Code Challange – Ntara

**Code Project**

You have been provided with a comma-separated value (.csv) file. This data set should remain unedited. Your mission, if you choose to accept it, is as follows:

* Create a single web page in asp.net C# with a textbox and a button that, when clicked, searches a given dataset and returns the entire record of results on screen
* Include a drop-down that defaults to all columns but can be selected to search only one column
* Provide error checking where appropriate
* Please use a technology framework like Angular, React, or Vue to help showcase how you’ll interact with the frontend.
* You will be evaluated on functionality not UI design.

Additional Requests:

* Please use a Database (SQLite or In-Memory) is sufficient
* Please use Entity Framework to serve the data
* This is to showcase your code, organization, and standards please showcase them to the team
* Your project should work on any developer’s machine when turned in as long as they have Visual Studio and run the project as a web application. If it doesn’t work you won’t pass. It is that important.
* Provide a write-up about your project. What was your approach? Provide things that you might change about the project if given the opportunity. Observations you made to improve the page. Length of time spent coding the application, not documentation.
* Focus on how a user would use the application
* If you’ve been asked to do something you haven’t mastered yet. It is perfectly fine to explain that in your documentation. We love to learn and grow at our company. So, if you are learning something new is not a negative. Just let us know in the write-up.

User Story for the project:

As a sports announcer, I would like to search for football teams and see their statistical records during a game day announcement.

**Candidate:** Edgar Vega  
**Estimated Delivery Date:** Fri, Oct 31, 2025  
**Repository:** <https://github.com/EVegaGT/FootballTeamWinsWithMascots>.

Architecture

For this project, we are following a microservice architecture, separating the backend and frontend into independent, maintainable components. This structure allows for better scalability, cleaner responsibilities, and easier collaboration among multiple developers.

Backend

We created a new ASP.NET Core 8 project to build a RESTful API, following Domain-Driven Design (DDD) and CQRS principles to ensure a clean separation of concerns and maintainability across layers:

* **FootballTeamWinsWithMascots.Api**  
  Contains the API controllers, dependency injection configurations, and logging.
* **FootballTeamWinsWithMascots.Domain**  
  Defines the core business model through entities, interfaces, enums, and value objects. This layer focuses purely on domain logic and remains independent of any external framework.
* **FootballTeamWinsWithMascots.Application**  
  Implements the application logic using the CQRS (Command Query Responsibility Segregation) pattern, powered by MediatR. This layer contains commands, queries, and their respective handlers, ensuring a clear distinction between read and write operations.
* **FootballTeamWinsWithMascots.Infrastructure**  
  Contains Entity Framework Core configurations, the DbContext, repository implementations, and database migrations. This layer manages data persistence and external integration.
* We use SQLite as the database, which is lightweight and perfectly suited for this challenge.
* MediatR is used to handle commands and queries in a decoupled way, promoting scalability and testability.
* xUnit is used for unit testing to ensure the stability and reliability of core services and API endpoints.

Frontend

The frontend is developed as an independent React + TypeScript application. It interacts with the backend API through HTTP endpoints, ensuring a clean separation between presentation and business logic.

* Uses React to build a responsive and dynamic single-page application.
* Integrates Material UI (MUI) to provide a clean, user-friendly, and accessible interface.
* Focuses on simplicity and usability to allow users to easily search and view the dataset.