# **Frontend Developer Code Assignment**

## **Use Case: Smart Insurance Application Portal**

### **Introduction**

Imagine we are building a **Smart Insurance Application Portal**, where users can apply for different types of insurance (**Health, Home, Car, Life, etc.**) through a **dynamic form**.

* Each **insurance type** has **different questions** based on the user’s responses.
* The forms are **fully dynamic**, meaning they are **fetched from an API** and built **without hardcoding** the structure.
* Users can **view, filter, and manage their applications** in a **customizable list view**.

### **Your Task**

Develop a **React.js web application** that:  
✅ Fetches **dynamic form structures** from an API.  
✅ Renders forms dynamically with **conditional logic** (fields appear/disappear).  
✅ Allows users to **submit applications** and save data.  
✅ Displays submitted applications in a **customizable list** with **sortable, filterable, and selectable columns**.

# **1. Features & Functionalities**

## **1.1 Smart Dynamic Forms**

📌 The application should fetch the form **structure from an API** (no hardcoded forms).  
📌 **Conditional fields** should appear/disappear based on user input.  
📌 Some form fields will have **nested sections** (e.g., Address, Vehicle Details).  
📌 Some fields will **dynamically fetch options from an API** (e.g., states based on country).  
📌 Form data will be **validated** before submission.

### **Example Conditions:**

* **Home Insurance**: "Security System Type" appears only if "Do you have a security system?" is "Yes".
* **Car Insurance**: "Number of Accidents" appears only if "Have you had any accidents?" is "Yes".
* **Health Insurance**: "Pregnancy Status" appears only if "Gender" is "Female".

## **1.2 Customizable List View**

📌 Users can view **submitted applications in a table**.  
📌 Users can **select which columns** to display dynamically.  
📌 List should support **sorting, searching, and pagination**.

### **Example UI:**

| **Name** | **Age** | **Insurance Type** | **City** | **Status** |
| --- | --- | --- | --- | --- |
| John Doe | 28 | Health | New York | Pending |
| Jane Smith | 32 | Life | LA | Approved |
| Alice Brown | 40 | Car | Chicago | Rejected |

# **2. API Endpoints**

## **2.1 Fetching a Dynamic Form**

GET /api/insurance/forms

* Returns the **form structure**.

## **2.2 Submitting the Form**

POST /api/insurance/forms/submit

* Sends the filled **form data** to the backend.

## **2.3 Fetching Submitted Applications**

GET /api/insurance/forms/submissions

* Returns the **list of submitted applications**.

#### **Example API Response for List View**

{

"columns": ["Full Name", "Age", "Insurance Type", "City", "Status"],

"data": [

{

"id": "1",

"Full Name": "John Doe",

"Age": 28,

"Insurance Type": "Health",

"City": "New York",

"Status": "Pending"

},

{

"id": "2",

"Full Name": "Jane Smith",

"Age": 32,

"Insurance Type": "Life",

"City": "Los Angeles",

"Status": "Approved"

}

]

}

BASE URL: https://assignment.devotel.io

# **3. Evaluation Criteria**

✅ **Handles dynamic fields & dependencies correctly.**✅ **Manages API calls properly.**✅ **Implements a column selector for the list view.**✅ **Ensures good UI/UX with responsive design.**✅ **Uses React best practices (Hooks, Context API, or Redux if necessary).**

# **4. Bonus Features**

⭐ **Autosave drafts before form submission.**⭐ **Dark Mode toggle.**⭐ **Localization support (multi-language).**⭐ **Unit tests with Jest/React Testing Library.**⭐ **Drag-and-drop field reordering.**

# **5. Submission Instructions**

1. **Fork the repository (if provided) or create a public GitHub repository.**
2. **Include a README.md** explanation:
   * Setup instructions.
   * API usage details.
   * Any assumptions made.
3. **Deploy to Vercel/Netlify** (optional but preferred).
4. **Submit the repository link.**