

Hitesh Bandhu

+91 7717541984 bandhuhitesh52003@gmail.com @.hiteshbandhu @hiteshbandhu hitesh-bandhu

Education

UIET, Panjab University
Bachelor of Engineering in ECE

May 2025
Chandigarh, India

Experience

Reliafin Inc.
Founding Engineer

March 2024 - June 2024
Remote - USA

- Used **Django Rest Framework** to build the backend system for handling **AI specific requests** with complete user authentication and **AWS** integration.
- Designed and Implemented the **RAG pipelines**, using **langchain** for consumers to **chat with advisors**.
- **Researching for AI projects**, to be executed in the future, including but not limited to **reading research papers**, technical documentations and **writing SOPs**.

Projects

- **youDocs (beta)** *Gemini API, Typer, FastAPI*
 - Tool that automatically generates **detailed documentation** and visual graphs from **YouTube playlists**, so that the users can read the content as if it is a **well written blog**.
 - The backend is **served using FastAPI**. User enters the youtube links, the playlist **videos are fetched**, and their **transcript** is sent to the **GEMINI API** to generate the content. Gemini was used because of its **2M tokens** context window
 - Instead of making a GUI for the app, i made it a **terminal utility** using Typer Package, the user can just write **'youdocs -l link'** and get the blog served at **localhost:3000**
- **LocalTranscribe** *Python, RAG, Whisper, OpenAI, FastAPI, Next.js*
 - A terminal application that **transcribes spoken ideas** in real-time, converts them into **written content**, and organizes them within a **user-friendly dashboard** to monitor. You can also **track your thought processes** over time. Tech used for Dashboard is **Next.js + shadcn/ui** for styling
 - **Whisper's "tiny" model** checkpoint is used on **local inference** for fast real-time transcription..
 - The Transcription is sent to the **OpenAI API** to analyse and result is stored in a **'neon.tech postgres DB'**
- **NeuraLeverage** *Python, Next.js, Django, Langchain*
 - This is my answer to **"Leetcode for Prompt Engineering"**. Try problems, learn **prompt techniques**, prompt injections just like Data Structures problems on Leetcode
 - Used **Retrieval Augmented Generation (RAG)** to create quizzes that adapt to users' skill levels, providing a **personalized learning experience** that evolves with the user's progress.
 - The Frontend was coded **Next.js** and **Django** was used to handle the backend due to complexity of the project and to use the **python packages in the ingestion** and data **processing pipelines**

Technical Skills

Languages: Python, C, C++, JavaScript

Technologies: Django, Flask, Django REST Framework, Numpy, Pandas, Matplotlib, Streamlit (Python); Next.js (JavaScript);

Databases: PostgreSQL, MongoDB, Supabase, Pinecone (Vector Database)

GenAI Stack: Langchain, LangGraph, RAGAS Evaluations, PromptLayer, OpenAI API, Whisper API, Google Gemini API, HuggingFace, MonsterAPI, ClaudeAI, Prompt Engineering

Hackathons, Competitions and Quizzes

- **1st Position** at a remote hackathon organised by **Textify AI**. Developed an **AI social bio writer tool** using prompting
- **Rank 9 out of 300+ students** in an all-round quiz on **Artificial Intelligence** in the college campus.
- Won several **dev bounties** on twitter ranging from **web scraping**, to **AI Voice agents**. Compensation range : **\$30-\$650**