Howzatt! Cricket Scorekeeper

A Web-Based Live Scoring Application

Project Report

April 2025

Overview

Howzatt! Cricket Scorekeeper is a comprehensive, web-based cricket scoring application developed with HTML, CSS, and JavaScript. It offers real-time ball-by-ball scoring, maintains detailed player statistics, provides a live commentary feed, and generates match summaries with results. All application logic is centralized in `score.js` and match data is persisted via `localStorage`.

1. Setup Page

Files Used: `setup.html`, `setup.css`, `score.js`

Purpose: Initialize match parameters and prepare the scoring session.

Features:

- Text inputs for Team 1 and Team 2 names.
- Dropdown menu to select the toss-winning team.
- Dropdown menu to choose toss decision (Bat or Bowl).
- **Start Match** button: validates inputs, stores match details in `localStorage`, and navigates to `live.html`.
- **Reset All Player Statistics** button: clears any existing player stats from previous sessions.

UI Design: A centered, minimalist form with clear headings and buttons, styled in `setup.css`.

2. Live Match Scoring Interface

Files Used: `live.html`, `live.css`, `score.js`

Purpose: Enable dynamic entry of each ball's outcome and update match status instantly.

Displayed Information:

- Current innings indicator (e.g., ”First Innings” or ”Second Innings”).
- Overall scoreline: runs/wickets and overs bowled.
- Striker and non-striker tables: Runs, Balls Faced, 4s, 6s, Strike Rate.
- Current bowler's table: Overs, Maidens, Runs Conceded, Wickets, Economy Rate.

Interactive Controls:

- Run buttons: Dot, 1, 2, 3, 4, 6.
- Extras: Wide, No Ball, Byes (handled per cricket rules).
- **Wicket** button: prompts for next batter's name and logs wicket event.
- **Run Out** button: prompts for number of runs completed before dismissal.
- Automatic strike rotation on odd runs.
- End-of-over prompt to enter new bowler's name.
- Navigation to **Live Commentary** (`livecom.html`) and **Full Scorecard** (`scorecard.html`).

Special Rules Implemented:

- Extras scoring: wides and no-balls add runs without counting balls; byes increment extras and balls.
- Current Run Rate (CRR) and Required Run Rate (RRR) are calculated during the chase.
- Each ball's event is recorded for commentary feed.

3. Scorecard Page

Files Used: `scorecard.html`, `scorecard.css`, `score.js`

Purpose: Display a full summary of batting and bowling performances for both innings.

Features:

- Batting scorecards for First and Second Innings: lists all batters (out or not out) with their Runs, Balls, 4s, 6s, and Strike Rate.
- Bowling scorecards for both innings: lists all bowlers with Overs, Maidens, Runs Conceded, Wickets, and Economy Rate.

- **Return to Live Score** button to resume scoring.

Data Binding: Reads stored JSON objects from `localStorage` and dynamically populates table rows.

4. Live Commentary Page

Files Used: `livecom.html`, `livecom.css`, `score.js`

Purpose: Provide a ball-by-ball textual log of the match.

Features:

- Header indicating current innings and over.
- Paragraph element that appends each ball's commentary string (e.g., "2.3 Bumrah to Kohli, 4 runs").
- Filter functionality: clicking a player's name in tables filters commentary by that player (optional enhancement).
- **Return to Live Score** button.

Storage: Commentary entries are stored in an array in `localStorage` and rendered on page load.

5. Match Summary Page

Files Used: `summary.html`, `summary.css`, `score.js`

Purpose: Calculate and display the final outcome once the match concludes.

Outcome Logic:

- If Team batting first has more runs than second innings total: display "*Team A wins by X runs!*"
- If Team chasing reaches target: display "*Team B wins by Y wickets (Z balls remaining)!*"

Additional Controls:

- Displays the final aggregate scoreline (e.g., "61/3 (2.0 ov) vs. 60/5 (2.0 ov)").
- **Start New Match** button clears `localStorage` and redirects to `setup.html`.

Storage Mechanism

All match datateam names, player stats, ball-by-ball logs, and summaryis stored as JSON in `localStorage`. This ensures data persistence across page navigations and browser refreshes.

Customizations

Optional enhancements implemented or possible:

- Support for wides, no-balls, byes, and leg-byes, each following official scoring rules. Also customized to handle the cases where some extra runs can be scored on these events.
- Run Out handling with intermediate run count. Also manages strike handling depending upon which end the runout was made and whether the batters crossed during the runout.
- Live commentary with filtering by batter or bowler. Live commentary also handles over changes, innings changes, wickets, new batters and all.
- Multi-match statistics archive stored in `localStorage`. The career stats (either batting or bowling) of any player can be accessed by clicking on the player's name on the scorecard page. Also there is a button to clear all player statistics from the local storage on the setup page.
- The scorecard page shows the batting and bowling scorecards for both innings. The scorecards are displayed in a tabular format with all the relevant statistics for each player.
- Extendable to longer formats by adjusting overs and innings logic.
- The live page shows the current run rate and required run rate for the chasing team along with the number of runs required and the number of balls left.
- Implemented the undo button logic which is very important functionality in case of such apps. There is an undo button on the live.html which undos your action in case any wrong button is pressed.
- The design is based on a single theme and includes background images to make it look similar to apps such as cricbuzz.
- Very minute things are handled in this code. For example, a bowler cannot bowl two consecutive overs. Also, there exists the logic to accept the extra runs input on no-ball, wide and byes. Also, in the case of runout, strike management is done depending on whether the batters crossed before runout and which end was the runout done.

References

- FreeCodeCamp: LocalStorage vs SessionStorage
- Cricbuzz and ESPN Cricinfo for design inspiration.

Conclusion

Howzatt! Cricket Scorekeeper delivers a user-friendly live scoring solution, capturing every aspect of even a multi-over match. Its modular design, centralized logic in `score.js`, and use of `localStorage` make it robust, maintainable, and easy to extend for future formats and features.