

# Project Documentation: English Premier League Fixtures and Stats Web Application

## Project Overview

**Project Name:** English Premier League Fixtures and Stats

**Description:** A web application built with React and an Express.js backend to display English Premier League (EPL) data, including fixtures, recent results, league standings, and player statistics for the 2024/25 or 2025/26 season. The app fetches data from the Football-Data.org API (or TheSportsDB as a fallback) and is deployed on Heroku for public access.

**Objective:** Provide football fans with a user-friendly interface to view EPL schedules, results, standings, and top scorers, styled with Tailwind CSS and optimized for both local and production environments.

**Technologies:** React, Express.js, Node.js, Tailwind CSS, Axios, Football-Data.org/TheSportsDB APIs, Heroku, Git, GitHub.

**Deployment URL:** <https://pl-fixtures-app.herokuapp.com> (replace with your actual Heroku URL after deployment).

**GitHub Repository:** <https://github.com/tkntsh/football-fixtures> (replace with your repository URL).

**Current Status:** Fully functional locally and deployed to Heroku as of October 24, 2025, 8:10 PM SAST.

---

## Table of Contents

1. Features (#features)
  2. Project Structure (#project-structure)
  3. Installation and Setup (#installation-and-setup)
  4. Running Locally (#running-locally)
  5. Deployment to Heroku (#deployment-to-heroku)
  6. API Integration (#api-integration)
  7. Troubleshooting (#troubleshooting)
  8. Dependencies (#dependencies)
  9. Environment Variables (#environment-variables)
  10. Known Issues and Limitations (#known-issues-and-limitations)
  11. Future Enhancements (#future-enhancements)
  12. Developer Notes (#developer-notes)
-

# 1. Features

- **Fixtures Tab:** Displays upcoming EPL matches with team names, crests, dates, and venues.
  - **Results Tab:** Shows the last 10 completed matches with scores and team details.
  - **Standings Tab:** Presents the current EPL league table with team positions, points, and statistics.
  - **Player Stats Tab:** Lists top scorers (Football-Data.org) or placeholder data (TheSportsDB).
  - **Responsive Design:** Styled with Tailwind CSS for mobile and desktop compatibility.
  - **Error Handling:** Displays loading spinners and error messages for API failures.
  - **Backend Proxy:** Express.js server proxies API requests to bypass CORS and hide API keys.
  - **Heroku Deployment:** Hosted on Heroku for public access.
  - **Environment Variables:** Securely manages API keys using `.env` and Heroku config vars.
- 

# 2. Project Structure

football-fixtures/

```
|— public/
|   |— index.html      # HTML template with favicon and title
|   |— favicon.ico     # Football icon
|— src/
|   |— components/
|       |— MatchCard.jsx # Displays individual match details
|       |— ResultsCard.jsx # Displays individual result details
|       |— LeagueTable.jsx # Displays league standings
|       |— PlayerStats.jsx # Displays top scorers
|       |— LoadingSpinner.jsx # Loading animation
|       |— ErrorMessage.jsx # Error display component
|   |— pages/
|       |— Home.jsx     # Main page with tabs for data display
|   |— App.js           # Core logic for API calls and state management
|   |— App.css          # Global styles
|   |— index.js         # Entry point for React
|   |— index.css        # Tailwind CSS imports
|— .env                 # Environment variables (API key)
|— .gitignore           # Ignores node_modules, build, .env
|— package.json         # Dependencies and scripts
```

— Procfile	# Heroku process definition
— server.js	# Express backend for API proxy and serving React
— postcss.config.js	# PostCSS configuration for Tailwind
— tailwind.config.js	# Tailwind CSS configuration

---

## 3. Installation and Setup

### Prerequisites

- **Node.js:** Version 14.x or higher (`node -v`).
- **Heroku CLI:** Installed (`heroku --version`).
- **Git:** Installed and configured (`git --version`).
- **Text Editor:** VS Code recommended.
- **API Key:** Football-Data.org API key (if used).

### Steps

1. **Clone Repository:**

2. `cmd`

`git clone https://github.com/yourusername/football-fixtures-app.git`

3. `cd football-fixtures`

4. **Install Dependencies:**

5. `cmd`

6. `npm install`

7. **Set Up Environment Variables** (if using Football-Data.org):

- Create `.env` in the project root:
- `env`
- `REACT_APP_FOOTBALLDATA_API_KEY=7e8f6f7708e8426d8289c7615a871bc7`
- Ensure `.env` is listed in `.gitignore`.

