



Highland Rim Basketball Stats Application

Developer Documentation

Spring 2025 Software Engineering Project



Table of Contents

1. Project Overview
2. Key Features
3. Team Roles
4. Tech Stack
5. Getting Started
6. Application Pages
 - Home Page
 - Game Page
 - Player Stats Page
 - Team Stats Page
 - Settings Page
7. Project Structure
8. Scripts
9. Deployment Instructions
10. Github Usage
11. Future Improvements

1. Project Overview

This project was developed to assist the Highland Rim High School Boys' and Girls' Basketball teams in tracking in-game and practice statistics. The goal of this app is to streamline stat recording, reduce errors, and provide valuable analytics. The final product is a web-based application deployed on a tablet for courtside use.

2. Key features

- Track the shot location on the court
- Monitor which third of the shot clock the shot was taken
- Record tempo and offensive events
- Separate views for games vs. practices
- Modular pages (Game Stats, Team Stats, Player Stats, Settings, Home)
- Optimized for tablet touch interface

3. Team Roles

Each team member was responsible for a specific page/section:

- Game Stats: Records real-time game actions
- Team Stats: Displays aggregated team performance
- Player Stats: Shows individual player data
- Home Page: Navigation and overview
- Settings: General app configuration
- Backend & DevOps: Handles data flow, APIs, and deployment
- Database: Manages storage using MongoDB

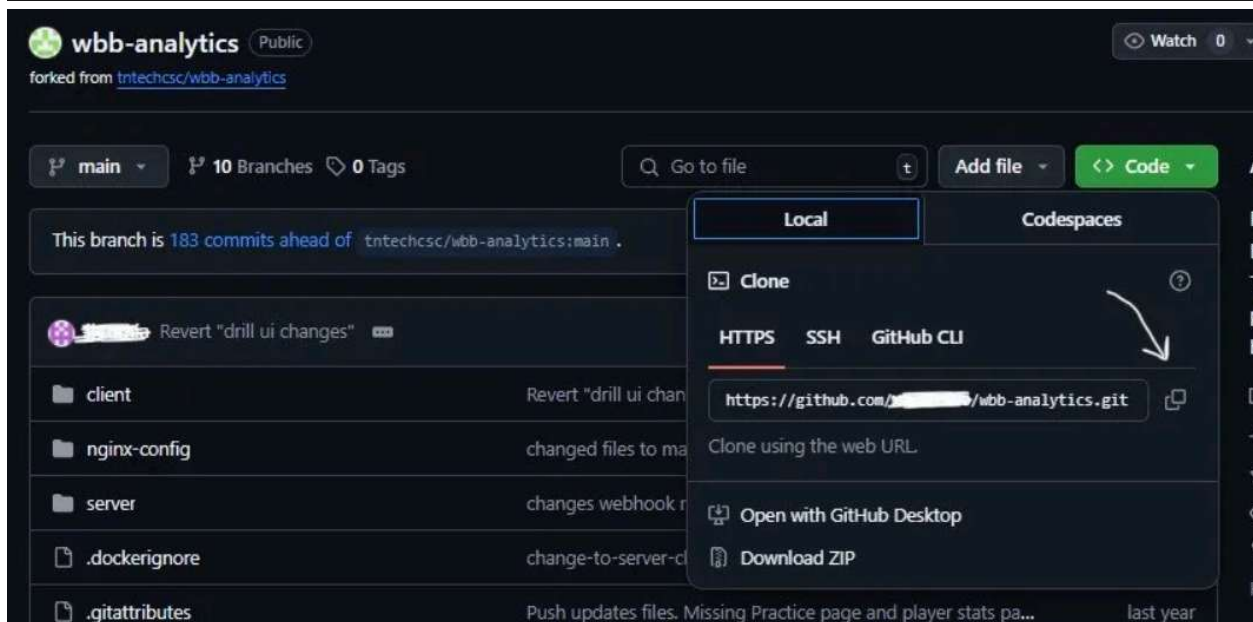
4. Tech Stack

Area	Tech
Frontend	React, CSS
Backend	Node.js, Express
Database	MongoDB
Deployment	Docker, NGINX

5. Getting Started

- Cloning the repository:

```
git clone https://github.com/your-org/WBB-Analytics.git
cd WBB-Analytics
```



