

# Fullstack Internship Assignment

## Objective:

Develop a full-stack application that allows users to upload PDF documents and ask questions regarding the content of these documents. The backend will process these documents and utilize natural language processing to provide answers to the questions posed by the users.

## Tools and Technologies:

**Backend:** FastAPI

**NLP Processing:** LangChain/LLamaIndex

**Frontend:** React.js

**Database:** SQLite or PostgreSQL (for storing document metadata, if necessary)

**File Storage:** Local filesystem or cloud storage (e.g., AWS S3) for storing uploaded PDFs

## Functional Requirements:

### PDF Upload:

- Users can upload PDF documents to the application.
- The application stores the PDF and possibly extracts and stores its text content for further processing.

### Asking Questions:

- Users can ask questions related to the content of an uploaded PDF.
- The system processes the question and the content of the PDF to provide an answer.

### Displaying Answers:

- The application displays the answer to the user's question.
- Include the functionality to ask follow-up or new questions on the same document.

## Non-Functional Requirements:

**Usability:** Ensure the user interface is intuitive and easy to navigate.

**Performance:** Optimize the processing of PDF documents and the response time for answering questions.

## Backend Specification:

**FastAPI Endpoints:**

- An endpoint for uploading PDF documents.
- An endpoint for receiving questions and returning answers based on the uploaded PDFs.

#### **PDF Processing:**

- Extract text from uploaded PDFs using a suitable library (e.g., PyMuPDF).
- Use the LangChain/llama index to process natural language questions and generate answers based on the PDF content.

#### **Data Management:**

- Store information about uploaded documents (e.g., filename, upload date) in a database.

## Frontend Specification:

#### **User Interface:**

- A page for uploading PDF documents.
- A feature for asking questions on uploaded documents and viewing answers.

#### **Interactivity:**

- Implement feedback mechanisms while uploading documents and processing questions.
- Display error messages for unsupported file types or processing errors.

#### **Design**

- We've created the design to give you an idea regarding how the application should look which you can find here:

<https://www.figma.com/file/QHpASp7wGRRcjh0oxCuspL/FullStack-Engineer-Internship-Assignment?type=design&node-id=0-1&mode=design&t=geu9rfpXEecN8eFZ-0>

## Assignment Deliverables:

**Source Code:** Provide the source code for both the frontend and backend, structured and commented appropriately.

**Documentation:** Include a README file with setup instructions, API documentation, and a brief overview of the application architecture.

**Demo:** A live demo or screencast showcasing the application's functionality.

## Evaluation Criteria:

**Functionality:** The application meets all the functional and non-functional requirements.

**Code Quality:** The code is clean, well-organized, and commented.

**Design and UX:** The application is user-friendly and easy to use.

**Innovation:** Any additional features or improvements that enhance the application's usability or performance.