

Documentation for UI-Search Content - assessment - by Rebal M

This documentation provides step-by-step instructions for setting up, deploying, and configuring the UI-Search Content application. It includes all commands, configurations, and dependencies required to run the application.

Overview

This React-based web application enables users to search, filter, and manage contact details. The key features include:

- Search filters for contact fields such as name, email, phone, address, etc.
 - A paginated results table.
 - Ability to select and display details of a specific contact.
-

Features

1. **Search Filters:** Users can search for contacts using parameters like name, email, or address. etc.
 2. **Results Table:** Displays contact results with pagination and selection functionality.
 3. **Selected Contact:** Displays detailed information about the selected contact.
 4. **Pagination:** Allows navigating through the results with "Next" and "Previous" buttons.
-

Prerequisites

Before setting up the application, ensure that the following tools are installed:

1. **Node.js** (v16 or higher) and **npm** (comes with Node.js)
 2. **React Developer Tools** (optional but recommended for debugging)
 3. I personally used VS code for my editor.
-

Setup Instructions

1. Clone the Repository

First, clone the project repository to your local machine using Git:

```
bash
git clone <repository-url>
cd <repository-folder>
```

2. Install Dependencies

Run the following command to install all necessary dependencies specified in the `package.json` file:

```
bash/cmd line (with in terminal):
npm install
```

3. Install React-Bootstrap (this project I made use of Bootstrap for easier integration with CSS for styling)

This application uses **React-Bootstrap** for styling. So, please download Bootstrap components, install it using the following commands:

```
bash
npm install react-bootstrap bootstrap
```

Add the following line to the `index.js` or `App.js` file to include the Bootstrap CSS

4. Start the Development Server (to Run the code for this Project)

Run the following command to start the application:

```
bash
npm start
```

This will launch the application in your default browser at <http://localhost:3000>.

PLEASE MAKE SURE THAT USE ARE NOT USING HOST 3000 for any other projects or file before beginning.

File Structure

Here's a brief overview of the project file structure:

```
root/
|
├── public/                # Public assets (e.g., favicon, index.html)
│   └── mockData.json
├── src/                   # Source files
│   ├── components/       # Reusable React components
│   │   ├── ResultsTable.js # Displays paginated search results
│   │   └── SearchFilters.js # Search input fields for filtering
├── contacts
│   ├── SelectedContact.js # Displays the selected contact details
│   ├── App.js             # Main application component
│   ├── index.js           # Application entry point
│   └── App.css            # Custom CSS styles
|
├── package.json           # Project metadata and dependencies
└── README.md             # Project documentation
```

Start Development Server:

bash

npm start

Install Dependencies:

bash

npm install

Install React-Bootstrap:

bash

npm install react-bootstrap bootstrap

Build for Production:

bash

npm run build

TESTING/ Explanation of application:

let's discuss why React was used for this assessment, react is a javascript tool that allows for building dynamic and interactive user interfaces which are perfect for the given front-end requirements:

So, for the following application in React, I made use of:

- searchBar for inputs.
- ResultsTable for displaying results.
- SelectedContact for showing selected details.
- Use useState for managing search criteria and results.
- Use useEffect to re-filter results when search criteria change.

1. **Add Contacts Data:** Update the Sample contacts data in the public folder of the mockData.json file in the project application. There are about 5 data samples for use.
2. **The current set pagination limit is 3, but can be changed depending on requirements.**

- **Change Pagination Limit:** To modify the number of items per page in

`ResultsTable.js`:

Search for the line:

```
const itemsPerPage = <desired-number>;
```

The code was not built into a production-ready build as it was not required for the assessment, but I included the steps just in case, to build the Application: Run the following command in the terminal always make sure user in the project folder PATH::

bash/cmd line:

```
npm run build
```

1. **Serve the Build:** Use any static server to serve the `build` folder. For example:

Using `serve`:

bash /cmd line:

```
npm install -g serve
```

```
serve -s build
```

Troubleshooting - Just in case

1. Port Already in Use

If the default port `3000` is already in use, you can run the app on a different port:

```
bash
PORT=3001 npm start
```

2. Dependency Issues

If you encounter issues while installing dependencies, try clearing the `node_modules` folder and reinstalling:

```
bash
rm -rf node_modules package-lock.json
npm install
```