AI Based Granting Tool

Approach:

1. **Problem Understanding:**
    - Identify the need to enhance grant writing using AI by providing suggestions for improvement.
    - Ensure a seamless user experience where only AI-generated suggestions are displayed on the frontend.
2. **Backend Development:**
    - Implemented an Express.js server to handle API requests from the frontend.
    - Integrated the Hugging Face GPT-2 model API for generating grant-writing suggestions.
3. **Frontend Development:**
    - Designed a React-based user interface for entering grant text and displaying AI suggestions.
    - Established a clear flow where users input grant text, trigger the API request, and view AI-generated suggestions.
4. **Testing and Optimization:**
    - Tested the backend and frontend integration to confirm the correct data flow.
    - Optimized API calls with retry logic for scenarios where the Hugging Face model might be loading.

AI Tools Used:

Tools and Frameworks

1. **Hugging Face GPT-2 Model API:**
    - Applied for generating grant-writing suggestions based on user input.
2. **Express.js:**
    - Used to create the backend server, manage API endpoints, and handle data flow between the frontend and Hugging Face API.
3. **React.js:**
    - Built the frontend user interface for the application.

    - Managed user inputs and displayed AI-generated suggestions using state variables.
4. **Axios:**
    - Used for making HTTP requests to the Hugging Face API from the backend and communicating between the backend and frontend.
5. **Cors:**
    - Enabled cross-origin requests between the frontend and backend during development.

Challenges and Learnings:

Challenges Faced

1. **API Response Parsing:**
    - Initial responses included both input and AI-generated text.
    - Resolved by extracting only the Generated_Text field from the Hugging Face API response.
2. **Frontend Display Issue:**
    - Input text appeared in the AI suggestions section.
    - Updated the backend response structure and adjusted the frontend logic to ensure only AI suggestions were displayed.
3. **Timeout Errors:**
    - Faced occasional timeout errors due to the Hugging Face model loading.
    - Implemented retry logic with exponential backoff to handle such scenarios.

Learnings

- Gained a deeper understanding of integrating AI models into web applications.
- Learned the importance of clear data flow and parsing between frontend and backend.
- Improved skills in error handling and optimizing user experience for AI-powered tools.