

## **TOPO SOLUTIONS**

### **CODING CHALLENGE**

As part of your challenge we want you to design and develop a simple web contact manager. The solution should be composed at least by a stand-alone REST API project, and a front-end project. For the sake of the test, we don't expect User authentication and authorization to be implemented. In the REST API project, we do expect a service authentication, such as ApiKey, or OAuth.

### **REQUIREMENTS**

- Contact must have a unique Id that will remain the same over time.
- Contact fields include: First Name, Last Name, Phones, Addresses, Emails, Categories, Organisation Name, WebSite Url, Notes, and Tags.
- Contact must have a computed name as concatenation between First Name and Last Name.
- Contact Notes is a open-text field that must accept up to 200 characters.
- Contact Name must be filled.
- Validate Phone.
- Validate Invalid Email.
- Validate Invalid WebSite.
- Contact phones must be a number field.
- Contacts must accept custom fields defined by the User.
- User must be able to download or upload their Contacts in VCard format. When uploading their Contact List, the system must decide which contact to Add, Skip or Update.
- The UI must allow select Contacts to be downloaded in VCard format.
- The UI must have an option to import Contacts in VCards format.
- The UI must allow the User Insert, Edit, Delete, Visualize contacts, and Search Contacts by Name, Phone, E-mail, Categories, WebSite Url, and Notes.
- The UI must group Contacts by Tags.
- The UI must display Contacts as cards, and once opened The UI should display a Form to Edit, Visualize or Delete the Contact.
- The UI must validate required fields and display a friendly validation message.
- The UI must format phone numbers.

### **RESTRICTIONS**

Use your favorite coding language for this challenge, we prefer C#. You may use any data store of your choosing. You may also use any outside libraries in support of your solution, but the logic and the model structure must be all your own. You can use any method you would like to test your solution, we prefer a suit of unit tests and integration tests.

### **DELIVERABLES**

- Please give us access to your repository.
- Step-by-steby instructions detailing how to get your solution up and running.
- Provide a sequence diagram from request to response.
- Documentation that lists out any configurations your solution may have.
- The schema of your data model.

## EVALUATION

We will evaluate your submission based on the following:

- Correctness
- Code quality, readability, testability, and extensibility.
- Time is not a major factor, but please let us know how long you took.
- Be prepared to participate in a code review to discuss your solution.
- Be prepared to also go through a brainstorming session that extends upon your solution.

Happy Coding! 😊