Copilot

**Project Overview**

**Resume Builder Application using React**

**Features:**

1. **Two Layouts:**
    - **Editor Layout:** Left side for users to fill in resume details.
    - **Preview Layout:** Right side for live resume preview.
2. **State Management:**
    - Use **TanStack Store** for state management.
3. **Save and Restore Functionality:**
    - **Save Button:** Stores resume data in local storage.
    - **Restore Data:** On revisiting, restores data from local storage to editor fields and updates preview.
4. **Drag and Drop Reordering:**
    - Reorder sections (Work Experience, Education, Skills, Awards, Volunteering) using drag-and-drop.

**Sections and Fields:**

1. **Basic Information Section:**
    - **Name:** Input for full name.
    - **Contact Information:** Fields for phone number, email, location.
    - **Professional Title:** Input for current job title or desired position.
2. **Work Experience Section:**
    - **Company Name:** Input for company name.
    - **Job Title:** Input for job title.
    - **Start and End Dates:** Date pickers for employment dates.
    - **Job Description:** Text area/rich text editor for job responsibilities and achievements.
    - **Current Job Indicator:** Checkbox/switch for current job status.
    - **Drag and Drop Reordering:** Reorder work experience.
3. **Education Section:**
    - **Institution Name:** Input for educational institution name.
    - **Degree:** Input for degree/certification.
    - **Field of Study:** Input for field of study.
    - **Start and End Dates:** Date pickers for education dates.
    - **Current Study Indicator:** Checkbox/switch for current study status.
    - **Drag and Drop Reordering:** Reorder education entries.
4. **Skills Section:**
    - **Skill Name:** Input for skill name.
    - **Proficiency Level:** Option to add proficiency level (beginner, intermediate, expert).
    - **Drag and Drop Reordering:** Reorder skills.
5. **Summary or Objective Section:**
    - **Summary:** Text area/rich text editor for brief summary or career objective.
6. **Awards and Achievements Section:**
    - **Award Title:** Input for award/achievement title.
    - **Awarder:** Input for awarding body/organization.
    - **Date:** Date picker for award date.
    - **Description:** Text area/rich text editor for award description.
    - **Drag and Drop Reordering:** Reorder awards.
7. **Volunteering Section:**
    - **Organization Name:** Input for organization name.
    - **Role:** Input for role/position.
    - **Start and End Dates:** Date pickers for volunteering dates.
    - **Description:** Text area/rich text editor for volunteering description.
    - **Current Volunteering Indicator:** Checkbox/switch for current volunteering status.

**Additional Features:**

1. **Template Selection:**
    - Choose from multiple resume templates.
    - Preview selected template.
2. **Export and Import Functionality:**
    - **Export to PDF:** Export resume as PDF.

    - **Export to JSON:** Export resume data as JSON.
    - **Import from JSON:** Import resume data from JSON.

3. **Error Handling:**

    - Error boundaries to catch and display errors gracefully.

**Implementation Plan:**

1. **Set up the project:**

    - Initialize a new React project.
    - Install necessary dependencies (TanStack Store, React DnD, etc.).

2. **Create the Layouts:**

    - Implement the Editor Layout with all required sections.
    - Implement the Preview Layout for resume preview.

3. **State Management:**

    - Use TanStack Store for state management.
    - Implement actions to update, add, remove, and reorder sections.

4. **Save and Restore Functionality:**

    - Implement the “Save” button to store data in local storage.
    - On component mount, check for data in local storage and populate editor fields and preview.

5. **Drag and Drop Functionality:**

    - Use React DnD for drag-and-drop reordering of sections.

6. **Export and Import Functionality:**

    - Implement functions to export and import resume data in JSON format.
    - Implement function to export the resume as a PDF.

7. **Error Handling:**

    - Implement error boundaries to catch and display errors gracefully.

