Voice-Controlled Scorekeeping

A voice-powered front-end interface for traditional baseball scorekeeping systems

*By **Vanessa Ching** and **Siva Indukuri***

Mentored by Dr. Anton Dahbura

Sports Analytics Research Group (https://sports-analytics.cs.jhu)

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Introduction

Scoring a baseball game requires tracking many events in real time, including every pitch, hit, out, and run.

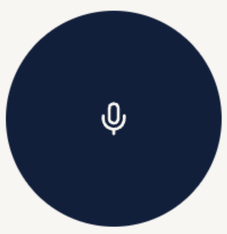
Traditional scorekeeping relies on manual data entry, forcing scorekeepers to split their attention between watching the game and recording plays. This often leads to mistakes and delays.

To address this, this project introduces a voice-controlled system where users simply describe plays out loud. The system converts speech into structured data, making scorekeeping faster, more accurate, and easier to manage in real time.

The system is designed to integrate with traditional scorekeeping platforms, offering voice as an alternative to point-and-click data entry.

Objectives

- Develop a voice-controlled front-end to capture game events in real time, designed to interface with existing scorekeeping systems
- Convert spoken input into accurate, structured data as a voice-driven alternative to point-and-click entry
- Reduce manual effort and errors while keeping focus on the game



Tech Stack

Frontend

Backend

AI / Speech

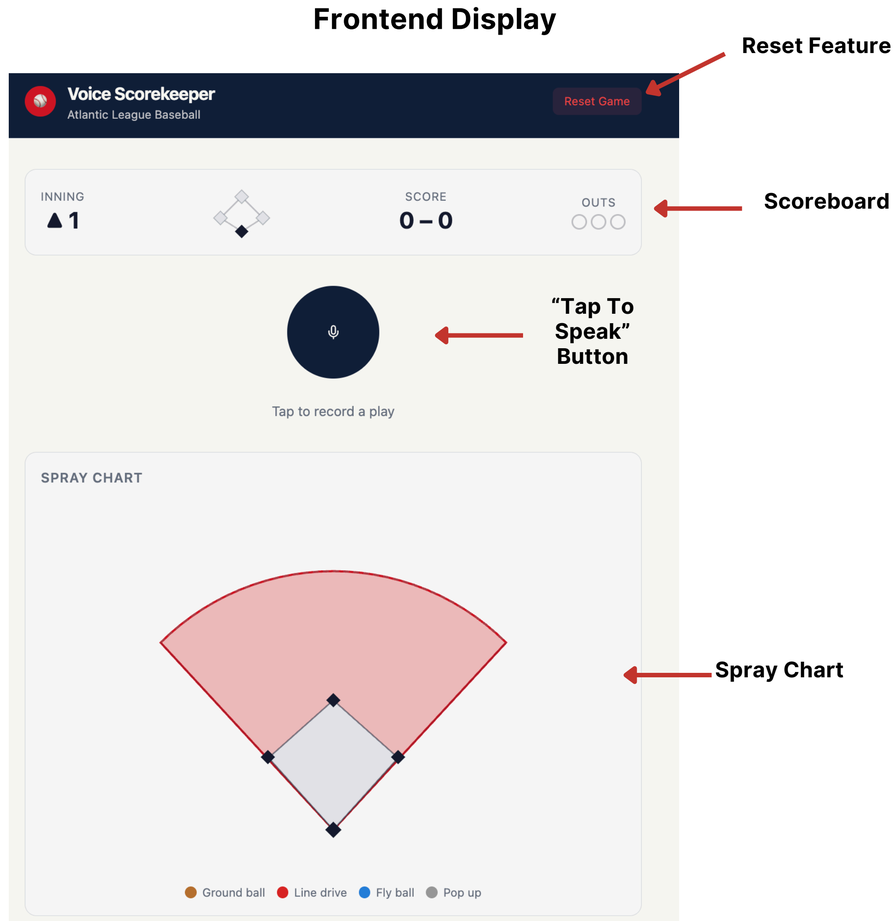
Features

1

Our scorekeeping tool runs entirely in the browser, no installation needed, just a microphone, making it easy to drop into any existing workflow without replacing the systems already in place.

