

Indio - Coding Challenge

Summary

We're excited to see your skills! In order to get a better idea of your programming abilities, we'd like you to complete the following sample project. You should be able to complete this project in under 5 hours.

Project Overview

One of the largest problems in the insurance industry today is the pure amount of client data that is stored in an inconsistent manner. Between 1000s of fillable PDFs, archaic online forms, and old databases, there's a real big need for standardization.

In order to better gather data from clients, we've built a complex form builder to handle these structures. For this project, you'll be building a simple front-end version of our form builder that will generate a form for users to fill out and allow the user to preview it. We'll also want the ability to export this structure as JSON.

Detailed Spec

[View Project Wireframe](#)

You'll see in the above wireframe that there are 3 types of form inputs. Each of these can also have sub-inputs which will only show when the parent input is answered a certain way. The types of conditions are as follows

- Text
 - Equals - Text entered is equal to this value
- Number
 - Equals - Number entered is equal to this value
 - Greater than - Number entered is greater than this value
 - Less than - Number entered is less than this value
- Yes / No (radio)
 - Equals - Radio selected is equal to this value (either yes or no)

The user should be able to keep creating sub-inputs with conditions to as many levels deep as they would like. Each sub-input's condition will correspond to the value of the parent input.

By default, the create tab should start out blank with just the Add Input button there for the user to create their first input.

Upon clicking the preview tab, the user should be presented with a working demo of the form based on the structure they created in the Create tab. When the preview tab is loaded, it should render only the parent level questions until the user answers questions. Based on the users answers to questions, the next level of conditional questions should be shown.

The preview tab should re-render every time it is clicked with no persistence.

The export tab should simply be a textarea containing a serialized JSON form of the form object. This should also be stored in the local storage for persistence and loaded at page load.

Technical Requirements

The project should store a persistent state for the form objects in the local storage as we aren't using a backend or any external APIs. The entire app will live on the front-end without a backend.

The final deliverable should be a github repo or zipped project that should contain everything needed to easily run the project locally with a few commands (npm install, npm start) and view the result at localhost:8080.

You may use webpack, browserify, or any similar library for easily running your project. Feel free to use a boilerplate setup as well so you can get to the actual development as soon as possible. You may use either Angular or React with either JSX, Typescript, or ES6/7.

Evaluation

The purpose of this project is to evaluate your skills on the following:

- Ability to work off of a wireframe and create user friendly UX
- Designing data structures for use in multiple modes:
 - Generating API friendly JSON from an object model
 - Rendering an interface based on an object model
 - Editing and deleting objects within a structure
- Understanding in how to navigate a recursive data structure
- Knowledge of front-end tooling

Upon delivery of this coding challenge, we'll schedule a 15-25 minute call to walk through what you've built.