

# Product Manager Recruitment Task

## Problem Statement:

Build a web application that allows users to upload documents (PDF, Word, etc.), perform natural language processing (NLP) on the documents using Langchain, and allow users to query the content of those documents through a simple search interface.

The application will have two main components:

1. **Front-End (MERN Stack)**
  - **React** will be used to create a user interface that allows users to:
    - Upload documents.
    - View uploaded documents and the results of a search.
    - Perform search queries on the document content.
2. **Back-End (Node.js, Express, and FastAPI)**
  - **Node.js with Express** will be used for the basic routing of the application and to handle user requests (e.g., file uploads and search queries).
  - **FastAPI** will be used to integrate Langchain for processing the documents and answering queries about them.
3. **Langchain Integration**
  - **Langchain** should be used to extract information from the uploaded documents, enabling users to search the content efficiently and retrieve specific answers based on natural language queries.

## Requirements:

1. **Front-End (React)**
  - A user-friendly interface for uploading documents (e.g., drag and drop or file selection).
  - Display the list of uploaded documents and a preview (name, file type).
  - A search bar that allows users to query the content of the documents.
  - Display search results with relevant information from the documents based on user input.
2. **Back-End (Node.js/Express)**
  - API endpoints for:
    - **File upload:** Endpoint to handle document uploads.
    - **Document Search:** Endpoint that queries the FastAPI service for relevant results from the documents.
3. **Back-End (FastAPI + Langchain)**
  - An API built using FastAPI that integrates Langchain to:
    - Parse and process the uploaded documents (convert documents to a readable format, e.g., text).
    - Use Langchain's document indexing and search features to retrieve answers based on search queries from the user.
4. **Document Handling:**
  - Support multiple document types (PDF, Word).
  - Store the uploaded documents and their processed versions (e.g., text extracted from PDFs).

## Deployment Requirements:

1. **Cloud Deployment (Render or Vercel):**
    - Deploy the full application (both the MERN stack front-end and the FastAPI back-end) to **Render** or **Vercel**.
      - **Render:** Deploy the back-end using **Render's web service**
      - **Vercel:** Deploy the front-end on Vercel
      - Ensure the deployment works smoothly in a production environment, with all endpoints functioning and no issues.
  2. **Environment Variables & Secrets:**
    - Store sensitive information (such as API keys, database credentials, etc.) in environment variables.
    - Ensure the deployment platform has appropriate configuration for production (e.g., setting up MongoDB Atlas connection, FastAPI API URL, etc.).
- 

## Deliverables:

- A fully functional MERN stack application that integrates with FastAPI and Langchain.
  - A working cloud deployment on **Render** or **Vercel** (accessible via a public URL).
  - Well-documented code
- 

## Evaluation Criteria:

- **MERN Stack Proficiency:** How well the front-end and back-end interact with each other and how the MERN stack is implemented.
  - **FastAPI & Langchain Integration:** The efficiency and correctness of the document processing and search functionality using Langchain.
  - **Cloud Deployment:** Successful deployment of both the front-end and back-end applications to **Render** or **Vercel**, with proper environment setup.
  - **Code Quality:** Clean, well-structured, and maintainable code. Proper use of asynchronous programming, error handling, and API best practices.
  - **Problem Solving:** Ability to integrate the various technologies effectively and handle edge cases like file formats, errors in processing, etc.
  - **UI/UX Design:** Usability and design of the front-end interface.
- 

## Submission Instructions:

- Please submit a GitHub repository link with the source code.
- Provide a link to the live deployed application with credentials or necessary access details (if applicable).