

# Technical Interview - Football Matches Manager

This document outlines a technical interview for a backend software developer position. The candidate will be tasked with building a simple Node.js application using NestJS that demonstrates proficiency in the following areas:

- **TypeScript:** The project must be written in TypeScript for strong typing and code maintainability.
- **NestJS:** The candidate should utilize NestJS, a popular framework for building scalable Node.js applications.
- **Dependency Injection:** The project should leverage NestJS's dependency injection capabilities to improve code modularity and testability.
- **Data Validation:** The candidate should implement robust data validation for API inputs to ensure data integrity and prevent security vulnerabilities.

## Project Description:

The candidate will build a basic RESTful API using NestJS that manages a collection of football matches. The API should offer the following functionalities:

- **Create a football match:** Allows users to create new match entries with details like home team, away team, date, and time.
- **Get all matches:** Retrieves a list of all the scheduled or completed matches.
- **Get a match by ID:** Retrieves a specific match based on its unique identifier.

## Technical Requirements:

- Use TypeScript for development.
- Implement NestJS for building the API.
- Utilize dependency injection to manage service dependencies.
- Implement strong data validation on API requests using libraries like class-validator or similar solutions (e.g., Zod, Joi).
- Include unit tests for the implemented functionalities using Jest as the testing framework.

## Evaluation Criteria:

The candidate will be evaluated based on the following criteria:

- **Code Quality:**
  - Code clarity, structure, and adherence to best practices.
  - Effective utilization of TypeScript features for type safety and code maintainability.
- **Use of NestJS:** Ability to leverage NestJS concepts like controllers, services, and modules for building the API.
- **Dependency Injection:** Proper implementation of dependency injection to manage dependencies effectively.
- **Data Validation:** Implementation of robust data validation for user input to ensure data integrity.
- **Testing:** Presence of unit tests covering the core functionalities of the API using Jest as the testing framework.
- **Error Handling:** Implementation of proper error handling mechanisms to gracefully handle unexpected situations and provide informative responses to the client. This includes using appropriate HTTP status codes, meaningful error messages, and logging errors for further analysis.

## Deliverables:

- A well-structured and documented NestJS project implementing the functionalities mentioned above.
- Unit tests covering the core functionalities of the API.

## Notes:

- The candidate is free to use any additional libraries or dependencies they deem necessary, but they should be justified and documented.
- This is a high-level overview of the project. The interviewer can provide further details or clarifications during the interview.