LABORATORY REPORT

Application Development Lab
(CS33002)

B.Tech Program in ECSc

Submitted By

Name:-BISHWAJEET DAS

Roll No: 2230078

Kalinga Institute of Industrial Technology
(Deemed to be University)
Bhubaneswar, India

Spring 2024-2025

Table of Content

Experiment Number	4
Experiment Title	Conversational Chatbot with PDF Reader
Date of Experiment	03/02/2025
Date of Submission	09/02/2025

1. Objective:- To build a chatbot capable of answering queries from an uploaded PDF/Word/Excel document.

2. Procedure:-

Procedure:

1. Integrate open-source LLMs such as LLama or Gemma from Ollama
2. Develop a Flask backend to process the PDF/word/excel content. 3. Implement Natural Language Processing (NLP) to allow queries. You can use
LLamaIndex or Langchain
4. Create a frontend to upload document files and interact with the chatbot, just like
OpenAI interface
5. Provide an option to choose the LLM model from a dropdown list. 6. Display the chatbot responses on the webpage.

3. Code:-

App.py -

```
from flask import Flask, request, jsonify, render_template
import requests
import os
import fitz  # PyMuPDF for extracting text from PDFs

app = Flask(__name__)

# Set your Groq API key as an environment variable
GROQ_API_KEY = os.getenv('GROQ_API_KEY')
GROQ_API_URL = "https://api.groq.com/openai/v1/chat/completions"
MODEL_ID = "llama-3.3-70b-versatile"

uploaded_file_content = ""  # Store extracted text from uploaded PDF

@app.route("/")
def home():
    """Serve the frontend HTML page."""
    return render_template("index.html")

def query_groq_llama(prompt, file_content=None):
    """Send a request to the Groq LLaMA API."""
    messages = [{"role": "user", "content": prompt}]
    if file_content:
        messages.append({"role": "system", "content": file_content})

    payload = {
```

```
        "model": MODEL_ID,
        "messages": messages,
        "max_tokens": 500,
    }
    headers = {
        "Authorization": f"Bearer {GROQ_API_KEY}",
        "Content-Type": "application/json"
    }

    response = requests.post(GROQ_API_URL, json=payload, headers=headers)
    return response.json()

@app.route("/chat", methods=["POST"])
def chat():
    """Handle chat requests."""
    global uploaded_file_content
    data = request.json
    prompt = data.get("message", "")

    if not prompt:
        return jsonify({"error": "Message is required"}), 400

    response = query_groq_llama(prompt, uploaded_file_content)
    return jsonify(response)

@app.route("/upload", methods=["POST"])
def upload():
    """Handle PDF file uploads and extract text."""
    global uploaded_file_content

    if "file" not in request.files:
        return jsonify({"error": "No file provided"}), 400

    file = request.files["file"]

    if not file.filename.endswith(".pdf"):
        return jsonify({"error": "Only PDF files are allowed"}), 400

    # Extract text from PDF
    doc = fitz.open(stream=file.read(), filetype="pdf")
    text = "\n".join([page.get_text() for page in doc])

    uploaded_file_content = text  # Store extracted text
    return jsonify({"message": "PDF uploaded successfully!", "extracted_text": text[:500] + "..."})  # Show preview

if __name__ == "__main__":
    app.run(debug=True)
```

Index.html -

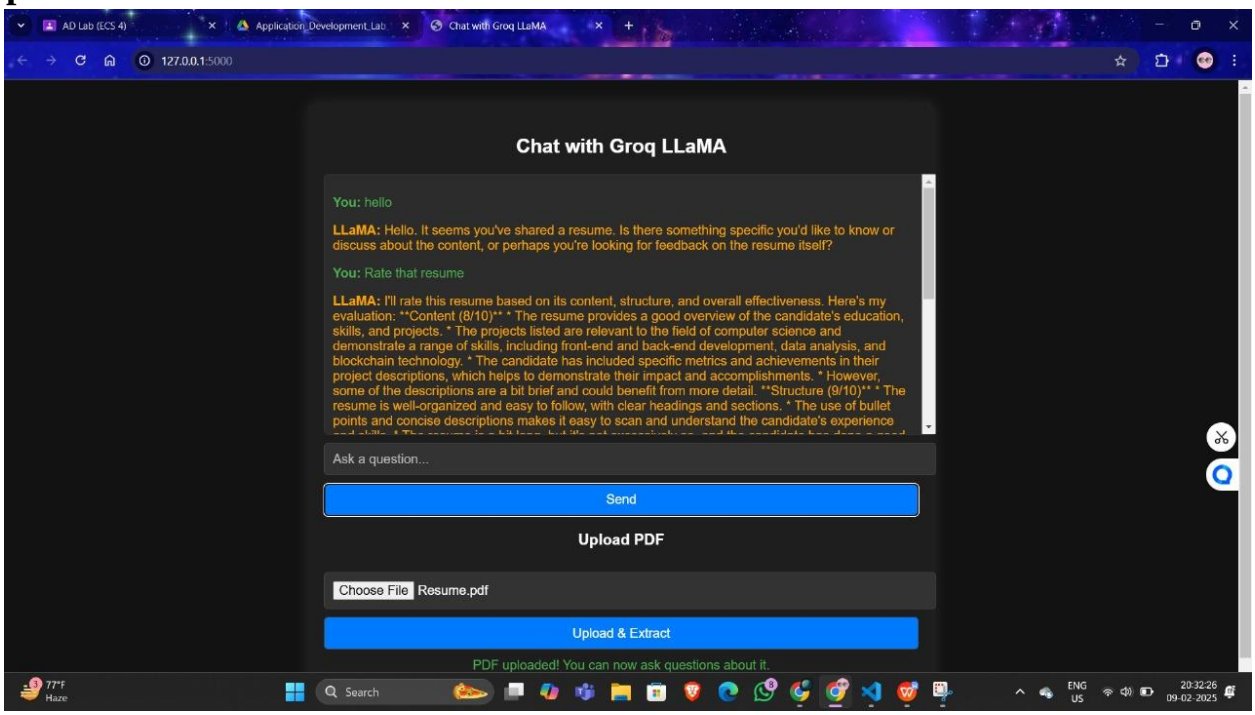
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chat with Groq LLaMA</title>
    <style>
        /* Dark mode styles */
        body {
```