- Install dependencies:

```
npm install
```

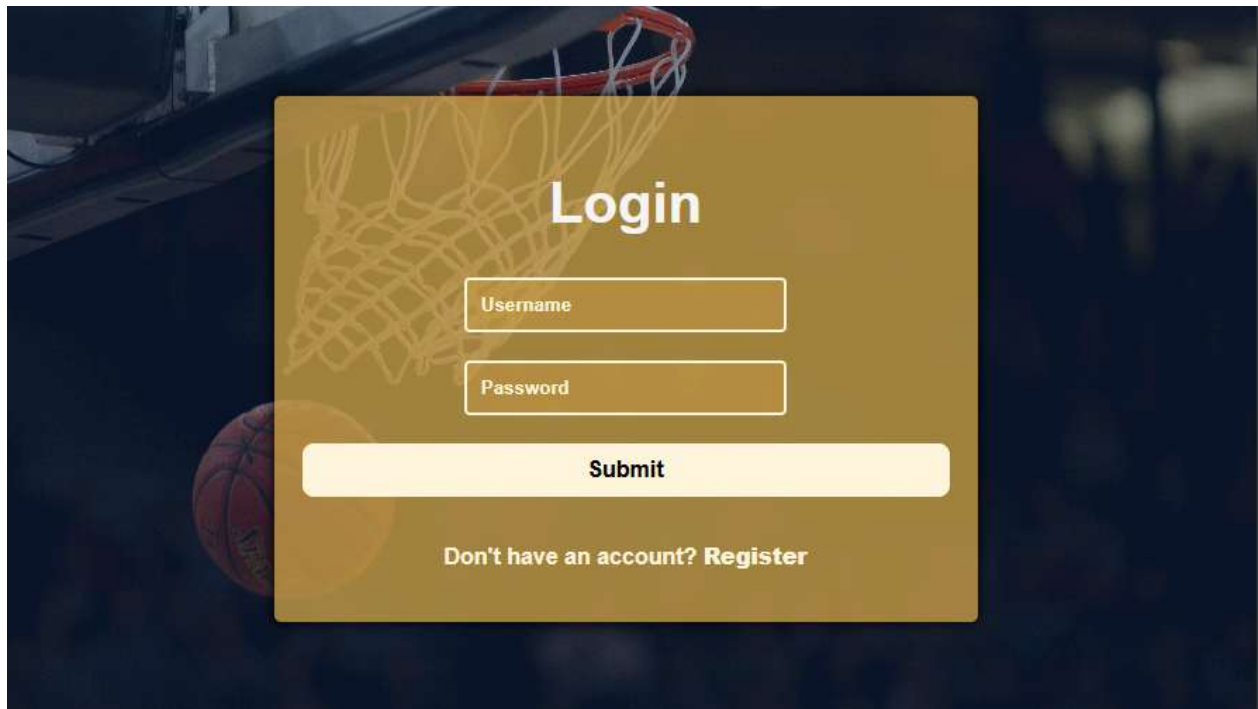
- Starting the development server:

```
npm run dev
# or
npm run dev_offcampus
```

```
C:\Users\Senior Capstone\wbb-analytics>npm run dev_offcampus
> wbb-analytics@0.1.0 dev_offcampus
> concurrently "npm run start-client" "cross-env NODE_ENV=DEVELOPMENT_OFFCAMPUS npm run start-server"

(node:33140) [DEP0060] DeprecationWarning: The 'util._extend' API is deprecated. Please use Object.assign() instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
[0]
[0] > wbb-analytics@0.1.0 start-client
[0] > cd client && cross-env REACT_APP_SERVER_URL=http://localhost:3001 npm start
[0]
[1]
[1] > wbb-analytics@0.1.0 start-server
[1] > cd server && node src/server
[1]
[0]
[0] > wbb-analytics@0.1.0 start
[0] > react-scripts start
[0]
[1] we are off campus
[1] mongodb+srv://admin1234:Admin!1234@cluster0.agkeq.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
[1] NODE_ENV: DEVELOPMENT_OFFCAMPUS
[1] mongoURI: mongodb+srv://admin1234:Admin!1234@cluster0.agkeq.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
```

