**Contact Us Form - Frontend (ReactJS)**

This is the frontend for the Contact Us form, built using React JS. The form allows users to submit their name, email, website URL, and project details. It is styled to be responsive and includes form validation, error handling, and animations to enhance the user experience.

https://github.com/Tejas2569/Contact-Us.git

**Table of Contents**

- [Project Overview](#project-overview)

- [Technologies Used](#technologies-used)

- [Setup and Installation](#setup-and-installation)

- [Folder Structure](#folder-structure)

- [Running the Project](#running-the-project)

- [Libraries Used](#libraries-used)

- [Form Validation](#form-validation)

- [Animations](#animations)

- [Contributing](#contributing)

**Project Overview**

This project is the frontend part of a full-stack Contact Us form application. It is built with React and allows users to fill out a form with their contact information. The form data is validated on the client side using Yup and submitted via an Axios request to a backend server. Animations are added to improve the user experience.

**Features**:

- Responsive design that adapts to desktop, tablet, and mobile screens.

- Form fields with client-side validation (name, email, project details).

- Animations for form transitions and button hover effects.

- Form submission with error handling.

**Technologies Used**

React.js: A JavaScript library for building user interfaces.

Redux Toolkit: For managing form state and handling form submissions.

Axios: For making HTTP requests to interact with the backend API.

Yup: For client-side form validation.

CSS: For styling the application, ensuring responsiveness and animations.

**Setup and Installation**

**1. Clone the Repository**

Clone the repository to your local machine:

git clone https://github.com/your-username/contact-us-form.git

cd contact-us-form

**2. Install Dependencies**

Navigate to the frontend directory and install the necessary dependencies:

cd frontend

npm install

**3**. **Start the React Development Server**

Once the dependencies are installed, you can start the development server:

npm start

This will start the React application at `http://localhost:3000`.

**Folder Structure**

Here’s an overview of the folder structure for the **frontend**:

frontend/

├── public/ # Static files (index.html, images, etc.)

└── src/

├── assets/ # Static assets like images or icons

├── components/ # React components

│ └── ContactForm.js # Contact form component

├── redux/ # Redux slices and actions

│ └── formSlice.js # Redux slice for handling form state

├── services/ # API calls (Axios instance)

│ └── api.js # Function to interact with the API

├── App.js # Main React component

├── index.js # Entry point for React application

└── styles/ # CSS styles

└── ContactUsPage.css

**Key Files**:

App.js: The main React component that renders the Contact Us form.

ContactForm.js: Contains the actual contact form component, handles user input, and dispatches Redux actions.

formSlice.js: Redux slice to manage the form’s state, including form data and form submission status.

api.js: Defines the API endpoint for submitting the form data using \*\*Axios\*\*.

ContactUsPage.css: CSS file for styling the Contact Us page and ensuring it’s responsive.

**Running the Project**

To run the frontend project:

1. Ensure you're in the `frontend` directory.

2. Install dependencies by running:

npm install

3. Start the development server:

npm start

4. Visit `http://localhost:3000` in your browser to view the Contact Us page.

**Libraries Used**

Frontend Libraries:

React: Used for building the user interface and managing state.

Redux Toolkit: For managing global state and handling asynchronous actions (such as form submissions).

Axios: For making HTTP requests to the backend API.

Yup: For validating form data before submission.

React-Spring or CSS Animations: For adding subtle animations to the page and form elements.

**Form Validation**

The form validation is implemented using **Yup**. The following validation rules apply:

1. **Name**: Must be a non-empty string.

2. **Email**: Must be in a valid email format.

3. **Website URL**: Optional, but if provided, it must be a valid URL.

The form is only submitted when all fields are valid.

**Example of Validation**:

Js

const validationSchema = Yup.object({

name: Yup.string().required('Name is required'),

email: Yup.string().email('Invalid email address').required('Email is required'),

website\_url: Yup.string().url('Invalid URL').optional(),

project\_details: Yup.string().min(20, 'Project details must be at least 20 characters').required('Project details are required'),

});

**Animations**

We use CSS to add subtle animations to enhance the user experience:

* **Page transitions** (fade-in) for a smoother user experience.
* **Form field transitions** to animate the input fields as they appear.
* **Button hover effects** to give the user visual feedback when hovering over the submit button.

Css

/\* Fade-in effect for form elements \*/

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

.fade-in {

animation: fadeIn 1s ease-in;

}

/\* Button hover effect \*/

button:hover {

background-color: #4CAF50;

cursor: pointer;

}

**Contributing**

Contributions are welcome! If you want to improve the project or fix any bugs, feel free to fork the repository and submit a pull request. You can also open an issue if you find any problems or have suggestions.

**License**

MIT License.

Notes:

` Tejas2569` in the clone command with your actual GitHub username.