2-Week Intern Project Plan: Document-Based Chatbot using Angular and Custom Backend

Project Title:

Document-Based Chatbot using Angular and Custom Backend (No External APIs)

Goal:

Create a chatbot system that:

- Allows users to upload a document (PDF/TXT)

- Parses and stores the document content in the backend

- Enables users to ask questions

- Responds based on the document's content using simple text matching/search

Duration:

2 Weeks (Ideal for Interns)

Tech Stack:

Frontend: Angular (v15+), HTML/CSS

Backend: Node.js (Express) or Spring Boot

File Parsing: pdf-parse (Node) or Apache Tika (Java)

Communication: REST API

Suggested Folder Structure:

```
document-chatbot/
 frontend/ (Angular)
    src/app/
       components/
          upload/
          chat/
          message/
```

```
backend/ (Node.js or Spring Boot)
   uploads/
   services/
   controllers/
README.md
```

WEEK 1: Backend & File Handling

Day 1 Project Setup & Planning

- Discuss chatbot flow: upload parse chat search respond
- Set up Angular and backend project scaffolding

Deliverables:

- Project initialized
- High-level architecture documented

Day 2 Backend: File Upload API

- Create /upload endpoint to accept .txt or .pdf
- Store file on server

Deliverables:

- Working file upload API
- Uploaded files visible in /uploads folder

Day 3 Backend: Parse Document Content

- Read .txt or .pdf using:
  - fs.readFile for .txt
  - pdf-parse (Node.js) for .pdf
- Save plain text in memory or file

Deliverables:

- Raw document text extracted and logged

Day 4 Backend: Basic Search API

- Create /chat endpoint:
  - Accept user question
  - Match keywords or sentences using basic includes() or fuzzy match
  - Return best-matched sentence(s) from doc

Deliverables:

- Simple Q&A working with backend logic

Day 5 Angular: File Upload UI

- Create upload component in Angular
- Use HttpClient to post file to backend

Deliverables:

- UI to upload document and see status

Day 6 Angular: Chat UI Setup

- Create static UI:
  - Chat bubbles
  - Input field
- Simulate sending/receiving messages

Deliverables:

- Working static chat interface

Day 7 Weekly Testing & Review

- Upload PDF & ask questions via Postman

- Clean up folder structures

- Document Week 1 progress in README.md

Deliverables:

- Backend and frontend partially integrated

- End-to-end file upload and message flow tested

WEEK 2: Frontend Integration + Smart Search + Polish

Day 8 Angular: Connect Chat UI to Backend

- Send user input from UI to backend /chat

- Show response in chatbot bubble

Deliverables:

- Frontend & backend connected for Q&A

Day 9 Improve Chat Flow

- Auto-scroll chat window

- Clear input after send

- Show loader ("Bot is typing...")

Deliverables:

- Chatbot with clean UX

Day 10 Improve Search Logic (Optional)

- Use:
  - Sentence splitting
  - Basic TF-IDF match (optional)

  - Word vector similarity (advanced)

- Fallback response: I couldnt find that in the document.

Deliverables:

- More relevant and robust responses

Day 11 Error Handling & Edge Cases

- Handle:
  - Unsupported files
  - Empty input
  - No matches

Deliverables:

- Stable chatbot with basic validation

Day 12 Testing with Different Docs

- Test various types of content
- Try long PDFs, short notes, FAQs

Deliverables:

- Bug report and fixes

Day 13 Polish & Styling

- Add colors, user profile icon
- Responsive layout
- Improve chat layout

Deliverables:

- Final styled chatbot

Day 14 Final Demo & README

- Record demo (optional)

- Write full README.md:

  - Setup steps

  - How to use chatbot

  - Known limitations

Deliverables:

- Project ready to showcase

Final Deliverables:

Angular chatbot with:

- File upload

- Real-time Q&A

Backend API with:

- Upload, parse, and match logic

Fully tested, working chatbot app

README.md with instructions and screenshots