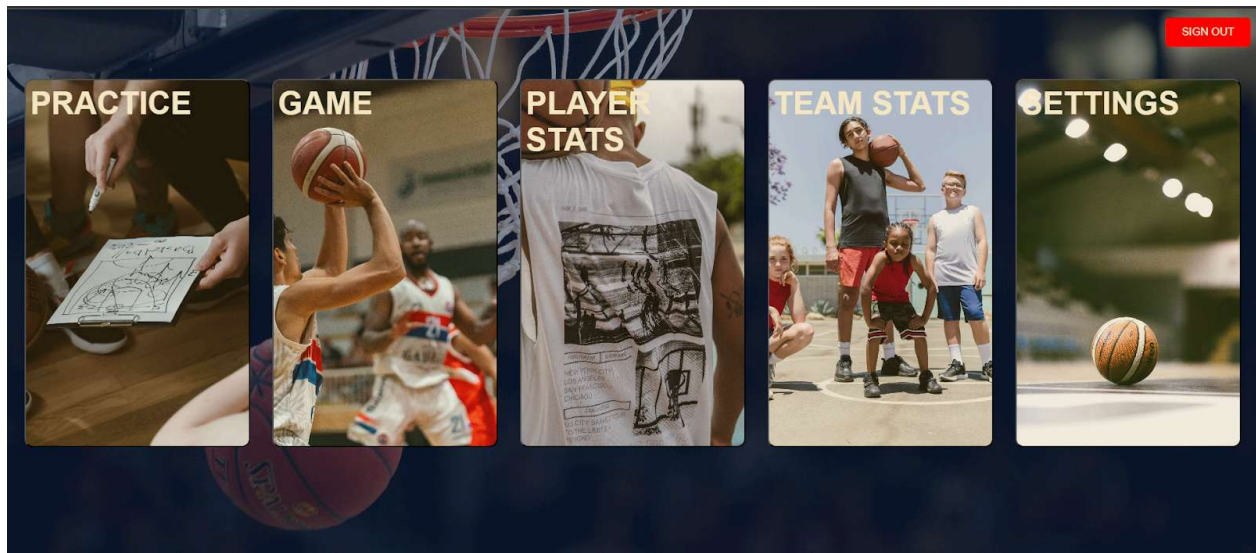
This runs both the client and server on <http://localhost:3000>. After launching, you should see the Login Page.



6. Application Pages

- **Home Page**

The Home Page acts as the main dashboard of the app. It provides a simple, visual navigation experience for coaches and staff, allowing them to quickly jump into the desired section based on their current needs, whether that's recording a practice, loading a game, or checking player and team stats.

