

V NAMAN MEHTA

+91 9980945993

✉ vnamanmehta.dev@gmail.com

in [v-naman-mehta](#)

🔗 [VNamanMehta](#)

SUMMARY

Building production software end to end, from desktop apps and backend APIs to the AI pipelines behind them, mostly in React and Python. Recently that's meant exam security software and AI-driven document processing running at real scale. Looking to take on technical ownership, from system design and architecture to building and shipping reliable products.

EDUCATION

Dayananda Sagar College of Engineering

B.E. in Computer Science and Engineering

CGPA: 9.0/10

2022 – 2026

TECHNICAL SKILLS

Languages: Python, JavaScript, TypeScript, C++

Frontend & Backend: React, Next.js, Electron, Flask, FastAPI, Node.js

AI & ML: LangChain, FAISS, OpenAI/Gemini APIs, NEAT-Python, ComfyUI

Databases & Cloud: PostgreSQL, Supabase, Prisma, Docker, Azure, GitHub Actions, Git

EXPERIENCE

[Talview](#)

Associate Software Developer

May 2026 – Present

- Built the Secure Browser's **React** front end and set up **Rsbuid** as the build tool for the Electron app, packaged across 4 installer formats (EXE, MSI, MSIX, and DMG) for Windows and macOS.
- Took the **Secure Browser** from prototype to release, rewriting the C++ native agent as a background service: a **Windows service** over **named pipes** and a **macOS daemon** over **Unix sockets**, carried through in-house alpha testing.
- Built **12+ security controls** for exams: process monitoring, screen recording prevention, virtual machine detection, and virtual camera blocking, using low-level OS APIs.
- Added remote proctoring with Twilio video and nut.js, getting real-time device control and monitoring down to **30–40 ms latency**.

Software Developer Intern

Sep 2025 – Apr 2026

- Built the first version of the **Secure Browser** with **Electron**, **React**, and a custom C++ native agent, including a policy system for **30+ process restrictions**, **25+ approved domains**, and automatic version checks.
- Compared **Electron** and **Tauri** on **10+ criteria**, including native integrations, multi-webview support, packaging, and ecosystem maturity, and went with Electron.

[Liquidmind.ai](#)

Product Intern

Jun 2025 – Aug 2025

- Wrote the backend in **Python** (Flask, FastAPI), wiring in Azure Functions, Container Apps, and Blob Storage to automate the pipeline end to end, cutting processing time from **2–3 days** to **under 5 minutes**.
- Built an AI document processing platform for international trade paperwork, combining OCR, Azure OpenAI, and rule-based checks to hit **95–98% field extraction accuracy**.
- Split the system into **4 containerized microservices** (ingestion, AI extraction, validation, report generation) on Docker and Azure Container Registry, handling **500+ shipment document packages a day**.

PROJECTS

Multimodal Document Retrieval and Q&A Tool | *Python, Flask, Next.js, LangChain, FAISS* | [GitHub](#)

- Built a RAG app in **Python** that pulls text and images out of PDFs, embeds them with OpenAI CLIP, and indexes everything in a FAISS vector store for semantic search.
- Built the frontend in Next.js/TypeScript and backend in Flask, with LangChain tying it together so users can ask questions about their documents and get answers grounded in the actual content, via the Gemini API.

Small Language Model – GPT-style Transformer | *Python* | [GitHub](#)

- Built a GPT-style transformer language model in **Python** from scratch: custom BPE tokenizer, 8 layers, 8 attention heads, about 9 million parameters, trained on the TinyStories dataset.
- Wrote the full training pipeline, including parallelized data preparation with memory-mapped files and checkpointing, getting the model to a validation loss of 1.35 and a perplexity of 3.84.

ACHIEVEMENTS

- Published envanalyzer, a TypeScript CLI tool, to npm as a 2-package suite that solves cross-file dynamic environment variable tracing; it's passed **300+ downloads**.