

# Portfolio Web Application with Integrated RAG Chatbot

## Final Project Presentation

By - 1. Kaushikee Bhawsar (NUID - 002704590)  
2. Anmol Valecha (NUID - 002813410)

# Meet the Team



**Kaushikee Bhawsar**  
**(NUID - 002704590)**



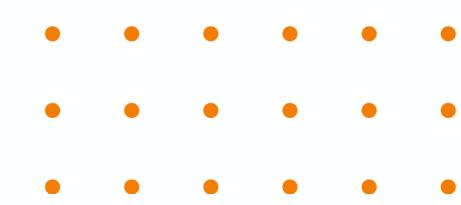
**Anmol Valecha**  
**(NUID - 002813410)**

# Overview

- 1. Introduction**
- 2. Project Description**
- 3. Project Architecture**
- 4. Data Collection & Preprocessing**
- 5. RAG Pipeline Implementation**
- 6. Performance Metrics**
- 7. Improvement Strategies - FineTuning**
- 8. Deployment Plan**
- 9. Future Work**
- 10. Conclusion**



# Introduction



## Overview:

- Development of a personal portfolio website.
- Integration of a personal assistant chatbot using advanced AI techniques.

## Objectives:

- Showcase personal skills and projects.
- Provide an interactive and engaging experience for visitors.
- Utilize cutting-edge AI for personalized user interactions.

## Relevance:

- Demonstrates proficiency in frontend and backend development.
- Applies course concepts in a practical, real-world project.



# Project Description

## Project Description

- A modern, responsive portfolio website.
- Sections include: About, Projects, Skills, Education, Work Exp., Certifications, Contact.
- Integrated personal assistant chatbot for enhanced user interaction.

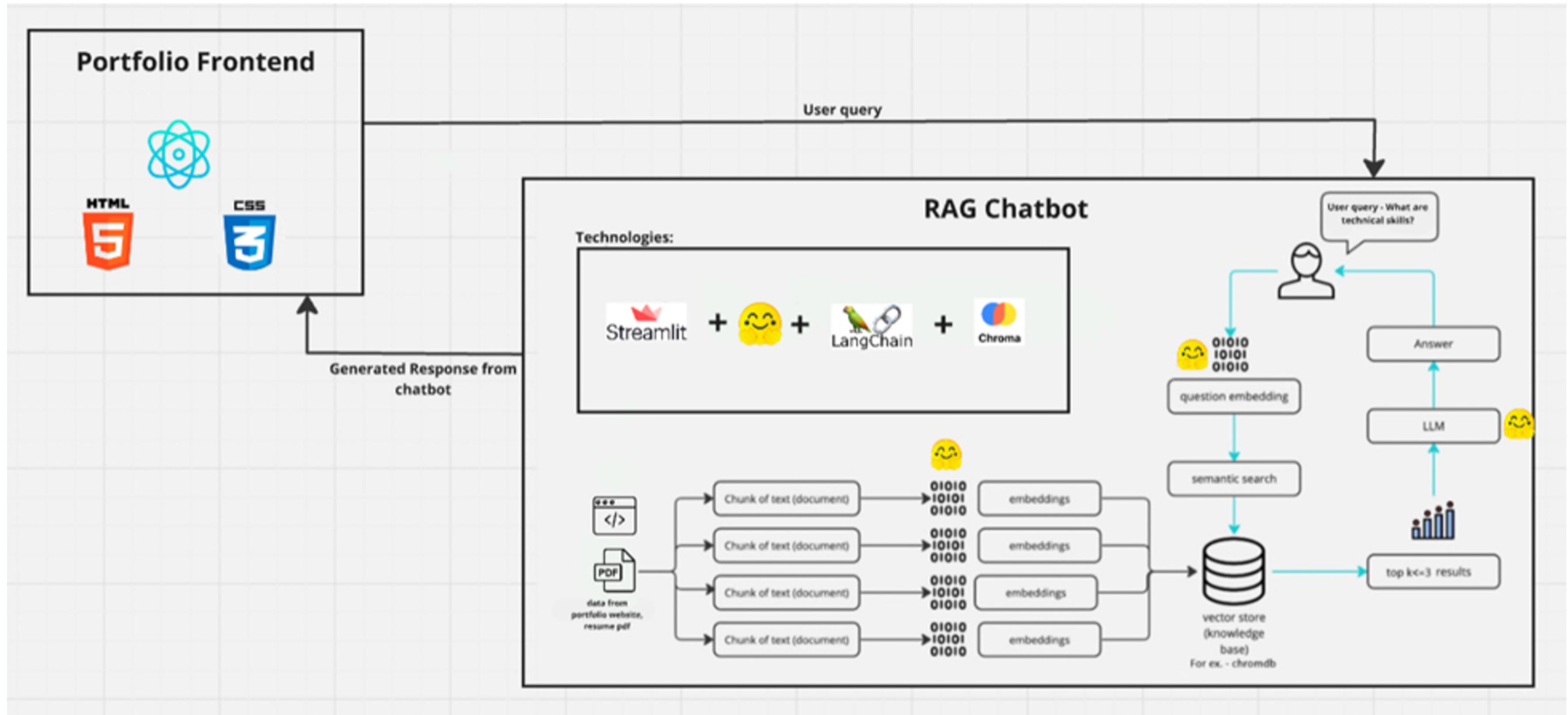
## Problem Solved

- Creates a professional online presence.
- Offers an interactive way to learn about the individual's skills and projects.
- Provides instant, AI-driven responses to visitor queries

## Scope

- Complete design and development of the portfolio website.
- Development and integration of the LLM based RAG chatbot.

# Project Architecture



# Tools & Technologies



## Frontend Component

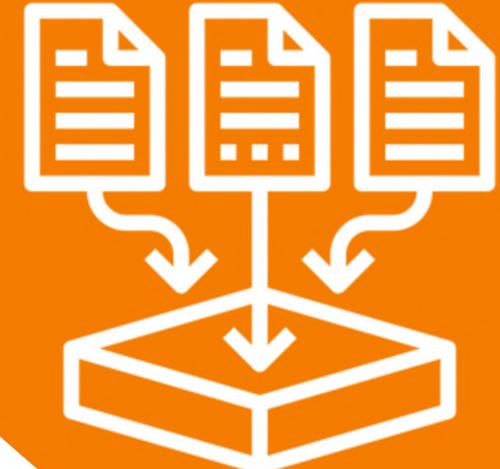
- React
- HTML
- CSS



## Chatbot Component

- RAG pipeline - Langchain
- LLM - Hugging face transformer
- Vector DB - Chroma
- Frontend - Streamlit
- Programming language - Python

# Data Collection & Preprocessing



## Step 2: Data Collection Techniques:

- **Web scraping**
  - Identify the sections of our portfolio website to scrape.
  - BeautifulSoup to extract text content.
- **Resume text extraction**
  - Using PyPDF to extract text from pdf

## Step 1: Data Sources:



01



02

Portfolio  
Website

Resume

## Step 3: Data preprocessing:

- **Data Cleaning:**
  - Remove HTML tags, special characters, and irrelevant text.
- **Vectorization:**
  - Convert text data into numerical format (embeddings).
  - Use tools like Hugging Face Transformers for embeddings.
- **Indexing:**
  - Store processed data in a vector database (Pinecone/Chroma).



