Phonebook Web Application

Create a phone book application for searching, displaying and updating contact information. Searchable fields will be First Name, Last Name, Email, Phone.

You can create this application using any UI technology you are comfortable with ie.. Razor Pages, Blazor, React, Angular, Vue etc.   It must have a separate API using .NET Core. The API must connect to a Postgres database. The stack needs to be deployed using docker.

There is no need for authentication.

We are interested in your approach and coding skills – we are not expecting you to finish the task and understand that you may run out of time. Think of this task as a showcase of your talents across backend, mid-tier and presentation layers.

We are also looking for how you interpret these very loose requirements – be prepared to justify any choices you make.

We are looking for:

- Clean, concise and easy to understand code
- Comments where necessary
- Appropriate use of frameworks and technology
- Good OOD and separation of concerns
- Recognised architectural patterns and techniques

What would be nice to see:

- AJAX driven updates/search
- JavaScript/JQuery with some nice looking CSS.
- If creating the database manually, would like to see sql scripts for creating the database, tables, test records. If using an ORM and EF code first, it must setup the DB with test data.
- Testable architectures and patterns.
- Some unit tests on the api using xUnit to test functional logic. Don't test controller actions.
- Dockerfile for each of the frontend, api, postgres
- Docker compose file to automatically create these containers into a docker stack

The application should support as many of the following features as you can in the time you have available:

- Ability to search for a specific name matching beginning or anywhere in the name
- Ability to view contact details of a user
- Ability to Update a contact in the database
- Ability to Add a new contact to the database
- Ability to Delete a new contact from the database with a confirmation step
- Paginate search results allowing users to page through results
- Basic validation on CRUD operations where appropriate

Keep it simple and don't over engineer! Good luck!