Detailed Tech Stack Overview

Development Environment & Coding Tools

- **Visual Studio Code:** Commonly using IDE for my projects. I frequently use extensions like Prettier, GitLens, Live Server, and Python for debugging. Strengths: fast and versatile.
- **PyCharm:** Used on large-scale Python/Django projects due to its powerful debugging and refactoring tools. Experience level: Intermediate. Advantage: built-in Django and virtual environment support.
- **Kiro:** A modern code editor I use experimentally for minimal distraction-free coding. Useful for focused sessions.
- **Cursor:** An AI-powered IDE that integrates coding assistance. I use it for experimenting with agentic workflows. Advantage: speeds up prototyping.
- **Git & GitHub:** Core to my workflow for version control and collaboration. Experienced with branching, pull requests, and CI/CD integrations. Advantage: industry-standard.
- **Postman:** Main tool for testing and automating APIs. I create collections and run environments for backend testing. Advantage: robust ecosystem.
- **Hoppscotch API:** A lightweight alternative to Postman I use for quick API testing in the browser. Advantage: fast and open-source.

Web Frameworks

- **Django:** My primary backend framework. Experience level: Proficient. I use it to build secure, scalable apps with ORM and authentication. Advantage: batteries-included.
- **React.js:** Frontend framework I use for interactive UIs. I’ve built dashboards and college apps with it. Strength: component-based structure.

AI/ML Frameworks

- **TensorFlow:** Used for training deep learning models. Experience: beginner. Advantage: production-ready and supports TPU.
- **PyTorch:** My preferred deep learning framework due to flexibility and ease of debugging. Advantage: dynamic computation graph.
- **scikit-learn:** My Preferred Choice for classical ML algorithms and preprocessing. Experience: beginner. Advantage: simple API.

Agentic AI Tools

- **LangChain:** I use it to build LLM-powered pipelines and agents. Experience: beginner. Advantage: modular and growing ecosystem.

- **Auto-GPT:** Experimented with autonomous agent workflows. Advantage: innovative automation. Limitation: unpredictable results

LLM Platforms

- **OpenAI API:** I use GPT models for chatbots, summarization, and app integrations. Experience: strong. Advantage: state-of-the-art models. Limitation: cost and rate limits.
- **Hugging Face Transformers:** I leverage it for fine-tuning and experimenting with custom NLP models. Advantage: large model hub. Limitation: training requires significant resources.

AI Tools You Frequently Use

- **ChatGPT:** My everyday assistant for brainstorming, debugging, and drafting documents. Advantage: versatile. Limitation: may produce hallucinations.
- **Copilot:** Integrated in coding workflow for autocompletion. Advantage: boosts productivity. Limitation: occasional irrelevant suggestions.
- **Google Colab:** Preferred for quick ML experiments in the cloud. Advantage: free GPU access. Limitation: session timeouts.
- **Gemini:** Used for experimenting with multimodal AI. Advantage: advanced reasoning. Limitation: limited public access.
- **Deepseek:** Tool for focused AI assistance in research tasks. Advantage: fast. Limitation: smaller ecosystem.
- **Mistral.ai:** Experimenting with lightweight, open-source LLMs. Advantage: cost-efficient. Limitation: smaller community support.

Cloud Platforms

- **Google Cloud Platform:** I use GCP mainly for hosting AI workloads and cloud storage. Experience: intermediate. Advantage: strong AI integrations. Limitation: pricing complexity.

Databases

- **MySQL:** Used for relational database management in web apps. Experience: strong. Advantage: widely supported. Limitation: less optimized for large-scale analytics.
- **PostgreSQL:** Preferred for advanced features like JSONB and indexing. Advantage: robust and scalable. Limitation: slightly steeper learning curve.