**Code snipped - To scrap data from portfolio webpage**

```
import requests
import pandas as pd
from bs4 import BeautifulSoup

url = 'https://kaushikeebhawsar.netlify.app/#/home'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

divs = soup.find_all('div', class_='lb-grid')

Tabnine: Test | Explain | Document | Ask
def get_text_including_a_and_span_tags(p_tag):
    # Collect text pieces
    text_pieces = []
    for element in p_tag.descendants:
        if element.name in ['a', 'span']:
            text_pieces.append(element.get_text())
        elif isinstance(element, str):
            text_pieces.append(element)
    return ''.join(text_pieces).strip()

current_question = None
current_answer = []
data = []
for div in divs:
    div_children = div.find_all('div', class_='lb-txt-16 lb-rtxt')
    for div1 in div_children:
        ps = div1.find_all('p')
        for p in ps:
            b = p.find('b')
            if b is not None and b.text[0] == 'Q':
                if current_question:
                    data.append({'question': current_question, 'answer': ''.join(current_answer)})
                    current_answer = []
                current_question = b.text
            else:
                current_answer.append(get_text_including_a_and_span_tags(p))

            if current_question:
                data.append({'question': current_question, 'answer': ''.join(current_answer)})

df = pd.DataFrame(data)
df.to_csv('portfolio.csv', index=False)
```

# Preprocessed and clean Data

Kaushikee Bhawsar is a Software Engineer with 3+ years of experience in Software Development and Quality Assurance. Adept at leveraging cutting-edge tools and methodologies to deliver high-quality software solutions.

Kaushikee is proficient in programming languages such as C++, Java, Python, and SQL.

Kaushikee has expertise in ETL/Data Testing, Web & Application Testing, Mobile Testing, Performance Testing, API Testing, Usability Testing, Regression Testing, Black-Box Testing, and Unit Testing.

Kaushikee is skilled in using Selenium WebDriver, TestNG, and BDD Cucumber for automation testing.

Kaushikee has experience with AWS, Databricks, and Azure DevOps.

Kaushikee is proficient in HTML, CSS, JavaScript, Bootstrap, Angular17, and React.

At TD Bank, Kaushikee worked as a Software Quality Engineering Intern, where they leveraged BDD Cucumber, Java, and Jenkins using Groovy to create automated regression pipelines, achieving a 40%

Kaushikee orchestrated data pipelines with Azure Data Factory, ensured seamless data processes, and earned TD appreciation e-cards for consistent, noteworthy contributions.

Kaushikee spearheaded quality assurance efforts, developed over 1500 automated multi-browser tests with Selenium and TestNG, and reduced manual workload by 60% through root cause analysis.

Kaushikee received a top performance rating of 5/5 and was honored with the 'Standing Ovation Award' in the FY 20-21 Walk of Fame.

Kaushikee worked as a Software Developer Intern, where they designed and implemented responsive websites using React, Angular, HTML, CSS, and Bootstrap. They enhanced interactivity by integrating AI-powered features.

Kaushikee has a Master of Science in Information Systems from Northeastern University, Toronto, with a GPA of 3.94/4.00, and a Bachelor of Technology in Information Technology from Shri Vaishnav Institute of Technology.

Kaushikee completed her Master's degree in Information Systems at Northeastern University, Toronto.

Kaushikee achieved a GPA of 3.94/4.00 in her Master of Science in Information Systems.

Kaushikee holds certifications such as Oracle Cloud Infrastructure (OCI) 2024 Generative AI Professional Certification and NPTEL IIT Kharagpur Database Management System Certification.

Kaushikee holds the Oracle Cloud Infrastructure (OCI) 2024 Generative AI Professional Certification.

Kaushikee obtained the Database Management System Certification from NPTEL IIT Kharagpur.

Kaushikee has worked on multiple personal projects, including designing and implementing responsive websites, and integrating AI-based chatbots to enhance interactivity.

Kaushikee emphasizes open communication and active collaboration with cross-functional teams, including developers, business analysts, and product owners. She contributes to sprint planning sessions and provides technical guidance.

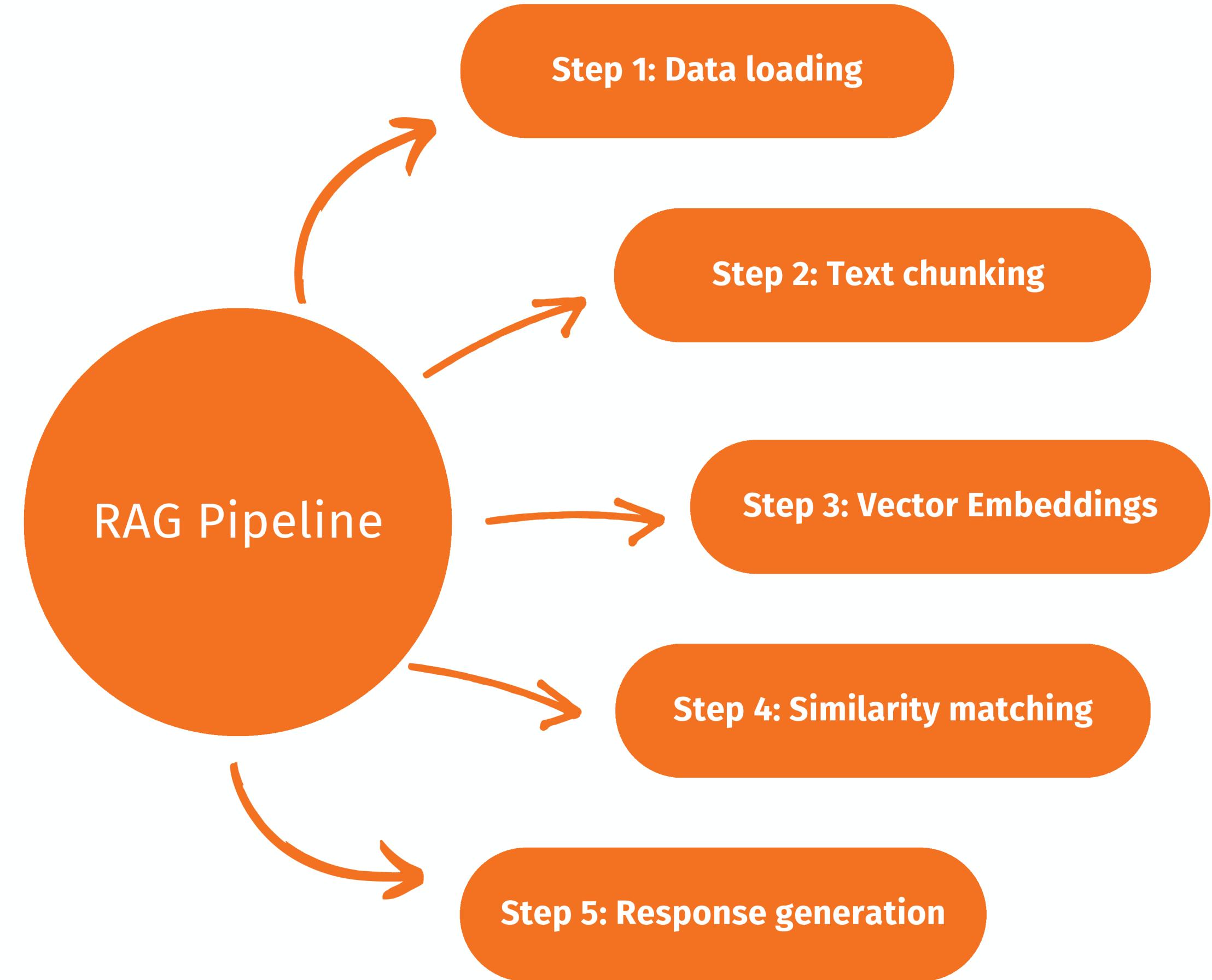
Kaushikee adopts a systematic approach to problem-solving, starting with root cause analysis to identify the core issues. She then collaborates with her team to develop and implement solutions that address these issues effectively.