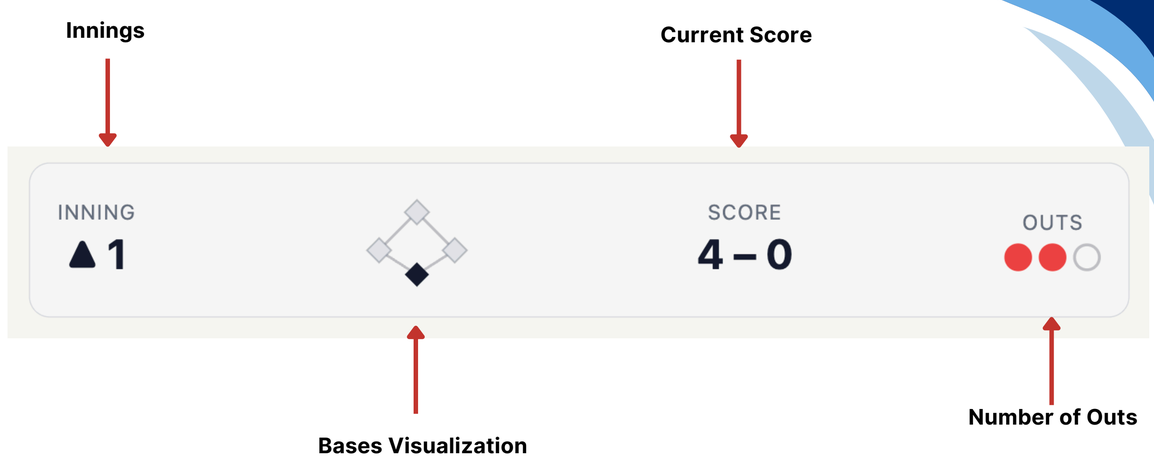
- **Voice-to-play transcription**
  - Users speak plays out loud and the browser microphone captures audio in real time, eliminating the need to type during live gameplay.
- **AI-powered play parsing**
  - The system extracts structured data (batter, action, runners, RBIs) from natural speech using an LLM backend.
- **Live scoreboard with inning tracking**
  - Scores update automatically after each confirmed play, keeping both teams' totals current throughout the game.
- **Spray chart visualization**
  - Batted ball locations are plotted on a field diagram, giving a quick visual summary of hitting patterns.
- **Play-by-play log with confirmation flow**
  - Every parsed play is shown to the scorekeeper for review before being committed, catching any AI misinterpretations.
- **Works with local models via Ollama**
  - The AI backend is swappable, so the app runs entirely offline and for free without any API key.

Frontend Display



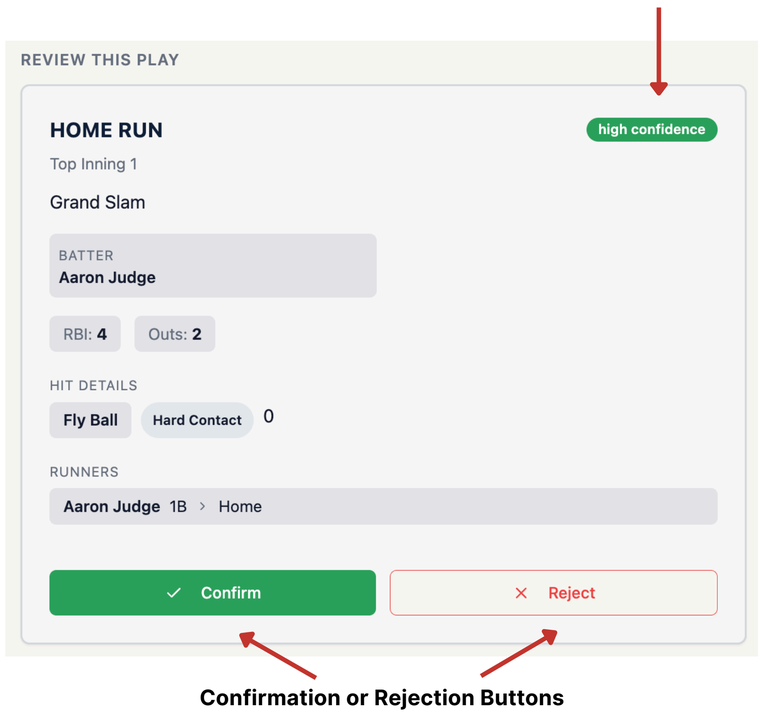
3 Scoreboard

The live scoreboard updates automatically after each confirmed play. It tracks runs per inning for both teams, displays the current inning, and maintains a running total, providing scorekeepers with a full, at-a-glance view of the game without requiring manual entry.



4 Verification System

Before any play is logged, the system displays a structured confirmation card with the parsed details — batter name, play type, runners affected, and RBIs. The scorekeeper reviews and either confirms or rejects the entry, ensuring each play is complete and unambiguous before it reaches the official record. This human-in-the-loop step prevents AI misinterpretations from corrupting the scoresheet, which is especially critical during fast or noisy sequences.



Conclusion

Voice-Controlled Scorekeeping demonstrates that AI-assisted tooling can meaningfully reduce the cognitive load of real-time sports data entry. By combining browser-native speech recognition with a flexible LLM backend, the system produces accurate, structured play logs with minimal user effort.

Try it out! Scan here!