until no valid combination exists for a clicked cell, at which point the **game is over**.

## REQUIREMENTS

1. **Technology Stack:**
   1. Use **TypeScript**
   2. Use **React** for client side
   3. Use **Nodejs** for server side.
2. **Code implementation:**
   1. Ensure the UI updates based on real-time game state.
   2. Use best practices
   3. No test or any other additional development tools and configs are required (such as ESLint, Prettier, advance TsConfig, etc...)
3. You are allowed to use any external resources or tools (e.g., Google, AI assistants), as long as you:
   1. Understand the code
   2. Know exactly what it does
4. **Time and delivery instructions –** 
   1. Submit the code within 5 hours
   2. Submit the code using **mail** - send either **ZIP file of the project code** or **public GitHub repo link.**

# GAME INSTRUCTIONS

## SETUP

1. **Grid Size:** 3 rows × 6 columns.
2. **Shapes** (4 types): Triangle, Square, Diamond, Circle
3. **Colors** (4 types): Red, Green, Blue, Yellow
4. **Initial Board Generation:**
   1. Each cell starts with a random shape and color.
   2. **Initial combinations must be valid** - no adjacent cells can share the same shape or color.
5. **Score**: Initialized to 0

## RULES FOR CHANGING A CELL

1. **On Click:** Both the shape and color of the clicked cell change randomly.
2. **Validation Rules:**
   1. **New shape** must differ from all adjacent shapes.
   2. **New color** must differ from all adjacent colors.
3. **Invalid Moves**: If no valid shape/color combination exists for a clicked cell, the **game is over.**
4. **Scoring:** A valid change increases the player’s **score by 1.**
5. **Cooldown Mechanism:** A clicked cell enters a **3-turn cooldown** and cannot be clicked again until then. Cooldown state should be displayed for each cell.

## GAME END

1. The game ends when the player clicked on a cell that has **no valid shape/color combination.**
2. This results a **Game over**.

## SCORING SYSTEM

1. +1 score for every **valid** shape/color change.
2. Score is tracked throughout the game.

## ONLINE GAMEPLAY (REAL-TIME MULTISESSION)

1. **Multisession Support**
   1. The game must support **simultaneous access from multiple browser tabs or devices.**
   2. All connected clients should display a **shared and synchronized game board.**
   3. There is only **one active game instance** at any given time.
   4. The gameplay experience must be **near-real-time and interactive** across all instances.
2. **Real-Time Updates**
   1. Any **cell interaction, score change**, or **game state update** should be reflected across all connected clients in **near real-time.**
3. **Cross-Device Synchronization**
   1. All game actions must be **synced** across all active sessions and devices to ensure consistent gameplay for all participants.

## CLARIFICATIONS

1. **Diagonal cells are not considered adjacent**
   1. Only cells directly **above, below, to the left, or to the right** of a given cell are considered adjacent.
   2. Diagonal cells **do not** factor into shape or color validation rules.
2. **No user session management is required**
   1. The game does **not** require player authentication, login, or session tracking.
   2. Any user joining the game will see the **shared game board** in its current state.
   3. The game is **not competitive multiplayer** — all users are collaborating in a **single shared game** with a **combined score.**

# BONUS REQUIREMENTS (FOR BONUS POINTS)

## BONUS 1 - LEADERBOARD

1. Add a **leaderboard** that displays high scores from completed games.
2. At the end of a game, prompt the player with an **input field** to enter a **nickname**, which will be saved with their score.
3. The leaderboard **does not need to update in real time** across clients.
4. Add a **dedicated button on the game screen** that, when clicked, will **display the leaderboard.**
5. Optionally, **persist the top 10 scores** across sessions for historical reference.

## BONUS 2 - CANVAS/SVG UI

1. Use **Canvas or SVG** to render shapes and colors.