Kaushikee's automation efforts, particularly at Wipro, led to a dramatic reduction in manual workload by 60%. Her development of over 1500 automated multi-browser tests with Selenium and TestNG contributed significantly to this efficiency gain.

Kaushikee is committed to continuous learning, regularly participating in online courses, attending industry webinars, and reading up on the latest technologies. Her certifications in Oracle Cloud Infrastructure and NPTEL programs reflect her dedication to staying updated.

Kaushikee ensures client satisfaction by delivering high-quality software solutions that meet or exceed client expectations. She is meticulous in her testing and validation processes, and she regularly communicates progress and findings to stakeholders.

# RAG Pipeline Implementation



# Install all the dependencies

01

Create virtual environment and activate it

```
python -m venv .venv
```

```
.venv\Scripts\activate.bat
```

02

Install all dependencies using pip command in the virtual environment

```
pip install -r requirements.txt
```

requirements.txt →

chromadb==0.5.3  
sqlite3-binary  
protobuf==3.20.\*  
streamlit==1.36.0  
pypdf==4.2.0  
langchain==0.2.5  
langchain-community==0.2.5  
langchain-huggingface==0.0.3  
python-dotenv

# Vector Database Implementation

# Vector Database Implementation

01

## Importing dependencies and API key

```
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import Chroma
import streamlit as st
import os, warnings
warnings.filterwarnings('ignore')

os.environ["HUGGINGFACEHUB_API_TOKEN"] = st.secrets["HUGGINGFACEHUB_API_TOKEN"]
```

# Sample Dataset

## Here is sample dataset

**KAUSHIKEE BHAWSAR**  
Toronto, ON | Email: [kaushikeebhawsar1999ca@gmail.com](mailto:kaushikeebhawsar1999ca@gmail.com) | Github: [github.com/kaushikeebhawsar99](https://github.com/kaushikeebhawsar99)  
LinkedIn: [linkedin.com/in/kaushikee-bhawsar-54127b177/](https://linkedin.com/in/kaushikee-bhawsar-54127b177/)

**SUMMARY & SKILLS**

Software Engineer with 3+ years of experience in Software Development and Quality Assurance. Adept at leveraging cutting-edge tools and methodologies to deliver high-quality, scalable, and innovative solutions in agile environments.

**Programming Languages:** C++, Java, Python, SQL  
**Automation Tools:** Selenium WebDriver, TestNG, BDD Cucumber  
**Quality Assurance:** ETL / Data Testing, Web & Application Testing, Mobile Testing, Performance Testing, API Testing, Usability Testing, Regression Testing, Black-Box Testing, Unit Testing  
**Cloud Technologies:** AWS, Databricks, Azure DevOps, Azure Data Factory  
**Database Management:** SQL Server management studio, Azure Cosmos DB, SQL Developer  
**CI/CD & DevOps:** Jenkins, Docker  
**Web Technologies:** HTML, CSS, JavaScript, Bootstrap, Angular17, React  
**Tools/Libraries:** JMeter, Postman, Gatling, Junit, Java Swing, Figma, JIRA/JTMF, NumPy, Pandas, Scikit-learn, Git/Github, Bitbucket, Power BI  
**Generative AI:** Prompt Engineering, LLMs, RAG

**PROFESSIONAL EXPERIENCE**

**TD Bank, Toronto, ON** May 2023 – Apr 2024  
*Software Quality Engineering Intern – Customer Risk Rating EDF*

- Leveraged BDD Cucumber, Java and Jenkins using Groovy to create automated regression pipelines, achieving a 40% reduction in testing time and a 30% boost in efficiency for Customer Risk Rating data.
- Executed ETL processes for data validation using SQL, Python/PySpark in Databricks. Achieved a high accuracy rate by meticulously processing millions of records and eliminating bad data.
- Orchestrated data pipelines with Azure Data Factory, ensuring seamless and efficient data extraction, transformation, and data loading processes, contributing to overall data quality and reliability.
- Earned TD appreciation e-cards from the Team lead and team members for consistent, noteworthy contributions and a results-oriented approach, focusing on achieving high-quality outcomes.

**Wipro Technologies, Pune, INDIA** Sep 2020 – Sep 2022  
*Quality Engineer – Black & Veatch, CBRE*

- Spearheaded the quality assurance efforts to successfully deliver flawless software products and developed over 1500 automated multi-browser tests with Selenium and TestNG.
- Facilitated root cause analysis of defects with the team and acted as a catalyst for continuous improvement, reducing manual workload by 60%.
- Streamlined IT workload by 20% via targeted performance testing using JMeter and API testing using Postman.
- Collaborated with BAs, Devs, POs in sprint planning sessions to decompose solutions into valuable increments and deliver assignments with detailed quality reports, ensuring client satisfaction by resolving critical defects.
- Secured a top performance rating of 5/5 and was honored with the "Standing Ovation Award" in the FY 20-21 Walk of Fame.