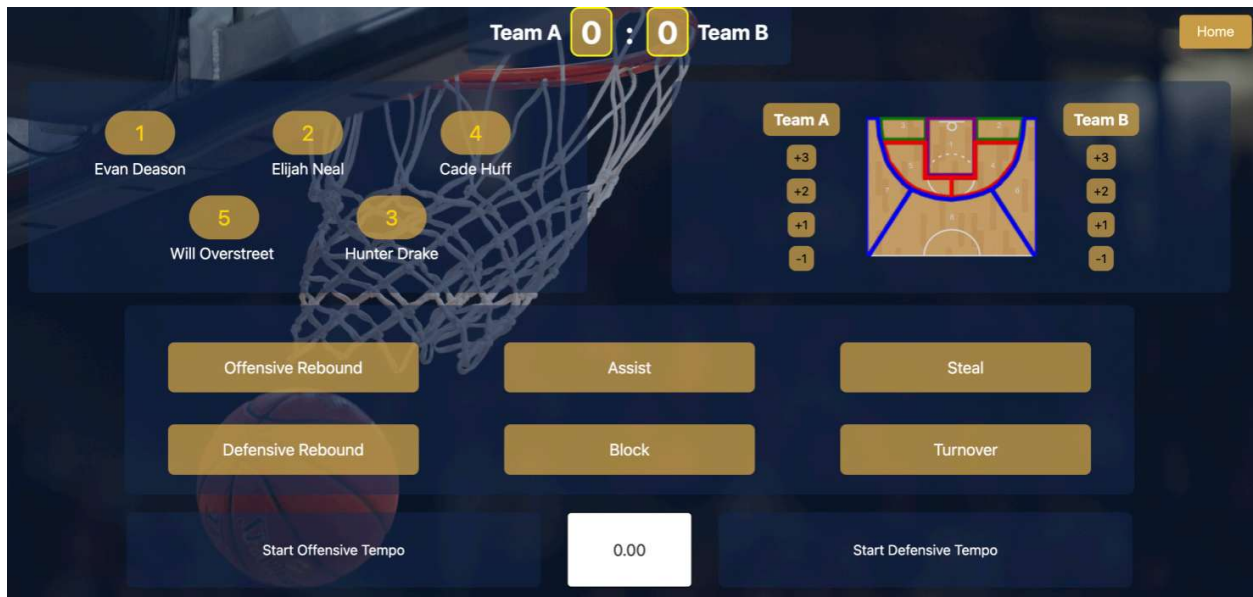
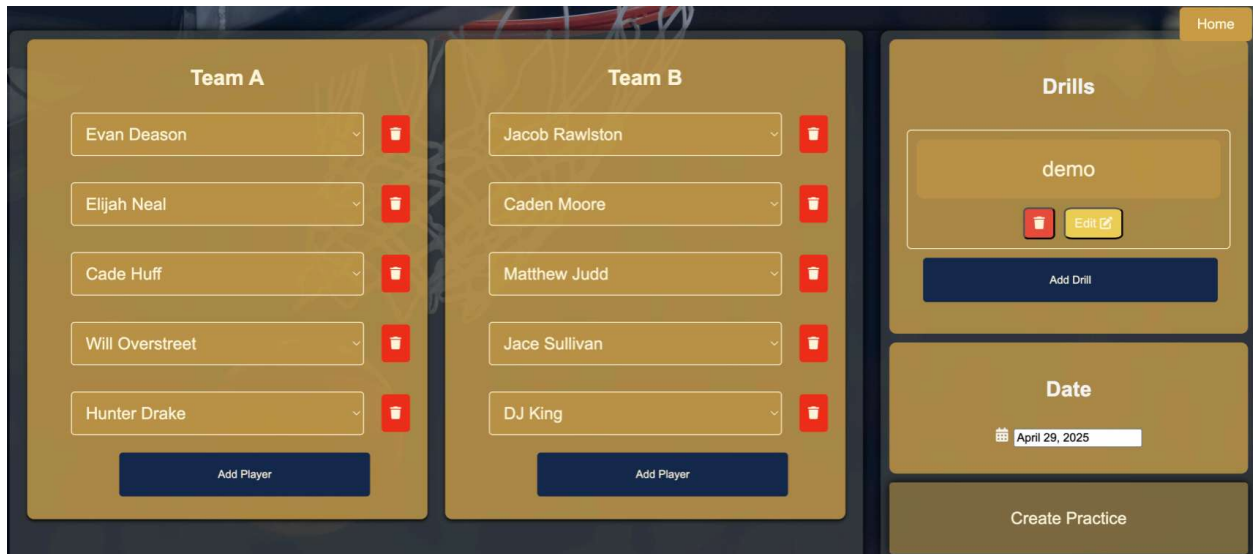


- **Key Features**

- **Practice:** Record and analyze practice stats
- **Game:** Start a new game or load an existing game
- **Player Stats:** Review individual player performance
- **Team Stats:** View aggregated team data and graphs
- **Settings:** Configure the app (export data)
- **Sign Out:** Log out of the app

- **Practice Page**

The Practice Page allows coaches to set up practice sessions by creating Team A and Team B, assigning players, selecting drills, and choosing the practice date. Once a session is created, the app enters tracking mode with stat buttons and a shot zone chart.



