

Short Resume

Sumit Chauhan

Email: sumit078385@gmail.com

Phone: +91 7982993214

LinkedIn: [Link](#)

Github: [Link](#)

Portfolio: [Link](#)

Profile Summary

Targeting **AI/ML Engineer** roles with a focus on leveraging **Machine Learning, Artificial Intelligence,** and **Python programming** to solve real-world problems. Experienced in building models for classification, NLP tasks, and deploying AI-powered applications.

Key Projects

1. Resume Screening System Using Gemini Pro API

- **Tools:** Python, Streamlit, Gemini Model API
- Developed an AI-powered system to analyze resumes against job descriptions, providing keyword matching, scoring, and improvement suggestions.

2. Credit Card Fraud Detection System

- **Tools:** Python, TensorFlow, Streamlit
- Built a web application to detect fraudulent transactions using machine learning models.

3. Diet and Fitness Recommendation Chatbot

- **Tools:** Python, Streamlit, FAISS, Gemini Pro API
 - Created a chatbot for personalized health and fitness advice, integrating calorie tracking and dynamic recommendations.
-

Skills

- **Machine Learning & AI:** TensorFlow, Scikit-learn, Keras, NLTK, OpenCV
- **Data Science Tools:** Pandas, NumPy, Matplotlib, Seaborn, Jupyter Notebooks
- **Deep Learning & NLP:** NLP, Computer Vision, CNN, RNN, LSTMs, Transformers
- **Tools & Frameworks:** Git, Streamlit, Docker, AWS Cloud Platform

Explanation of the Implementation

This is a **research and summarization system** that leverages **Tavily** for web search and **Google's Generative AI (Gemini)** for summarization. Here's a detailed breakdown of its implementation:

1. Environment Setup

- **Purpose:** The code imports necessary libraries and loads environment variables from a .env file.
 - **Key Libraries:**
 - TavilyClient: For performing web searches.
 - google.generativeai: For accessing Google's Gemini AI model.
 - langgraph: For managing the workflow between research and drafting agents.
 - dotenv: To securely load API keys from the environment.
-

2. Configuration and Initialization

- **Purpose:** Configures the Gemini AI model and initializes the Tavily client with API keys.
 - **Key Steps:**
 - genai.configure: Sets up the Gemini API key.
 - llm: Initializes the Gemini 1.5 Pro model for generating summaries.
 - tavily: Initializes the Tavily client for web searches.
-

3. Research Agent

- **Purpose:** Performs a web search using Tavily based on the user's query.
 - **Key Steps:**
 - tavily.search: Executes the search and retrieves results.
 - Returns the search results along with the existing state.
-

4. Drafting Agent

- **Purpose:** Summarizes the research results using the Gemini model.
 - **Key Steps:**
 - Combines the content from all research results into a single context string.
 - Uses llm.generate_content to generate a summary.
 - Returns the summarized answer along with the research results and query.
-

5. Workflow Construction

- **Purpose:** Defines the workflow for the research and summarization process.

- **Key Steps:**
 - Initializes a StateGraph to manage the workflow.
 - Adds nodes for the research and drafting agents.
 - Connects the nodes to define the flow: research → drafting → end.
 - Compiles and returns the workflow.
-

6. Running the Research System

- **Purpose:** Executes the workflow with a user-provided query.
 - **Key Steps:**
 - Builds the workflow using build_workflow.
 - Invokes the workflow with an initial state containing the query.
 - Returns the summarized answer generated by the drafting agent.
-

7. Main Execution

- **Purpose:** Runs the research system with a sample query and prints the result.
- **Key Steps:**
 - Calls run_research_system with a query (e.g., "Impact of AI on cybersecurity in points").
 - Prints the summarized answer.