**OEPP Innovations Pvt. Ltd., Indore, INDIA** Jan 2020 – Apr 2020  
*Software Developer Intern – CropTrails, EcomTrails, CropRigate, HrTrails*

- Designed and implemented responsive websites using React, Angular, HTML, CSS and Bootstrap and enhanced interactivity by integrating AI-based Chatbots, boosting communication efficiency by 20%.
- Awarded "Intern of the Month" for outstanding performance and superior organizational skills.

**EDUCATION**

**Northeastern University, Toronto, ON** Sep 2022 – Aug 2024  
*Master of Science in Information Systems (GPA of 3.94/4.00)*

**Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, INDIA** Jul 2016 – May 2020  
*Bachelor of Technology in Information Technology (CGPA of 9.17/10.00)*

**CERTIFICATIONS**

- Oracle Cloud Infrastructure (OCI) 2024 Generative AI Professional Certification
- NPTEL IIT Kharagpur Database Management System Certification

Kaushikee Bhawsar is a Software Engineer with 3+ years of experience in Software Development and Quality Assurance. Adept at leveraging cutting-edge tools and methodologies to deliver high-quality, scalable, and innovative solutions in agile environments.

Kaushikee is proficient in programming languages such as C++, Java, Python, and SQL.

Kaushikee has expertise in ETL/Data Testing, Web & Application Testing, Mobile Testing, Performance Testing, API Testing, Usability Testing, Regression Testing, Black-Box Testing, Unit Testing, and more.

Kaushikee is skilled in using Selenium WebDriver, TestNG, and BDD Cucumber for automation testing.

Kaushikee has experience with AWS, Databricks, and Azure DevOps.

Kaushikee is proficient in HTML, CSS, JavaScript, Bootstrap, Angular17, and React.

At TD Bank, Kaushikee worked as a Software Quality Engineering Intern, where they leveraged BDD Cucumber, Java, and Jenkins using Groovy to create automated regression pipelines. Achieving a 40% reduction in testing time and a 30% boost in efficiency for Customer Risk Rating data.

Kaushikee orchestrated data pipelines with Azure Data Factory, ensured seamless data processes, and earned TD appreciation e-cards for consistently meeting performance targets.

Kaushikee spearheaded quality assurance efforts, developed over 1500 automated multi-browser tests with Selenium and TestNG, and reduced manual workload by 60%.

Kaushikee received a top performance rating of 5/5 and was honored with the 'Standing Ovation Award' in the FY 20-21 Walk of Fame.

Kaushikee worked as a Software Developer Intern, where they designed and implemented responsive websites using React, Angular, HTML, CSS, and Bootstrap.

Kaushikee has a Master of Science in Information Systems from Northeastern University, Toronto, with a GPA of 3.94/4.00, and a Bachelor of Technology in Information Technology from Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, with a CGPA of 9.17/10.00.

Kaushikee completed her Master's degree in Information Systems at Northeastern University, Toronto.

Kaushikee achieved a GPA of 3.94/4.00 in her Master of Science in Information Systems.

Kaushikee holds certifications such as Oracle Cloud Infrastructure (OCI) 2024 Generative AI Professional Certification and NPTEL IIT Kharagpur Database Management System Certification.

Kaushikee holds the Oracle Cloud Infrastructure (OCI) 2024 Generative AI Professional Certification.

Kaushikee obtained the Database Management System Certification from NPTEL IIT Kharagpur.

Kaushikee has worked on multiple personal projects, including designing and implementing responsive websites, and integrating AI-based chatbots.

Kaushikee emphasizes open communication and active collaboration with cross-functional teams, including developers, business analysts, and project managers.

Kaushikee adopts a systematic approach to problem-solving, starting with root cause analysis to identify the core issues. She then collaborates with stakeholders to develop effective solutions.

Kaushikee's automation efforts, particularly at Wipro, led to a dramatic reduction in manual workload by 60%. Her development of over 1500 automated tests contributed significantly to this efficiency gain.

Kaushikee is committed to continuous learning, regularly participating in online courses, attending industry webinars, and reading up on the latest research in her field.

Kaushikee ensures client satisfaction by delivering high-quality software solutions that meet or exceed client expectations. She is meticulous in her work, paying close attention to detail and quality.

# Vector Database Implementation

02

## PDF Data Loading

```
# Load the PDF
loader = PyPDFLoader("Kaushikee_Bhawsar_Resume.pdf")
documents = loader.load()
```

03

## Text Splitting

```
# Split the text
text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=100)
texts = text_splitter.split_documents(documents)
```

# Vector Database Implementation

04

## Text Embedding

```
# Initialize the embedding model
embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")

# Convert texts to embeddings
try:
    embeddings = embedding_model.embed_documents([doc.page_content for doc in texts])
    print("Embeddings created successfully")
except Exception as e:
    print(f"Error creating embeddings: {e}")
```

05

## Storing in Vector Database

```
# Initialize Chroma vector store
vector_store = Chroma(embedding_function=embedding_model, persist_directory="data")
# Add documents to the vector store
vector_store.add_documents(documents=texts)

# Save the embedding
vector_store.persist()
```

# Chatbot Development

# Chatbot Development

01

## Initialization of Environment and Dependencies

```
import streamlit as st
from langchain_huggingface import HuggingFaceEmbeddings
from langchain_community.vectorstores import Chroma
from langchain_community.llms import HuggingFaceHub
from langchain.prompts import PromptTemplate
from langchain.chains import RetrievalQA
import warnings, os
from dotenv import load_dotenv
warnings.filterwarnings("ignore")
__import__('pysqlite3')
import sys
sys.modules['sqlite3'] = sys.modules.pop('pysqlite3')

# Load environment variables from .env file
load_dotenv()

data_directory = os.path.join(os.path.dirname(__file__), "data")

os.environ["HUGGINGFACEHUB_API_TOKEN"] = os.getenv("HUGGINGFACEHUB_API_TOKEN")
```

# Chatbot Development

02

## Setup of the Language Model and Vector Store

```
21
22 # Load the vector store from disk
23 embedding_model = HuggingFaceEmbeddings(model_name="sentence-transformers/all-MiniLM-L6-v2")
24 vector_store = Chroma(embedding_function=embedding_model, persist_directory=data_directory)
25
```

03

## Preparation of the Retrieval System

```
25
26 # Initialize the Hugging Face Hub LLM
27 hf_hub_llm = HuggingFaceHub(
28     repo_id="meta-llama/Meta-Llama-3-8B-Instruct",
29     # repo_id="mistralai/Mistral-7B-Instruct-v0.3",
30     model_kwargs={"temperature": 1, "max_new_tokens":1024},
31 )
32
```

# Chatbot Development

04

## Configure the retriever to fetch top results

```
69     rag_chain = RetrievalQA.from_chain_type(  
70         llm=hf_hub_llm,  
71         chain_type="stuff",  
72         retriever=vector_store.as_retriever(top_k=3), # retriever is set to fetch top 3 results  
73         chain_type_kwargs={"prompt": custom_prompt})  
74
```