- **Key Features**

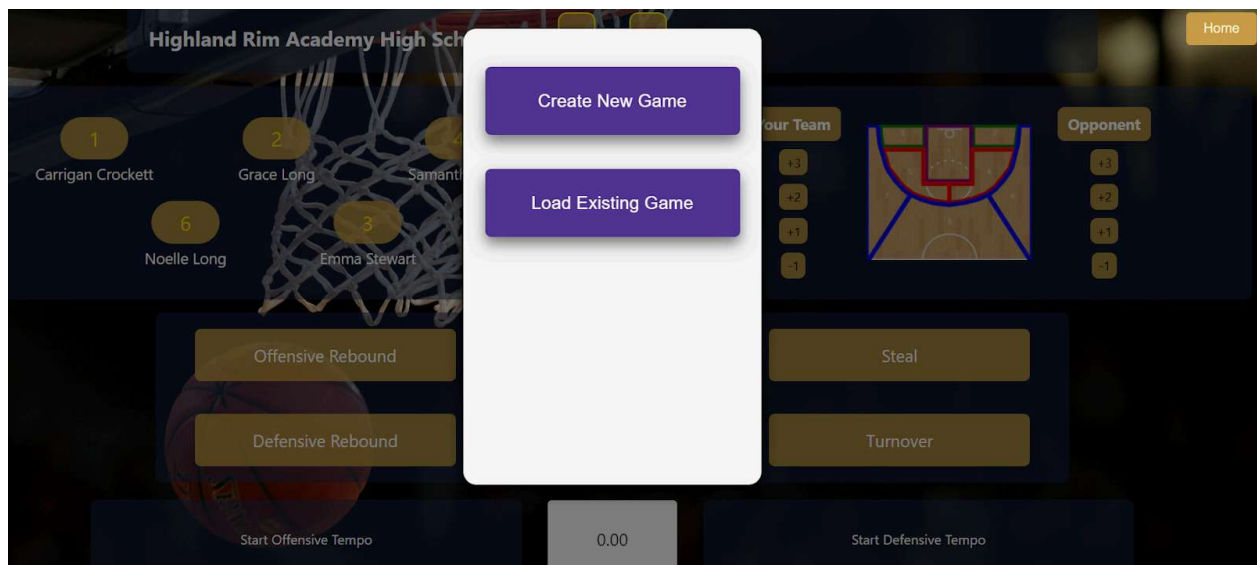
- Add players to either team
- select and edit drills (with option to create one)
- choose practice date
- Track stats (rebounds, assists, steals, block, turnovers)
- Track tempo using start buttons for offense and defense

- **Game Page**

The Game Page is the starting point for recording a basketball session. It allows coaches to either start a new game or load an existing game (such as one from a previous practice or match). This page is optimized for fast in-game access and allows flexible tracking, whether it's a practice session or a live game.

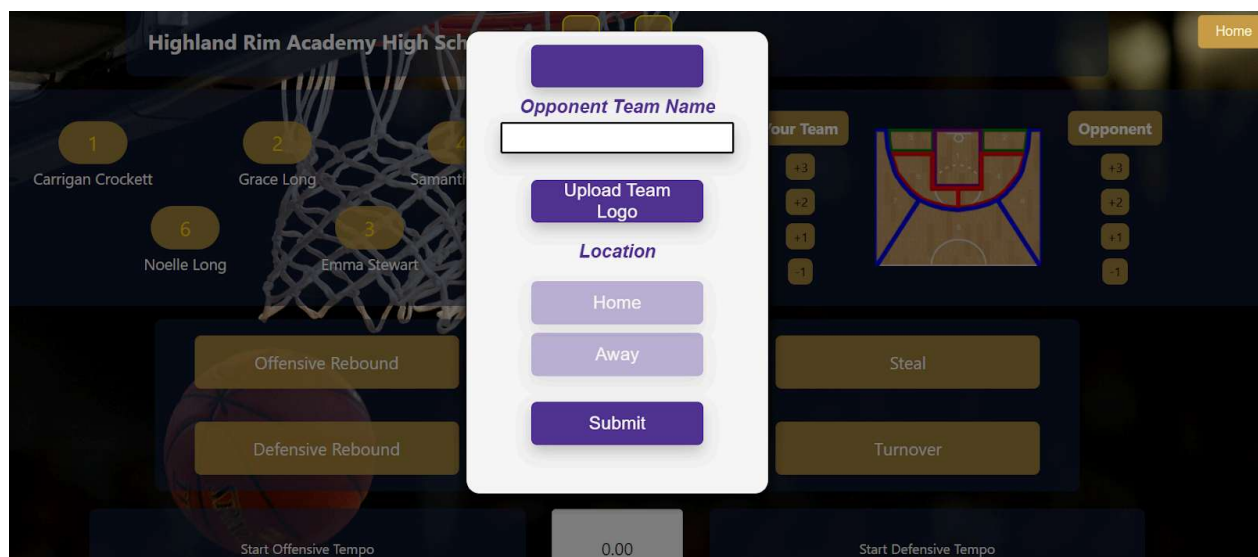
1. Landing View – Choose Game Mode (When a user first visits the Game page, they are presented with two options:)

- Create New Game – Start tracking a brand new game or practice session.
- Load Existing Game – Resume or review data from a previously saved session.



