

Recursive Folder Explorer

Project Overview

The Recursive Folder Explorer is a web application developed to manage nested folder structures, emulating a basic file explorer. Users can create, view, and organize files and folders in a recursive, hierarchical view. The application supports adding new files and folders, deleting existing ones, and viewing the contents of nested folders with intuitive folder and file icons.

Technologies Used

- **React:** For building the user interface using reusable, functional components.
- **Context API:** For managing and sharing state across components without prop drilling.
- **JavaScript (ES6+):** Core programming language used to implement the application's logic and functionality.
- **CSS:** Styling to enhance UI/UX, providing a clean, accessible layout and responsive design.

Project Structure and Components

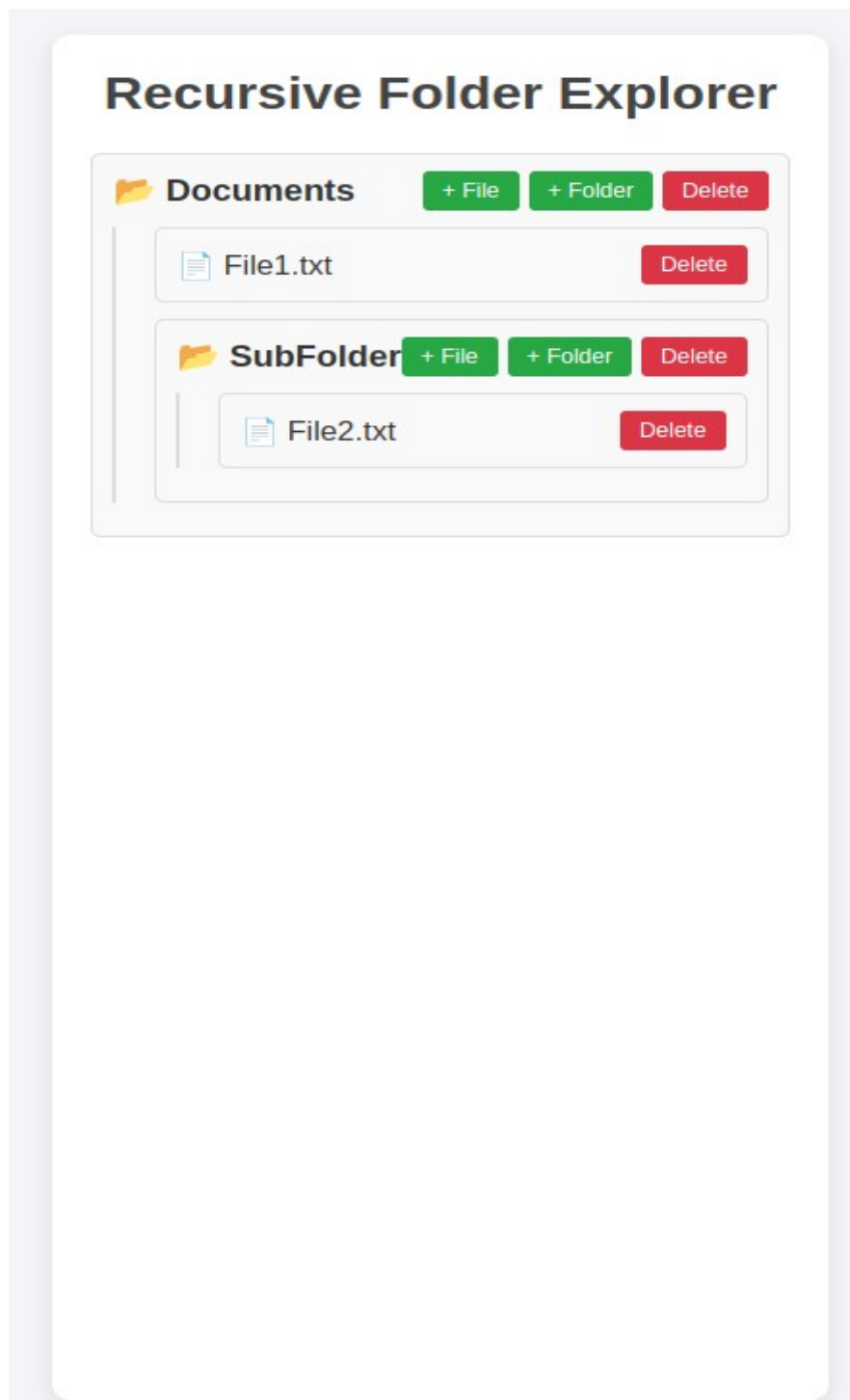
The project is divided into several key files:

1. **App.js:** The main application component where the `FolderProvider` context is provided. It houses the `FolderTree` component that maps and displays folders and files from the root structure.
2. **Folder.js:** This component represents each folder and file item, displaying them with icons. It includes interactive buttons for adding files/folders and deleting items, with recursive functionality to manage nested folders.
3. **FolderContext.js:** Contains the core logic for managing state using React Context. It provides `addItem` and `removeItem` functions to update the folder structure and share data across components.
4. **App.css:** Styling file to enhance the visual design, with folder/file icons, padding, borders, and background colors to create a user-friendly interface.

Workflow and Functionalities

1. **Launching the App:** Run the app locally using `npm start`, and navigate to `http://localhost:3000` to view it in the browser.
2. **Folder and File Management:**
 - **View Folder Contents:** Click on any folder icon to expand or collapse its contents.
 - **Add New Items:** Click "+ File" to add a file or "+ Folder" to add a folder within any folder.
 - **Delete Items:** Use the "Delete" button beside any item to remove it from the structure. Nested items will be deleted recursively.

3. **Recursive Display:** The folder structure is displayed recursively, allowing each folder component to contain other folders or files. This is achieved through recursive rendering within `Folder.js`, providing an organized, tree-like view.



Challenges Faced

Implementing the recursive structure and managing state updates across deeply nested components was a critical challenge. The Context API proved essential in keeping the code clean and efficient while minimizing prop drilling.