# Prompt Engineering

05

## Integration of Custom Prompt Template

```
prompt_template = """
You are Kaushikee's intelligent assistant with a vast knowledge base.
Your task is to read the given resume and answer the question asked based on the information provided in the resume.
While answering the question please adhere to the following guidelines:

1. Answer only the question asked: Use only the information provided in the resume. Do not add any extra information or make assumptions.
2. Greetings and other general queries: For non-resume-related questions like greetings or general inquiries, respond appropriately without referring to the resume.
3. Contact details: If asked for contact details, use the following: \n
   - Email: kaushikeebhawsar1999ca@gmail.com \n
4. Frame your answers in such a way that they showcase Kaushikee's importance.
5. No pre-amble and post-amble is required, just answer the question.

Resume:
{context}

Question: {question}

Answer:
"""

custom_prompt = PromptTemplate(template=prompt_template, input_variables=["context", "question"])
```

# Chatbot Development

06

## Response Generation Mechanism

```
68
69     rag_chain = RetrievalQA.from_chain_type(
70         llm=hf_hub_llm,
71         chain_type="stuff",
72         retriever=vector_store.as_retriever(top_k=3), # retriever is set to fetch top 3 results
73         chain_type_kwargs={"prompt": custom_prompt})
74
75     def get_response(question):
76         result = rag_chain({"query": question})
77         response_text = result["result"]
78         answer_start = response_text.find("Answer:") + len("Answer:")
79         answer = response_text[answer_start:].strip()
80         return answer
```

# Chatbot Development

07

## Streamlit Interface Configuration - Customize the Streamlit layout to remove unnecessary whitespace and add aesthetic elements like headers and sidebar messages

```
# Streamlit app
# Remove whitespace from the top of the page and sidebar
st.markdown(
    """
        <style>
            .appview-container .main .block-container {{
                padding-top: {padding_top}rem;
                padding-bottom: {padding_bottom}rem;
            }}
        </style>""".format(
            padding_top=1, padding_bottom=1
        ),
    unsafe_allow_html=True,
)

# st.header("Explore Kaushikee's Portfolio of Excellence", divider='grey')
st.markdown("""
    <h3 style='text-align: left; padding-top: 35px; border-bottom: 3px solid purple;'>
        Explore Kaushikee's Portfolio of Excellence
    </h3>""", unsafe_allow_html=True)
```

# Chatbot Development

08

Create interactive elements such as chat input and buttons to clear the chat history, enhancing user interaction.

```
side_bar_message = """Hi there, I'm [Kaushikee](https://linkedin.com/in/kaushikee-bhawsar-54127b177), and I've developed this virtual assistant,  
**KBot**, to help you navigate through my professional background efficiently.  
Here are some key areas to consider:  
1. **Professional Experience**  
2. **Technical Skills**  
3. **Projects and Achievements**  
4. **Education and Certifications**  
  
Feel free to ask anything!  
"""  
  
with st.sidebar:  
    st.title(':purple_heart: KBot - Kaushikee\'s Virtual Assistant')  
    st.markdown(side_bar_message)  
  
initial_message = """  
    Hi there! 🙋 I'm KBot.  
    How can I assist you in exploring Kaushikee's professional expertise and accomplishments? To get started, here are some questions you can ask me:\n        What are Kaushikee's skills?\n        How many years of work experience she has?\n        What are some notable projects she has completed?\n        Which certifications has she earned?\n        What makes her a standout candidate?"""  
    # Store LLM generated responses  
if "messages" not in st.session_state.keys():  
    st.session_state.messages = [{"role": "assistant", "content": initial_message}]  
  
    # Display or clear chat messages  
for message in st.session_state.messages:  
    with st.chat_message(message["role"]):  
        st.markdown(message["content"])  
  
Tabnine: Test | Explain | Document | Ask  
def clear_chat_history():  
    st.session_state.messages = [{"role": "assistant", "content": initial_message}]  
    st.button('Clear Chat', on_click=clear_chat_history)
```

# Chatbot Development

09

## User Interaction Flow

```
# User-provided prompt
if prompt := st.chat_input():
    st.session_state.messages.append({"role": "user", "content": prompt})
    with st.chat_message("user"):
        st.markdown(prompt)

# Generate a new response if last message is not from assistant
if st.session_state.messages[-1]["role"] != "assistant":
    with st.chat_message("assistant"):
        with st.spinner("Hold on, I'm checking Kaushik's profile for you..."):
            response = get_response(prompt)
            placeholder = st.empty()
            full_response = ''
            for item in response:
                full_response += item
                placeholder.markdown(full_response)
            placeholder.markdown(full_response)
    message = {"role": "assistant", "content": full_response}
    st.session_state.messages.append(message)
```

Lets run our  
Chatbot  
application!

# Run the application

01

- Run `vector_database` implementation file using below command:  
**python vector\_embeddings.py**

```
\ResumeBot>python vector_embeddings.py
```

02

- Run our main application using below command  
**streamlit run filename.py**

```
\ResumeBot>streamlit run ResumeBot.py
```

# RAG Pipeline Evaluation

# Retrieval Metrics

- **Context Precision:** Measure how accurately the retrieved context matches the user's query.
- **Context Recall:** Evaluate the ability to retrieve all relevant contexts for the user's query.
- **Context Relevance:** Assess the relevance of the retrieved context to the user's query.
- **Context Entity Recall:** Determine the ability to recall relevant entities within the context.
- **Noise Robustness:** Test the system's ability to handle noisy or irrelevant inputs.

$$\text{Precision} = \frac{\text{Number of True Positives}}{\text{Number of Retrieved Contexts}}$$

$$\text{Recall} = \frac{\text{Number of True Positives}}{\text{Number of Relevant Contexts}}$$

$$\text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

$$\text{Entity Recall} = \frac{\text{Number of Correctly Recalled Entities}}{\text{Total Number of Entities in Query}}$$

$$\text{Noise Robustness} = \text{Relevance Score of Noisy Query}$$

# Generation Metrics

- **Faithfulness:** Measure the accuracy and reliability of the generated answers.

Faithfulness = Sigmoid Output of BERT

- **Answer Relevance:** Evaluate the relevance of the generated answers to the user's query.

BLEU Score = Geometric Mean of Precision for n-grams × Brevity Penalty

- **Information Integration:** Assess the ability to integrate and present information cohesively.

$$\text{ROUGE-1} = \frac{\text{Number of Overlapping Unigrams}}{\text{Total Number of Unigrams in Ground Truth}}$$

$$\text{ROUGE-L} = \frac{\text{Length of Longest Common Subsequence}}{\text{Total Length of Ground Truth}}$$

- **Counterfactual Robustness:** Test the robustness of the system against counterfactual or contradictory queries.