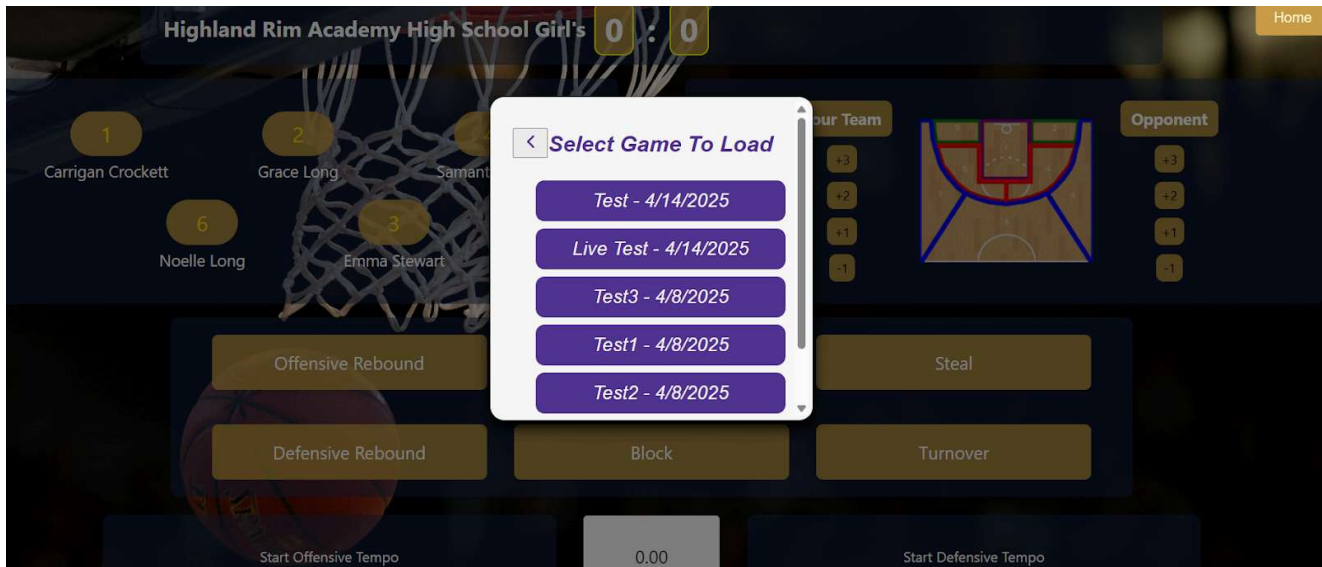
2. Create New Game (If "Create New Game" is selected, a new form is displayed):

- Opponent Team Name – Type in the name of the team you're playing against
- Upload Team Logo – (Optional) Upload a custom logo for the opponent
- Location – Choose between Home or Away
- Submit Button – Confirms the setup and redirects to the live game tracking interface



3. Create New Game (If "Load Existing Game" is selected, the user is shown a list of previously saved games, currently used mostly for testing. Each game entry includes):

