

Technical Interview Exercise v3

Your task is to develop a simple web application with an API and data layer using .NET C#, ASP.NET MVC, Web API, and a database or data store, while adhering to Clean Architecture principles and using Test-Driven Development (TDD) methodologies.

Your development should be driven by an informal user story that you will create, and which should be included in your presentation.

The application should allow users to create, read, update, and delete records from the data via the API endpoints. Additionally, you should create a user, login as the user, and ensure that the user information is stored in the data.

To showcase your ability to work with modern data storage systems you cannot use Entity Framework, Dapper, or Mediator to complete this assignment.

The application should include the following components

Database: Create a database or other data storage solution with at least one table/object/container to store data for the application and an additional one to store users. The table/object/container should have a unique identifier (primary key) and at least two other fields.

API: Develop an ASP.NET Web API with endpoints that allow users to perform CRUD operations on the data. Each endpoint should have appropriate HTTP verbs, parameters, and return values. Additionally, a second API should include endpoints for user creation, user login, and authorized and non-authorized endpoints.

Data layer: Develop a data access layer that interacts with the data and provides the necessary CRUD operations for the API endpoints.

Business logic layer: Develop a business logic layer that includes all of the business rules and validation for the application. This layer should be independent of the data layer and the API.

Unit tests: Write unit tests for all components of the application, including the data access layer, business logic layer, and API endpoints.

Front end (optional): Develop a simple front-end interface to consume the API endpoints. The front-end can be developed using any technology or framework of your choice, as long as it runs on a web browser. We will also accept a Postman workspace, or using Swagger.

The application should be fully runnable locally, and Docker is preferred. There is no requirement to deploy any code or leverage any cloud services. We do NOT expect your sample

to ship with a functional database, we should be able to deploy the database or data storage using your docker or other build resources.

Presentation and Code Review

Once you have completed the exercise, you will be required to present your project to the technical interview panel. During the presentation, you should explain your user story, design choices, the technical architecture, and demonstrate the functionality of the application. This will be done over Zoom and you will screen share either your GitHub repository or IDE.

After the presentation, the interview panel will conduct a code review of your project. You will be asked to explain your coding decisions and answer any questions related to the code. The interview panel will evaluate your project based on the following criteria:

Clean Architecture: Your architecture should adhere to Clean Architecture principles, including separation of concerns and independence of components.

Test-Driven Development: Your project should follow TDD methodologies and include unit tests for all components.

Code quality: Your code should be well-organized, readable, and adhere to best practices.

Functionality: Your application should perform the required CRUD operations and user authentication without errors or bugs.

User Story: Your user story should drive the development of the application and be included in your presentation.

Presentation: Your presentation should be clear, concise, and demonstrate a good understanding of the project.

Scoring

You will be scored based on your adherence to Clean Architecture principles, TDD methodologies, code quality, user story, and code review remarks. Each criterion will be weighted equally. The maximum score for the exercise is 100 points.

We hope this exercise challenges your technical abilities and helps us assess your suitability for the intermediate full-stack software developer position. Good luck!