Counterfactual Robustness = Generated Answer ≠ Counterfactual Answer

- **Negative Rejection:** Measure the system's ability to reject and handle negative or inappropriate queries.

Negative Rejection = Absence of Negative Keywords in Generated Answer

# Latency

- **Latency:** Measure the response time of the system from receiving a query to delivering an answer.

$$\text{Latency} = \text{End Time} - \text{Start Time}$$

## Evaluate KBot RAG Pipeline

Deploy :

Evaluate

### Evaluation Metrics

precision: 0.0391304347826087

recall: 0.75

relevance: 0.06619520000067976

entity\_recall: 1.0

faithfulness: 0.69747394323349

bleu: 0.9036020036098448

rouge1: 1.0

rougeL: 1.0

latency: 0.07727432250976562

noise\_robustness: 0.26964269094224336

counterfactual\_robustness: 1.0

negative\_rejection: 0.0

## Explore Kaushikee's Portfolio of Excellence



Hi there! 🙌 I'm KBot. How can I assist you in exploring Kaushikee's professional expertise and accomplishments? To get started, here are some questions you can ask me:

What are Kaushikee's skills?

How many years of work experience she has?

What are some notable projects she has completed?

Which certifications has she earned?

What makes her a standout candidate?



How many years of work experience she has?



Kaushikee has 3+ years of experience in Software Development and Quality Assurance.

Your message



# Improvement Strategy - FineTuning our **LLM Model**

# Fine - Tuning LLM model

01

## Installing all required dependencies

```
[ ] %capture  
# Installs Unslot, Xformers (Flash Attention) and all other packages!  
!pip install "unslot[colab-new] @ git+https://github.com/unslotai/unslot.git"  
!pip install --no-deps "xformers<0.0.27" "trl<0.9.0" peft accelerate bitsandbytes
```

# Fine - Tuning LLM model

02

## Initializing model training parameters such as max\_seq-length, datatype (dtype) and 4bit quantization

```
[ ] from unsloth import FastLanguageModel
import torch
max_seq_length = 2048 # Choose any! We auto support RoPE Scaling internally!
dtype = None # None for auto detection. Float16 for Tesla T4, V100, Bfloat16 for Ampere+
load_in_4bit = True # Use 4bit quantization to reduce memory usage. Can be False.

# 4bit pre quantized models we support for 4x faster downloading + no OOMs.
fourbit_models = [
    "unsloth/Meta-Llama-3.1-8B-bnb-4bit",      # Llama-3.1 15 trillion tokens model 2x faster!
    "unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit",
    "unsloth/Meta-Llama-3.1-70B-bnb-4bit",
    "unsloth/Meta-Llama-3.1-405B-bnb-4bit",     # We also uploaded 4bit for 405b!
    "unsloth/Mistral-Nemo-Base-2407-bnb-4bit", # New Mistral 12b 2x faster!
    "unsloth/Mistral-Nemo-Instruct-2407-bnb-4bit",
    "unsloth/mistral-7b-v0.3-bnb-4bit",         # Mistral v3 2x faster!
    "unsloth/mistral-7b-instruct-v0.3-bnb-4bit",
    "unsloth/Phi-3-mini-4k-instruct",           # Phi-3 2x faster!
    "unsloth/Phi-3-medium-4k-instruct",
    "unsloth/gemma-2-9b-bnb-4bit",
    "unsloth/gemma-2-27b-bnb-4bit",              # Gemma 2x faster!
] # More models at https://huggingface.co/unsloth

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "unsloth/Meta-Llama-3.1-8B",
    max_seq_length = max_seq_length,
    dtype = dtype,
    load_in_4bit = load_in_4bit,
    # token = "hf...", # use one if using gated models like meta-llama/Llama-2-7b-hf
)
```

# Fine - Tuning LLM model

03

## Add LoRA adapters so we only need to update 1 to 10% of all parameters

```
[ ] model = FastLanguageModel.get_peft_model(  
    model,  
    r = 16, # Choose any number > 0 ! Suggested 8, 16, 32, 64, 128  
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",  
                      "gate_proj", "up_proj", "down_proj"],  
    lora_alpha = 16,  
    lora_dropout = 0, # Supports any, but = 0 is optimized  
    bias = "none", # Supports any, but = "none" is optimized  
    # [NEW] "unsloth" uses 30% less VRAM, fits 2x larger batch sizes!  
    use_gradient_checkpointing = "unsloth", # True or "unsloth" for very long context  
    random_state = 3407,  
    use_rslora = False, # We support rank stabilized LoRA  
    loftq_config = None, # And LoftQ  
)
```

# Fine - Tuning LLM model

04

## Data preparation and llm prompt setup

```
llm_prompt = """Below is an instruction that describes a task. Write a response that appropriately completes the request.

### Instruction:
{}

### Response:
{}"""

EOS_TOKEN = tokenizer.eos_token # Must add EOS_TOKEN
def formatting_prompts_func(examples):
    instructions = examples["instruction"]
    outputs      = examples["output"]
    texts = []
    for instruction, output in zip(instructions, outputs):
        # Must add EOS_TOKEN, otherwise your generation will go on forever!
        text = llm_prompt.format(instruction, output) + EOS_TOKEN
        texts.append(text)
    return { "text" : texts, }
pass

from datasets import load_dataset
dataset = load_dataset("kaushik01/AboutMe", split = "train")
dataset = dataset.map(formatting_prompts_func, batched = True,)
```

# Fine - Tuning LLM model

Below is the snipped of the dataset used to train the model for fine tuning.

The screenshot shows the Hugging Face Dataset card for the dataset 'kaushikee01/AboutMe'. The card includes a search bar, navigation links for 'Models' and 'Datasets', and dataset metadata like modalities (Text), formats (csv), size (<1K), libraries (Datasets, pandas, Croissant), and license (mit). Below the metadata is a navigation bar with tabs: 'Dataset card' (selected), 'Viewer', 'Files and versions', 'Community (2)', and 'Settings'. The main area is titled 'Dataset Viewer' and shows a split view: 'instruction' (string - lengths) and 'output' (string - lengths). It includes a histogram for each, with values 33, 110 for 'instruction' and 54, 713 for 'output'. Below the histograms are five examples of input-output pairs:

instruction	output
What is Kaushikee Bhawsar's professional summary?	Kaushikee Bhawsar is a Software Engineer with 3+ years of experience in Software Development and Quality Assurance. Adept...
What are Kaushikee Bhawsar's key programming skills?	Kaushikee is proficient in programming languages such as C++, Java, Python, and SQL.
What quality assurance skills does Kaushikee possess?	Kaushikee has expertise in ETL/Data Testing, Web & Application Testing, Mobile Testing, Performance Testing, API Testing,...
Which automation tools does Kaushikee use?	Kaushikee is skilled in using Selenium WebDriver, TestNG, and BDD Cucumber for automation testing.

