EAD
Assignment - I

Cricket Scoreboard Web Application Report

Name - M. Srujan Nayak

Roll NO - 1601-23-737-189

Link: https://github.com/srujannayak32/EADASSIGNMENT_189

The Approach Taken to Implement the Scoreboard

The Cricket Scoreboard Web Application was developed using a standard front-end web development stack: HTML for structure, CSS for styling, and JavaScript for all interactive functionality. The core approach was to maintain the game state within JavaScript variables and use Document Object Model (DOM) manipulation to dynamically update the HTML elements displayed on the page.

State Management

The state of the game was managed using a JavaScript object containing key variables:

- teamScore: The total runs for the team.
- wickets: The total number of wickets fallen.
- overs: The number of completed overs.
- balls: The number of balls in the current over.
- striker: An index to determine which player (Rahul or Rohit) is currently at the strike.
- rahulScore: Rahul's individual runs.
- rohitScore: Rohit's individual runs.
- isFreeHit: A boolean flag to handle the free hit logic.

UI and Event Handling

The HTML structure was created with dedicated div elements for each score display (team score, overs, player scores). Buttons for runs, wickets, and other events were given unique IDs or data attributes to be easily selected by JavaScript.

Event listeners were attached to each button. When a button is clicked, the corresponding JavaScript function is executed. This function updates the game state variables, applies the required game logic (e.g., switching the striker on odd runs, incrementing overs after six balls), and then calls a separate function to update the UI.

Key Logic Implementation

- **Scoring:** Functions were created for each scoring event. For a standard run, the function adds the runs to the teamScore and the current striker's score. For extras like Wide, Bye, and Leg Bye, only the teamScore is incremented.
- **Striker Switching:** After each ball, a check is performed to see if the runs scored were odd. If so, the striker variable is toggled between 0 and 1. The "Switch Striker" button also manually toggles this variable.
- **Overs:** The balls count is incremented for every legal delivery. When balls reaches 6, it is reset to 0, and the overs count is incremented.
- **Wickets:** The "Wicket" and "LBW" buttons trigger a function that increments the wickets count. It also updates the current striker's status to "out" and sets the other player as the new striker. A check is implemented to ensure the total wickets do not exceed 10.
- **Free Hit:** When a "Free Hit" button is clicked, the isFreeHit flag is set to true. In the next ball's scoring function, this flag is checked. If it's true, the wicket logic is bypassed, and the flag is reset to false.

Challenges Faced and Resolutions

Challenge: Managing the Striker

The primary challenge was correctly identifying and switching the striker, especially after odd-numbered runs and wickets.

- **Resolution:** A simple integer striker (0 for Rahul, 1 for Rohit) was used to index an array of player objects. This made it easy to access and update the current striker's data. A conditional statement after each run button click checked if the run value was odd (run % 2 !== 0).

Challenge: Handling Free Hit Logic

Implementing the free hit rule where a wicket cannot be taken on the next ball was tricky. The logic needed to be conditional and one-time use.

- **Resolution:** A boolean variable, isFreeHit, was introduced. It's set to true when the "Free Hit" button is clicked. The wicket function includes an if (!isFreeHit) condition at the beginning. If isFreeHit is true, the function simply returns without processing the wicket, and the flag is reset to false. This ensures the rule is applied only to the immediate next ball.

Challenge: Edge Case Handling

Ensuring the application correctly handled edge cases like a maximum of 10 wickets was important for robustness.

- **Resolution:** An if (wickets < 10) check was added before incrementing the wickets variable. If the team has already lost 10 wickets, the wicket button becomes non-functional, or an alert message is displayed to the user.