

Project Title: Building an AI-Powered PDF Knowledge Assistant with LangChain and Llamaindex

Objective: Students will build a full-stack application that leverages LangChain and Llamaindex to create a chatbot capable of answering questions from a custom dataset. The chatbot will index and query PDF documents, making use of vector databases and language models to generate accurate responses.

Learning Outcomes:

- Understand how to integrate LangChain for prompt orchestration
- Learn how to use Llamaindex for PDF ingestion and indexing
- Understand the basics of vector databases through user-friendly APIs (FAISS or Chroma)
- Use OpenAI or other LLM APIs to generate intelligent responses
- Develop a simple frontend to interact with the backend AI agent

Project Requirements:

- Node.js with Express for backend
- LangChain
- Llamaindex
- FAISS or Chroma as vector DB (minimal manual configuration required)
- OpenAI API key or similar LLM
- Basic React (or Vite + React) frontend

Features to Implement:

1. PDF Upload:
 - Users can upload PDF documents to the backend
2. Indexing:
 - Llamaindex ingests and creates a vector index from the uploaded PDFs
3. Query Interface:
 - A chat interface where users type questions about the PDF content
4. Response Generation:
 - LangChain uses prompts + the index to return intelligent answers
5. Memory (Optional Bonus):
 - Add conversational memory using LangChain memory components

Step-by-Step Guide:

1. Set Up the Backend: Initialize Node.js project and install dependencies: `npm init -y`
 - `npm install express multer openai dotenv cors`
 - Set up an Express server with endpoints for uploading PDFs and querying
2. Ingest PDFs:
 - Use Llamaindex and a wrapper to parse PDF content
 - Create a vector index (FAISS or Chroma) using simple API calls
3. Build the LangChain Chain:
 - Use a RetrievalQA chain that takes user questions and finds context from the indexed PDF content

4. Frontend (React):

- Build a basic chat UI
- Connect to the backend via REST API

5. Test & Deploy:

- Allow users to test by uploading various PDFs
- Deploy using Render, Railway, or Vercel