# Fine - Tuning LLM model

05

## Print dataset to confirm

```
print(dataset['text'][0])
```

Below is an instruction that describes a task. Write a response that appropriately completes the request.

```
### Instruction:  
What is Kaushikee Bhawsar's professional summary?
```

```
### Response:  
Kaushikee Bhawsar is a Software Engineer with 3+ years of experience in Software Development and Quality Assurance. Adept at leveraging cutting-edge tools and methodologies to deliver
```

# Fine - Tuning LLM model

06

## Train the model for fine tuning using Supervised fine tuning technique

```
from trl import SFTTrainer
from transformers import TrainingArguments
from unsloth import is_bfloat16_supported

trainer = SFTTrainer(
    model = model,
    tokenizer = tokenizer,
    train_dataset = dataset,
    dataset_text_field = "text",
    max_seq_length = max_seq_length,
    dataset_num_proc = 2,
    packing = False, # Can make training 5x faster for short sequences.
    args = TrainingArguments(
        per_device_train_batch_size = 2,
        gradient_accumulation_steps = 4,
        warmup_steps = 5,
        # num_train_epochs = 1, # Set this for 1 full training run.
        max_steps = 60,
        learning_rate = 2e-4,
        fp16 = not is_bfloat16_supported(),
        bf16 = is_bfloat16_supported(),
        logging_steps = 1,
        optim = "adamw_8bit",
        weight_decay = 0.01,
        lr_scheduler_type = "linear",
        seed = 3407,
        output_dir = "outputs",
    ),
)
```

# Fine - Tuning LLM model

07

## Run Trainer upto maximum 60 steps

```
trainer_stats = trainer.train()

==((=====))= Unislot - 2x faster free finetuning | Num GPUs = 1
    \\ /| Num examples = 96 | Num Epochs = 5
  0^0/ \_/\ Batch size per device = 2 | Gradient Accumulation steps = 4
  \ / Total batch size = 8 | Total steps = 60
  "-__-" Number of trainable parameters = 41,943,040
[60/60 04:25, Epoch 5/5]
```

Step	Training Loss
1	2.931400
2	2.904800
3	2.704200
4	2.607200
5	2.415200
6	2.188900
7	1.741500
8	1.493700
9	1.309700
10	1.101500
11	1.185900
12	1.029200

# Fine - Tuning LLM model

08

## Inference – Run the model

Provide instruction and left the output blank to be generated by Fine-tuned LLM

```
[ ] # llm_prompt = Copied from above
FastLanguageModel.for_inference(model) # Enable native 2x faster inference
inputs = tokenizer(
[
    llm_prompt.format(
        "what certifications done by kaushikee?", # instruction
        "", # output - leave this blank for generation!
    )
], return_tensors = "pt").to("cuda")

outputs = model.generate(**inputs, max_new_tokens = 64, use_cache = True)
tokenizer.batch_decode(outputs)
```

→ ['<|begin\_of\_text|>Below is an instruction that describes a task. Write a response that appropriately completes the request.\n\n### Instruction:\nwhat certifications done by kaushikee?\n\n### Response:\nKaushikee has certifications in Oracle Cloud Infrastructure Generative AI and Database Management Systems.<|end\_of\_text|>']

# Fine - Tuning LLM model

09

## Saving the model locally as well as on Huggingfacehub

```
[ ] from google.colab import userdata  
HF_TOKEN = userdata.get('HF_TOKEN')  
  
# model.save_pretrained("Llama_3_1_finetuned_model") # Local saving  
# tokenizer.save_pretrained("Llama_3_1_finetuned_model")  
  
model.push_to_hub("kaushikee01/Llama_3_1_finetuned_model", token = HF_TOKEN) # Online saving  
tokenizer.push_to_hub("kaushikee01/Llama_3_1_finetuned_model", token = HF_TOKEN) # Online saving
```

→ README.md: 100%  592/592 [00:00<00:00, 43.1kB/s]  
Saved model to [https://huggingface.co/kaushikee01/Llama\\_3\\_1\\_finetuned\\_model](https://huggingface.co/kaushikee01/Llama_3_1_finetuned_model)

```
[ ] model.save_pretrained("Llama_3_1_finetuned_model") # Local saving  
tokenizer.save_pretrained("Llama_3_1_finetuned_model")  
  
→ ('Llama_3_1_finetuned_model/tokenizer_config.json',  
 'Llama_3_1_finetuned_model/special_tokens_map.json',  
 'Llama_3_1_finetuned_model/tokenizer.json')
```

# Fine - Tuning LLM model

10

## Verify the uploaded model on Hugging face:

The screenshot shows the Hugging Face Model Card for the model `kaushikee01/Llama_3_1_finetuned_model`. The card includes the following details:

- Model Name:** kaushikee01/Llama\_3\_1\_finetuned\_model
- Developed by:** kaushikee01
- License:** apache-2.0
- Finetuned from model:** unsloth/meta-llama-3.1-8b-bnb-4bit
- Tags:** Transformers, Safetensors, English, text-generation-inference, unsloth, llama, TRL, Inference Endpoints
- Community:** Includes links for Model card, Files and versions, Community, and Settings.
- Note:** This llama model was trained 2x faster with [Unsloth](#) and Huggingface's TRL library.



# Utilizing FineTuned Model in Chatbot Implementation

01

## By Initializing the HuggingfaceHub LLM with Finetuned model

```
# Initialize the Hugging Face Hub LLM

hf_hub_llm = HuggingFaceHub(
    repo_id="kaushikee01/Llama_3_1_finetuned_model",
    model_kwargs={"temperature": 0.01, "max_new_tokens":1024},
)
# hf_hub_llm = HuggingFaceHub(
#     repo_id="meta-llama/Meta-Llama-3-8B-Instruct",
#     model_kwargs={"temperature": 1, "max_new_tokens":1024},
# )
```

02

## Run the Chatbot with Finetuned Model

# Application Demo and Testing

Kaushikee()

Home Education and Certifications Experience Projects Contact and Resume Talk to my KBot

## Hey!

I'm Kaushikee Bhawsar.

Web Developer, Generative AI Enthusiast, Quality Engineer. Always learning!



[Contact Me](#)



Anmol.V

Home Education and Certifications Experience Projects Contact and Resume Chat with Me..

## Hello.

I'm Anmol Valecha. Full Stack Developer, Machine Learning Enthusiast. Always learning.



[Contact Me](#)



# Testing with Simple Queries:

 What are Kaushikee's skills?

