

Om Bhaltilak

B.Tech

Artificial Intelligence And Data Science

CGPA:7.82

Pune

Mob: +91-9767910617

Email: omnbholtlak@gmail.com

Links

[Github](#)

[LeetCode](#)

[HackerRank](#)

[LinkedIn](#)

Skills

Programming Languages

Python, C++, JavaScript

Generative AI & Frameworks

Agentic RAG, LangChain,
Prompt Engineering,
Vector Databases (FAISS, ChromaDB)

AI, ML & Data Science

NLP, Deep Learning,
Machine Learning, Pandas,
NumPy, Matplotlib, Seaborn

Databases & BI Tools

SQL (MySQL), MongoDB,
Power BI, Tableau

Web Development

HTML, CSS, JavaScript, React.js,
Tailwind CSS, Bootstrap

Other Skills

DSA,OOP

Education

2022–2026

B.Tech. in AI & Data Science

AISSMS Institute of Information
Technology, Pune
CGPA : 7.82

2021–2022

12th

Vidyanchal The School
Marks: 73%

2019–2020

10th

Shri Bhausaheb Pote Vidyalaya
Marks: 88.6%

Certifications

[The Joy of Computing using Python](#)
[OCI 2025 Generative AI Professional](#)

Experience

Infosys Springboard

FEB 2025 – MAR 2025 Intern

AI-powered Fraud Management System for UID Aadhaar

- Designed and developed an AI-powered Fraud Management System for UID Aadhaar that automated document verification using OCR pipelines, classification models, and YOLO-based detection.
- Applied advanced image preprocessing (resizing, grayscaling, normalization, contrast enhancement) to improve accuracy on noisy inputs.
- Automated fraud scoring with structured Excel/JSON outputs, reducing manual errors and improving efficiency for government, banking, and e-KYC workflows.
- Built with Python, TensorFlow/PyTorch, YOLO, OpenCV, Tesseract OCR, Pandas, NumPy, Flask, and proposed upgrades such as multi-lingual OCR, cloud integration, and real-time fraud alerts.

Projects

YouTube RAG Assistant – Multilingual Video AI Chatbot

[GitHub](#)

- Architected a full-stack AI browser extension and scalable backend allowing users to chat with long-form YouTube videos in real-time using an advanced Retrieval-Augmented Generation (RAG) pipeline.
- Engineered a robust data ingestion pipeline featuring language detection, on-the-fly translation, and sliding-window chunking with semantic overlap to maintain context boundaries across large transcripts.
- Built a two-stage retrieval pipeline using FAISS with all-MiniLM embeddings for high-speed vector search, followed by a MS-MARCO Cross-Encoder for reranking retrieved documents to maximize context precision.
- Implemented Agentic RAG features including LLM-driven query rewriting, guardrailing, and dynamic intent routing.
- Designed a strided document sampling strategy (map-reduce style) to accurately summarize multi-hour videos without exceeding LLM context windows or losing global context.
- Utilized strict prompt engineering with Qwen2.5-7B-Instruct to generate hallucination-free, dynamically sized answers, strictly enforced with multiple inline clickable video timestamps [MM:SS].
- Tech Stack: Python, LangChain, FAISS, Hugging Face Inference Endpoints (Qwen 2.5), Sentence-Transformers (Cross-Encoders), Flask, Docker, JS.

Offline Document Finder (ODF) – AI Semantic Search Engine

[GitHub](#)

- Architected and developed a fully offline, privacy-first AI-powered desktop search engine enabling semantic retrieval across local documents (PDF, DOCX, XLSX, TXT) using natural language queries instead of exact keyword matching.
- Designed a high-performance semantic indexing pipeline using FastEmbed with ONNX Runtime, dynamically leveraging NVIDIA CUDA for GPU acceleration while maintaining optimized CPU-only fallback execution.
- Implemented persistent local vector storage using ChromaDB, enabling efficient similarity search with disk-based embedding storage for scalable document indexing.
- Built a lightweight desktop interface using CustomTkinter with global system-wide hotkey support (Ctrl + K) for instant search access.
- Packaged the application into a standalone Windows executable using PyInstaller, ensuring zero external dependencies and 100% local execution.
- Tech Stack: Python, ChromaDB, FastEmbed, ONNX Runtime (CPU/GPU), CustomTkinter, PyInstaller, PyMuPDF (Fitz), OpenPyXL.