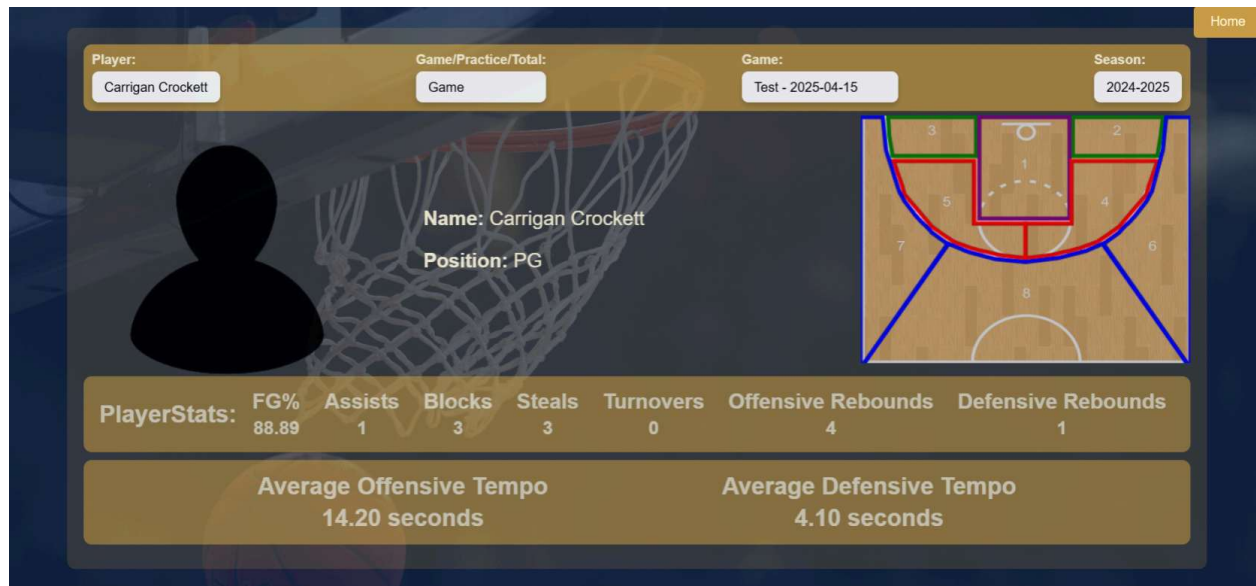
- Game name or ID
- Date and time (if implemented)
- Load button to continue recording or view stats



This allows coaches to revisit older data or resume a session that was paused mid-game

- **Player Stats Page**

The Player Stats Page displays individual player performance for a selected game or practice session.



- **Key Features:**

- Player Selector: Dropdown or input field to select a player by name
- Game or Practice: Toggle to specify whether the data displayed is from a game or practice
- Game Selector: Dropdown to choose which saved session (e.g., "Test - 2025-04-15")
- Season Selector: Dropdown to filter stats based on academic season
- Shot Chart: Court map with zones — used for visualizing shot location and heatmaps
- Player Name/Position: Pulled dynamically from the roster database
- Player Stats Table: Displays key stats like MPG, FG%, 3P%, Assists, Blocks, etc.
- Tempo Averages: Calculates and displays average offensive and defensive tempo (in seconds)

- **Developers Note:**

- The Player Stats page shows player data by pulling information from the backend, based on the player, game, and season selected. Stats like shooting percentage and assists are calculated from the game events. Tempo averages are based on timing between plays.

- **Team Stats Page**

The Team Stats Page provides a high-level overview of the entire team's performance in a selected game or practice.