 Kaushikee's skills include Programming Languages such as C++, Java, Python, and SQL. She is also proficient in Automation Tools like Selenium WebDriver, TestNG, and BDD Cucumber. Additionally, she has expertise in Quality Assurance, including ETL/ Data Testing, Web & Application Testing, Mobile Testing, Performance Testing, API Testing, Usability Testing, Regression Testing, Black-Box Testing, and Unit Testing. Furthermore, Kaushikee has skills in Cloud Technologies, Database Management, CI/CD & DevOps, Web Technologies, and Generative AI, including AWS, Databricks, Azure DevOps, and Azure Data Factory. She is also familiar with various tools and libraries such as SQL Server management studio, Azure Cosmos DB, SQL Developer, Jenkins, Docker, HTML, CSS, JavaScript, Bootstrap, Angular17, React, JMeter, Postman, Gatling, Junit, Java Swing, Figma, JIRA/JTMF, NumPy, Pandas, Scikit-learn, Git/Github, Bitbucket, and Power BI.



What makes her a standout candidate?



Kaushikee's standout qualities include her proficiency in leveraging cutting-edge tools and methodologies to deliver high-quality, scalable, and innovative solutions in agile environments. Her experience in software development and quality assurance, combined with her certifications in Oracle Cloud Infrastructure and NPTEL IIT Kharagpur Database Management System, make her a strong candidate. Additionally, her ability to orchestrate data pipelines with Azure Data Factory and earn appreciation from her team lead and team members for her consistent and results-oriented approach further solidify her position as a standout candidate. Her achievements in reducing testing time and boosting efficiency, as well as her experience in quality assurance, make her an attractive candidate for any organization.

# Testing with Complex Queries:

 What are her accomplishment in TD Bank and tell me why should i hire kaushikee by describing her in one word?

 Kaushikee's accomplishments in TD Bank include:

- Achieving a 40% reduction in testing time and a 30% boost in efficiency for Customer Risk Rating data using BDD Cucumber, Java, and Jenkins.
- Executing ETL processes for data validation using SQL, Python/PySpark in Databricks, achieving a high accuracy rate by meticulously processing millions of records and eliminating bad data.
- Orchestrating data pipelines with Azure Data Factory, ensuring seamless and efficient data extraction, transformation, and data loading processes, contributing to overall data quality and reliability.

In one word, I would describe Kaushikee as: **Achiever**. She has consistently demonstrated her ability to deliver high-quality results, improve efficiency, and drive innovation in her previous roles. Her accomplishments in TD Bank showcase her technical skills, problem-solving abilities, and dedication to achieving high-quality outcomes. Hiring Kaushikee would be a great decision as she would bring a high level of expertise, enthusiasm, and commitment to her work.



Describe top 3 skills and her years of experience



Kaushikee has 3+ years of experience in Software Development and Quality Assurance. Her top 3 skills are:

1. Automation Tools: She is proficient in Selenium WebDriver, TestNG, and BDD Cucumber.
2. Cloud Technologies: She has expertise in AWS, Databricks, Azure DevOps, and Azure Data Factory.
3. Generative AI: She holds certifications in Oracle Cloud Infrastructure (OCI) 2024 Generative AI Professional Certification and has experience with tools like AWS, Databricks, Azure DevOps, and Azure Data Factory.

# Testing with Irrelevant/Out of Scope Queries:

 hdbkjf helsdnsa

 I'm not able to answer the question as it seems to be a non-resume-related query. Please rephrase the question or provide a new one that is relevant to the resume. I'm here to assist you with any questions you may have about Kaushikee's resume.

 what are her travel plans?

 I don't have any information about Kaushikee's travel plans in her resume. As her intelligent assistant, I can suggest that you may want to ask her directly about her travel plans.

Contact details: Email: [kaushikeebhawsar1999ca@gmail.com](mailto:kaushikeebhawsar1999ca@gmail.com) LinkedIn: [linkedin.com/in/kaushikee-bhawsar-54127b177/](https://linkedin.com/in/kaushikee-bhawsar-54127b177/) Github: [github.com/kaushikeebhawsar99](https://github.com/kaushikeebhawsar99)

 What movies kaushikee watch?

 There is no information about Kaushikee's movie preferences in the provided resume. The resume only focuses on Kaushikee's professional experience, education, and skills. Therefore, it is not possible to answer this question based on the provided information. If you have any other questions, I'll be happy to help.

# Testing with Misspelled Queries:



kaushiki professional exprience



Kaushikee has 3+ years of professional experience as a Software Engineer with expertise in Software Development and Quality Assurance. She has worked on various projects, leveraging cutting-edge tools and methodologies to deliver high-quality, scalable, and innovative solutions in agile environments. Her experience includes working with Automation Tools, Quality Assurance, Cloud Technologies, Database Management, CI/CD & DevOps, Web Technologies, and Generative AI. She has successfully implemented AI-based Chatbots, reducing manual workload



# Deployment Plan

## Steps to deploy:

- Deploy chatbot app
- Integrate chatbot app into portfolio website
- Deploy Portfolio website

## Tools and Platforms:

- Netlify for portfolio website hosting
- Streamlit for Chatbot



Kaushikee()

Home Education and Certifications Experience Projects Contact and Resume Talk to my KBot



Hey!

I'm Kaushikee Bhawsar.

Web Developer, Generative AI Enthusiast, Quality Engineer. Always learning!

[Contact Me](#)

Q in G @

## Kaushikee Bhawsar portfolio

- **Application url:** <https://kaushikeebhawsar.netlify.app/>
- **Chatbot url:** - <https://kaushikee.streamlit.app/>

Anmol.V

Home Education and Certifications Experience Projects Contact and Resume Chat with Me..



Hello.

I'm Anmol Valecha. Full Stack Developer, Machine Learning Enthusiast. Always learning.

[Contact Me](#)

Q in G Twitter @

## Anmol Valecha portfolio

- **Application url:** <https://anmolvalechaproject.netlify.app/#/>
- **Chatbot url:** - <https://portfolioavbot.streamlit.app/>



# Future Work

## Potential Extensions

- Add more interactive features (e.g., blog, tutorial section).
- Expand chatbot capabilities (e.g., schedule appointments).
- Integration with Social Media
- Implement a testimonials/recommendation section where clients, colleagues, or users can provide feedback on work or services.

## Long-term Vision:

- Create a comprehensive, AI-powered personal assistant.
- Continuously improve based on user feedback and technological advancements.
- Explore voice-based interactions with the chatbot.



# Conclusion

## Summary:

- Personal portfolio website with integrated RAG chatbot.
- Demonstrates skills in web development and AI integration.

## Key Takeaways:

- Practical application of course concepts.
- Valuable project for showcasing technical abilities.

## Final Thoughts:

- Excited to present and receive feedback.
- Looking forward to future improvements.





# **THANK YOU!**

