**AI Document Assistant**

**Overview**

The AI Document Assistant is a React-based web application designed to help users improve their documents by providing suggestions for grammar, clarity, spelling, and conciseness. Users can upload documents in various formats (TXT, DOCX, PDF) and receive dynamic feedback and suggestions for improvements based on the content of their documents.

**Features**

* **Document Upload**: Users can upload text files, Word documents, and PDF files for analysis.
* **Text Improvement**: The application analyzes the document and generates suggestions for improving grammar, clarity, spelling, and wordiness.
* **Interactive Suggestions**: Users can accept or reject suggestions, applying changes directly to the improved document.
* **Editable Document Fields**: Users have the option to manually edit the documents after receiving suggestions.
* **User Notifications**: Alerts are displayed to inform users of successful improvements or remaining suggestions.

**Technologies Used**

* **React**: The main framework used for building the user interface.
* **JSX**: Syntax used for creating components and structuring the web application's interface in a way that looks like HTML but behaves like JavaScript.
* **CSS**: Styling of the application for a responsive and user-friendly interface.
* **JavaScript (React)**: Handles functionality for processing documents, generating suggestions, and user interactions.
* **PDF.js**: A JavaScript library to display PDF documents within the web browser.
* **Mammoth.js**: A library to extract plain text from DOCX files.

**Documentation**

**1. Instructions for Setting Up and Running the Frontend Application**

**Prerequisites**

* Ensure that you have **Node.js** and **npm** installed on your machine. You can download them from [Node.js Official Website](https://nodejs.org/).

**Installation Steps**

1. **Clone the Repository**:

Copy code

git clone https://github.com/newtonMM/Document-assistant.git

cd Document-assistant

1. **Install Dependencies**:

Copy code

npm install

1. **Run the Application**:

Copy code

npm start

This command will start the development server, and the app will be accessible at http://localhost:3000 in your web browser.

**Deployment Instructions**

* You can deploy the React application to GitHub Pages, Netlify, Vercel, or any other hosting service that supports static web apps.
* Follow the specific deployment steps for your chosen platform to ensure that the site is accessible online.

**2. User Guide on How to Use the Document Assistant Interface**

**How to Use the Application**

1. **Upload a Document**:
   * Click on the **"Choose File"** button to upload a document in TXT, DOCX, or PDF format.
   * The selected file's name will appear below the button once it is chosen.
2. **Improve the Document**:
   * Click on the **"Improve Document"** button to analyze the uploaded document and generate improvement suggestions.
   * The system will display a loader while the document is being processed.
3. **Interact with Suggestions**:
   * The suggestions will appear in the sidebar, categorized according to the types of errors found.
   * You can **accept** or **reject** each suggestion:
     + **Accept**: The suggestion will be applied directly to the improved document, and the suggestion will be removed from the list.
     + **Reject**: The suggestion will be removed from the list, and the user can decide whether to keep the document as is or make manual changes.
4. **Manual Editing**:
   * If you choose to make manual changes, the text fields in the document viewer will become editable, allowing you to directly edit both the original and improved documents.
5. **Notifications and Alerts**:
   * Notifications will appear to inform you when all suggestions have been handled or when an improvement process has been successfully completed.

**Example Document**

You can test the application with the following example text that contains various errors:

In this report, it has been observed that there are numerous issues that need to be fixed. First, the usage of passive voice is common in the writing, and it should be avoided where possible. For example, “The report was completed by the team” could be more direct if rewritten to “The team completed the report.”

Furthermore, there is also a significant amount of repetition present in the document. Phrases like “in order to” are used repeatedly and could be simplified. This document aims to ensure that the quality of writing is improved by reducing such repetitive phrases that make the text feel wordy and unengaging.

Additionally, spelling mistakes are evident throughout the report. Words like "adress" and "ocurr" are common errors that need to be corrected to enhance readability.

The conclusions drawn from this analysis indicate that there are a lot of areas that could use improvement, although it may seem that the document is concise. However, the document is not concise at all; instead, it is quite lengthy, and certain sections could benefit from summarizing the content to improve clarity and flow.

Overall, it is essential to review the entire document for grammatical errors, clarity issues, and to check for any spelling mistakes. The aim is to enhance the overall quality of the writing while also ensuring that the message is conveyed effectively.

**Contributing**

Contributions are welcome! If you have suggestions for improvements or additional features, feel free to fork the repository and submit a pull request.

**License**

This project is licensed under the MIT License - see the LICENSE file for details.

**Acknowledgments**

* **PDF.js** and **Mammoth.js** for providing the tools to handle document parsing and display.
* The React community for creating a robust framework for building web applications.