8. **Verify Files:**

- Ensure `Procfile` exists:
  - web: `node server.js`
  - Check `package.json` scripts and dependencies (see Dependencies (#dependencies)).
-

## 4. Running Locally

1. **Start Backend:**
2. `cmd`
3. `node server.js`
  - Output: Server running on port 5000
  - Serves API proxy at `http://localhost:5000/api/v4/...` (Football-Data.org) or static files (TheSportsDB).
4. **Start Frontend:**
  - In a new terminal:
  - `cmd`
  - `npm run start:frontend`
  - Opens `http://localhost:3000` in the browser.
5. **Verify Functionality:**
  - **UI:** Check title ("PL-25/26" or "PL-24/25"), favicon, and tabs (Fixtures, Results, Table, Stats).
  - **Console Logs** (F12 > Console):

Fetching fixtures for 2024/25...

Fixtures Response: { matches: [...] }

...

- API Data: { fixtures: [...], recentResults: [...], standings: [...], playerStats: [...] }
- **Network Tab** (F12 > Network): Confirm 200 OK for API requests.

---

## 5. Deployment to Heroku

### Steps

1. **Log in to Heroku:**
2. `cmd`
3. `heroku login`
4. **Create Heroku App** (if not already created):
5. `cmd`
6. `heroku create pl-fixtures-app`

7. **Add Heroku Remote:**

8. `cmd`

9. `heroku git:remote -a pl-fixtures-app`

10. **Set Environment Variable** (if using Football-Data.org):

11. `cmd`

12. `heroku config:set`

`REACT_APP_FOOTBALLDATA_API_KEY=7e8f6f7708e8426d8289c7615a871bc7 -a  
pl-fixtures-app`

13. **Commit Changes:**

14. `cmd`

`git add .`

`git commit -m "Prepare for Heroku deployment"`

15. `git push origin main`

16. **Deploy to Heroku:**

17. `cmd`

18. `git push heroku main`

19. **Verify Deployment:**

- Open app:
- `cmd`
- `heroku open -a pl-fixtures-app`
- Check logs:
- `cmd`
- `heroku logs --tail -a pl-fixtures-app`
- Confirm UI, Console, and Network tab as described in Running Locally (#running-locally).

---

## 6. API Integration

The app supports two APIs, with Football-Data.org as the primary choice and TheSportsDB as a fallback due to free tier restrictions.

### Option 1: Football-Data.org

- **Endpoints:**
  - **Fixtures:** `/v4/competitions/PL/matches?status=SCHEDULED&season=2024`

- **Results:** `/v4/competitions/PL/matches?status=FINISHED&season=2024`
- **Standings:** `/v4/competitions/PL/standings?season=2024`
- **Player Stats:** `/v4/competitions/PL/scorers?season=2024`
- **API Key:** Required, stored in `REACT_APP_FOOTBALLDATA_API_KEY`.
- **Proxy:** `server.js` proxies requests to `https://api.football-data.org/v4/...` to bypass CORS and hide the API key.
- **Limitations:** Free tier restricts 2025/26 season data, rate-limited to 10 requests/minute.

## Option 2: TheSportsDB

- **Endpoints:**
  - **Fixtures/Results:**  
`https://www.thesportsdb.com/api/v1/json/3/eventsseason.php?id=4328&s=2024-2025`
  - **Standings:**  
`https://www.thesportsdb.com/api/v1/json/3/lookuptable.php?l=4328&s=2024-2025`
  - **Player Stats:** Placeholder data (not available).
- **API Key:** Not required.
- **Direct Calls:** No proxy needed; frontend makes direct HTTP requests.
- **Advantages:** No CORS issues, no rate limits, simpler deployment.

## Key Files

- `server.js` (Football-Data.org):
- `javascript`

# 7. Troubleshooting

## Common Issues and Fixes

1. **403 Forbidden (Football-Data.org):**
  - **Cause:** Invalid API key or restricted endpoint (e.g., 2025/26 season).
  - **Fix:** Verify key in `.env` and Heroku (`heroku config`). Switch to `season=2024` or TheSportsDB.
  - **Test:** `curl -X GET "https://api.football-data.org/v4/competitions/PL/matches?status=SCHEДУLED&season=2024" -H "X-Auth-Token: 7e8f6f7708e8426d8289c7615a871bc7"`.