**Initial Implementation Plan:**

1. **Set up the Project:**

    - Initialize a new React project.
    - Install necessary dependencies:
        - TanStack Store for state management.
        - React DnD for drag-and-drop functionality.
        - Tailwind CSS for styling.
        - html2pdf.js for exporting to PDF.

2. **Create the Editor Layout:**

    - Create components for each section:
        - Basic Information: `components/BasicInformation.js`
        - Work Experience: `components/WorkExperience.js`
        - Education: `components/Education.js`
        - Skills: `components/Skills.js`
        - Summary: `components/Summary.js`
        - Awards: `components/Awards.js`
        - Volunteering: `components/Volunteering.js`
    - Combine these components in:
        - Editor Layout: `components/EditorLayout.js`

3. **Create the Preview Layout:**

    - Create components for displaying each section.
    - Combine these components in:
        - Preview Layout: `components/PreviewLayout.js`

4. **Implement State Management with TanStack Store:**

    - Define the state and actions for managing resume data in:
        - Store Entry Point: `store/index.js`
        - Resume Store: `store/resumeStore.js`

5. **Save and Restore Functionality:**

    - Implement utility functions for saving to and loading from local storage:
        - Local Storage Utilities: `utils/localStorage.js`
    - Create a Save button component:
        - Save Button: `components/SaveButton.js`
    - Integrate save and restore functionality in the main App component:
        - Main App Component: `pages/App.js`

6. **Implement Drag and Drop Functionality:**

    - Enable drag-and-drop reordering for sections:
        - Work Experience: `components/WorkExperience.js`

        - Education: `components/Education.js`
        - Skills: `components/Skills.js`
        - Awards: `components/Awards.js`
        - Volunteering: `components/Volunteering.js`

7. **Export and Import Functionality:**

    - Implement utility functions for exporting to PDF and JSON:
        - PDF Export Utilities: `utils/pdfExport.js`
    - Add export and import buttons in the main App component:
        - Main App Component: `pages/App.js`

8. **Error Handling:**

    - Create an error boundary component:
        - Error Boundary: `components/ErrorBoundary.js`
    - Wrap the main components with the error boundary:
        - Main App Component: `pages/App.js`

**Finalized File Structure with Implementation Plan:**

**Components:**

- `components/BasicInformation.js`: Input fields for user's full name, contact information, and professional title.
- `components/WorkExperience.js`: Input fields for company name, job title, start and end dates, job description, and current job indicator. Includes drag-and-drop reordering.
- `components/Education.js`: Input fields for institution name, degree, field of study, start and end dates, and current study indicator. Includes drag-and-drop reordering.
- `components/Skills.js`: Input fields for skill name and proficiency level. Includes drag-and-drop reordering.
- `components/Summary.js`: Text area for summary or career objective.
- `components/Awards.js`: Input fields for award title, awarder, date, and description. Includes drag-and-drop reordering.
- `components/Volunteering.js`: Input fields for organization name, role, start and end dates, and description. Includes drag-and-drop reordering.
- `components/TemplateSelection.js`: Options for selecting resume templates.
- `components/EditorLayout.js`: Combines all the editor sections and provides the layout for the editor.
- `components/PreviewLayout.js`: Displays the preview of the resume based on the data from the editor.
- `components/SaveButton.js`: Button to save the resume data to local storage.
- `components/ErrorBoundary.js`: Component to catch and display errors gracefully.

**Store:**

- `store/index.js`: Entry point for TanStack Store.
- `store/resumeStore.js`: Defines the state and actions for managing resume data.

**Pages:**

- `pages/index.js`: Entry point for the React app.
- `pages/App.js`: Main App component that includes routing and layout structure.
- `pages/EditorPage.js`: Main page for the resume editor, includes both the editor and preview layouts.

**Utils:**

- `utils/localStorage.js`: Utility functions for saving to and loading from local storage.
- `utils/pdfExport.js`: Utility functions for exporting resume to PDF.

**Styles:**

- `styles/index.css`: Main CSS file for global styles.