```
            font-family: Arial, sans-serif;
            background-color: #121212;
            color: #ffffff;
            text-align: center;
            padding: 20px;
        }
        .container {
            width: 50%;
            margin: auto;
            background: #1e1e1e;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0px 0px 15px rgba(255, 255, 255, 0.1);
        }
        #chatbox {
            width: 100%;
            height: 300px;
            overflow-y: scroll;
            border: 1px solid #333;
            padding: 10px;
            background: #2c2c2c;
            text-align: left;
            border-radius: 5px;
        }
        .user-message {
            color: #4CAF50;
        }
        .llama-message {
            color: #FFA500;
        }
        input, button {
            width: 100%;
            margin-top: 10px;
            padding: 10px;
            font-size: 16px;
            border-radius: 5px;
            border: 1px solid #444;
            background: #333;
            color: white;
        }
        input::placeholder {
            color: #bbb;
        }
        button {
            background-color: #007bff;
            cursor: pointer;
        }
        button:hover {
            background-color: #0056b3;
        }
        #uploadStatus {
            margin-top: 10px;
            color: #4CAF50;
        }
    </style>
</head>
<body>

    <div class="container">
```

```
    <h2>Chat with Groq LLaMA</h2>
    <div id="chatbox"></div>

    <input type="text" id="message" placeholder="Ask a question..." onkeypress="handleKeyPress(event)">
    <button onclick="sendMessage()">Send</button>

    <h3>Upload PDF</h3>
    <input type="file" id="fileInput" accept=".pdf">
    <button onclick="uploadFile()">Upload & Extract</button>

    <p id="uploadStatus"></p>
  </div>

  <script>
    // Handle Enter key press
    function handleKeyPress(event) {
      if (event.key === "Enter") {
        sendMessage();
      }
    }

    async function sendMessage() {
      let message = document.getElementById("message").value.trim();
      if (!message) return;

      let chatbox = document.getElementById("chatbox");
      chatbox.innerHTML += `<p class="user-message"><b>You:</b> ${message}</p>`;
      chatbox.scrollTop = chatbox.scrollHeight;  // Auto-scroll

      let response = await fetch("/chat", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ message })
      });

      let result = await response.json();
      if (result.choices && result.choices[0] && result.choices[0].message) {
        chatbox.innerHTML += `<p class="llama-message"><b>LLaMA:</b>
${result.choices[0].message.content}</p>`;
      } else {
        chatbox.innerHTML += `<p class="llama-message"><b>LLaMA:</b> No response received.</p>`;
      }

      document.getElementById("message").value = "";
      chatbox.scrollTop = chatbox.scrollHeight;  // Auto-scroll
    }

    async function uploadFile() {
      let file = document.getElementById("fileInput").files[0];
      if (!file) {
        alert("Please select a PDF file to upload.");
        return;
      }

      let formData = new FormData();
      formData.append("file", file);

      let response = await fetch("/upload", { method: "POST", body: formData });
      let result = await response.json();
```

```
        if (result.error) {
            document.getElementById("uploadStatus").innerText = `Error: ${result.error}`;
        } else {
            document.getElementById("uploadStatus").innerText = "PDF uploaded! You can now ask questions about it.";
        }
    }
</script>

</body>
</html>
```

4. Output -



5. **Remarks:-**

Signature of the Lab Coordinator

Prof. Bhragav Appasani

-