**Full Stack Exam (ReactJS + .NET)**

This practical exam attempts to test your ability to implement a simple React application that interacts with a REST Web API. The important aspect is to determine how well you can work with both solutions. We will attempt to build and run your completed exam, so please test it!

This exam consists of three parts. (Frontend, Backend, and Integration)

**Git Repository**

1. You can use our GitHub repository with a starter pack code base or create your own project.

Fork our GitHub repository from our starter pack code base.

* + 1. Repo: <https://github.com/DSS-Specpoint/contacts-directory>

**React-Side/Front-End**

1. **Required:** Write the code using TypeScript.
2. Create a form component for the ‘Contact’ that can be used for both creation and modification.
3. Create a list component to display ‘Contacts’.
   1. Use whatever ‘Contact’ fields you think appropriate.
   2. The Contacts data should come from your backend API.

**.NET Side/Backend**

1. Add an API controller class Contact Controller that will accept appropriate CRUD methods.
   1. Set up appropriate and intuitive routing for the controller.
2. Add a model class Contact, and appropriate fields for a Contact found in most applications.
3. Add an **in-memory** database context using entity framework.
   1. Please use an in-memory database so that we can run your solution locally.
   2. Generate backend default data (10 contacts)

**Integrate**

1. Integrate the backend API to the frontend react app, so it responds properly for each action that occurs in the application.
2. The Frontend should call the Backend API to fetch data. Use async/await and try/catch blocks as necessary.
3. It should also Create/Delete/Update the in-memory data on the backend for each action that is executed.

Use any remaining time to add some flair and show off a bit. Some ideas: Create unit tests; validate input; add a search; improve the styling; use GraphQL instead of REST. Impress us, we will be paying attention.

Coding is 90% Reading, and 10% coding. So, we hope you code like you are telling us a story (through comments on why and how you come up with the solution). We’d appreciate it. We are looking for your “personal best practices / personal coding standards” so let us know who you are when we read your code. Good luck! Enjoy the coding and the learning!