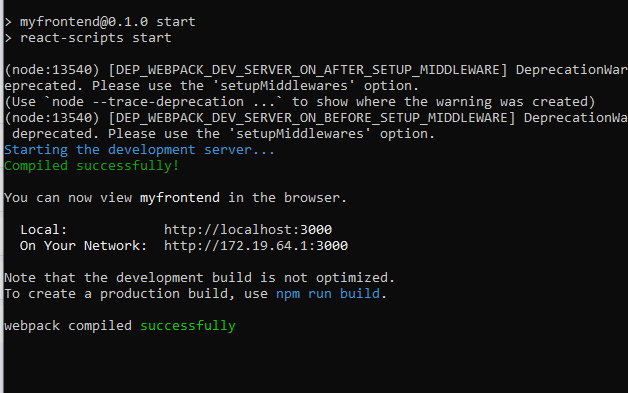
# ABOUT THE WEBSITE

The website I have created is a simple EPL (English Premier League) Results Dashboard. It features a React.js frontend that allows users to view and submit EPL match results.

**Frontend (React.js)**



**Figure 1: Reactjs Server**

Main Components

* ResultList

This is displaying a list of EPL match results retrieved from the Django backend.

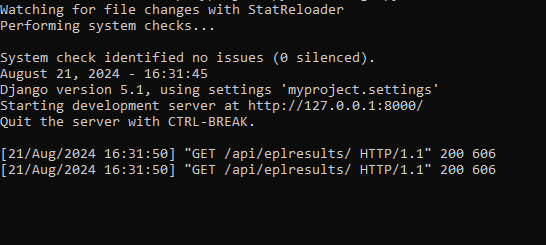
* ResultForm

This component is for allowing users to add new match results by filling out a form.

* User Interface

The interface is straightforward, with a main heading that reads "EPL Results Dashboard". Below it, users can submit new results through the form and view existing results in a clean, organized list.

**Backend (Django):**



**Figure 2: Django Backend Server**

* APIs

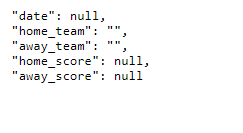
I have created two API endpoints using Django

* A GET API to retrieve the list of EPL matches results.
* A POST API to add new match results to the database.



**Figure 3: Django Rest framework**

**Database**



**Figure 4: Variables in Database Table**

The backend stores the EPL results in a database table, which includes information like the teams that have previously played the score, and the date of the match.

**Deployment**

The website is designed to be deployed on AWS, following a 3-tier architecture:

* Frontend which will be hosted on an S3 bucket or served through a CDN.
* Backend running on an EC2 instance or AWS Lambda.
* Database managed through Amazon RDS or DynamoDB.

This architecture ensures that the website is scalable, secure, and efficient.