- **Key Features:**

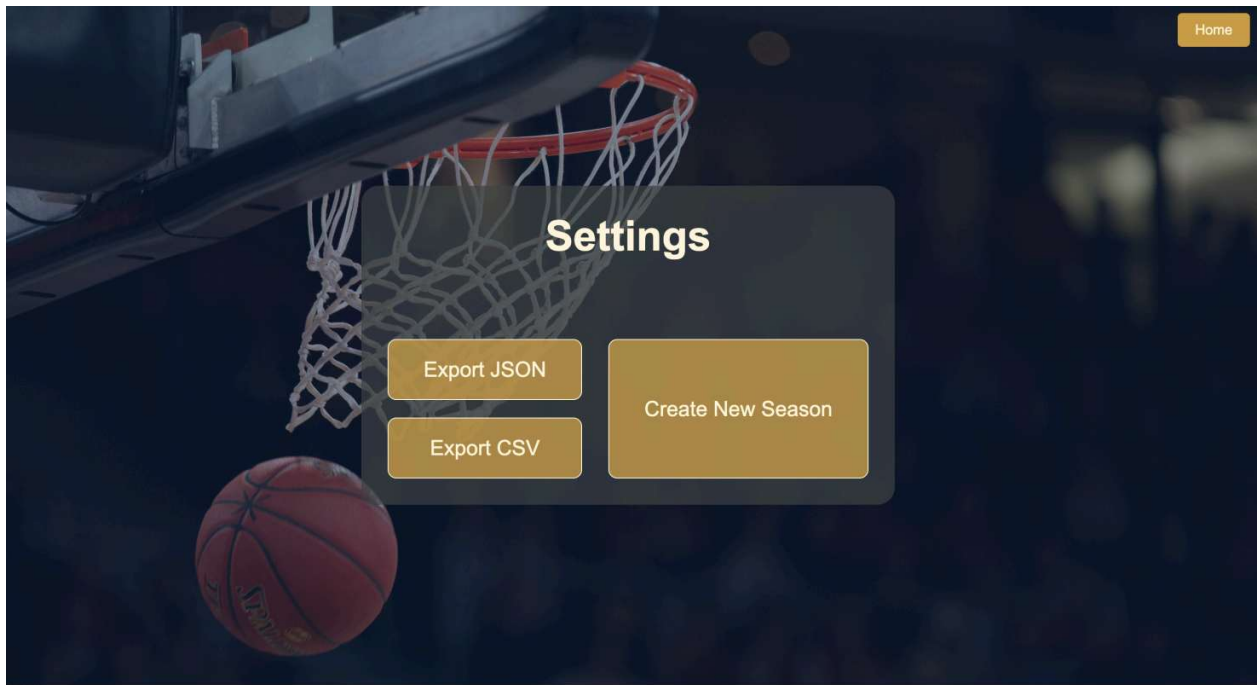
- Season Selector: Choose the academic season (e.g., 2024-2025)
- Game Selector: Pick the specific game or practice session to view stats
- Game Type Selector: Toggle between Game and Practice
- Team Leaders Selector: Highlights top players in points, rebounds, assists, steals, and blocks
- Zone Chart: Visualizes the percentage of made shots by each court zone
- Tempo Stats: Displays average offensive and defensive tempo (sec)

- **Developers Note:**

- The Team Stats page pulls team performance data from the backend based on the selected game and season. It shows top players in different categories, like points and rebounds, by sorting the stats. A bar chart shows shot accuracy by court zones, using data from the backend. Tempo stats are averaged from the timing of plays.

- **Settings Page**

The Settings Page lets coaches or admins manage app setup and download data for external use.



- **Key Features:**

- Export JSON: Downloads all game and player data in `.json` format
- Export CSV: Downloads a spreadsheet-friendly version of the data (`.csv`)
- Create New Season: Resets or starts a new season for stat tracking (data separation)

- **Developers Note:**

- The Settings page connects to backend routes that let users export game data or reset the app for a new season. Export options return structured files (JSON or CSV) that coaches can save or open in Excel. “Create New Season” should either clear or archive previous data. Make sure the page works well on all devices and handles errors like failed downloads.

7. Project Structure

/client	- React frontend
/server	- Node.js backend
/mongo	- MongoDB schemas and config
/nginx-config	- NGINX setup for production
/docker	- Docker & Docker-compose setup

8. Scripts

Script	Description
<code>npm start</code>	Runs the frontend only
<code>npm run dev</code>	Runs frontend + backend together
<code>npm run build</code>	Builds frontend for production
<code>docker-compose up --build</code>	Full production stack with Docker

9. Deployment Instructions

- Update `nginx-config/default.conf` with your server IP or domain:

```
server_name your-ip-or-domain;
```

- Open port 80 on your router or firewall.
- Deploy with Docker:

```
docker-compose up --build
```

10. Github Usage

Once you finish working on your part of the project (for example, adding a new page, fixing something, or updating styles), you should upload your changes to GitHub so the rest of the team can see it.

Here's the basic flow:

- Make sure you are inside the project folder.
- Check what files you changed:

```
git status
```

- Add the files you want to upload:

```
git add .
```

- Write a short and clear commit message:

```
git commit -m "Added Game Page new features"
```

- Push your changes to GitHub:

```
git push
```

- Always **pull the latest code** before you start working, so you don't overwrite someone else's work:

```
git pull
```

11. Future Improvements

- Custom team colors and branding
- Editable court layout
- Support for multiple schools
- Exporting reports to CSV/JSON/PDF
- Role-based login for different types of users