## 2. Empty Data:

- **Cause:** No data for the requested season.
- **Fix:** Use `season=2024` or TheSportsDB with `s=2024-2025`.

## 3. Heroku Deployment Failure:

- **Cause:** Missing Procfile, dependencies, or build errors.
- **Fix:** Ensure Procfile (`web: node server.js`), heroku-postbuild in `package.json`, and check `heroku logs --tail`.

## 4. CORS Issues:

- **Cause:** Direct API calls from frontend (Football-Data.org).
- **Fix:** Use `server.js` proxy or switch to TheSportsDB (no CORS restrictions).

## 5. API Key Not Loaded:

- **Cause:** Missing `dotenv` or `.env` file.
  - **Fix:** Install `npm install dotenv`, add `require('dotenv').config()` in `server.js`, verify `.env`.
- 

# 8. Dependencies

## package.json

```
json
{
  "name": "football-fixtures",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "axios": "^1.4.0",
    "cors": "^2.8.5",
    "dotenv": "^16.0.3",
    "express": "^4.18.2",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "tailwindcss": "^3.3.0"
  },
  "devDependencies": {
    "@types/react": "^18.2.0",
```

```

"@types/react-dom": "^18.2.0",
"autoprefixer": "^10.4.0",
"postcss": "^8.4.0"
},
"scripts": {
  "start": "node server.js",
  "start:frontend": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject",
  "heroku-postbuild": "npm run build"
},
"engines": {
  "node": ">=14.x"
},
"proxy": "http://localhost:5000",
"eslintConfig": {
  "extends": ["react-app"]
},
"browserslist": {
  "production": [">0.2%", "not dead", "not op_mini all"],
  "development": ["last 1 chrome version", "last 1 firefox version", "last 1 safari version"]
}
}

```

---

## 9. Environment Variables

- **File:** `.env` (local) and Heroku config vars.
- **Variables** (Football-Data.org only):
  - `REACT_APP_FOOTBALLDATA_API_KEY=7e8f6f7708e8426d8289c7615a871bc7`
- **Heroku Setup:**
  - `cmd`
  - `heroku config:set`  
`REACT_APP_FOOTBALLDATA_API_KEY=7e8f6f7708e8426d8289c7615a871bc7 -a`  
`pl-fixtures-app`
- **Note:** Not required for TheSportsDB.



---

## 10. Known Issues and Limitations

- **Football-Data.org Free Tier:**
    - 2025/26 season data restricted (403 Forbidden).
    - Rate limit: 10 requests/minute.
  - **TheSportsDB:**
    - No player stats; uses placeholders.
    - Season format (2024–2025) differs from Football-Data.org (2024).
  - **Empty Data:** Possible if no matches are scheduled/finished for the selected season.
  - **Heroku Free Tier:** Limited dyno hours; consider upgrading for production use.
- 

## 11. Future Enhancements

- **Real-Time Updates:** Implement polling or WebSockets for live match updates.
  - **Additional Data:** Add team details, match highlights, or news feeds.
  - **Authentication:** Allow users to save favorite teams or matches.
  - **Improved Styling:** Enhance UI with animations or custom team crests.
  - **Alternative APIs:** Integrate ESPN or SportRadar for richer data.
  - **Testing:** Add unit tests with Jest for components and API calls.
- 

## 12. Developer Notes

- **Developer:** Ntokozo
- **Date:** October 24, 2025
- **Challenges Overcome:**
  - Fixed `MODULE_NOT_FOUND` for `server.js`.
  - Resolved 403 Forbidden error by adding `dotenv` and switching to 2024/25 or TheSportsDB.
  - Configured Heroku deployment with `Procfile` and environment variables.
- **Contact:** Replace with your contact info or repository issues page.
- **Acknowledgments:** Thanks to Football-Data.org and TheSportsDB for free APIs, Heroku for hosting, and Tailwind CSS for styling.

**Tech Stack:**

- **Frontend:** React, Tailwind CSS, Axios
- **Backend:** Express.js, Node.js
- **APIs:** Football-Data.org (primary), TheSportsDB (fallback)
- **Deployment:** Heroku
- **Version Control:** Git, GitHub
- **Environment Management:** dotenv
- **Build Tools:** Create React App, PostCSS, Autoprefixer