

DEV PATEL

Aspiring AI Engineer | Full-Stack Developer

+91 70433 38711 | pateldev2317@gmail.com | github.com/PATELDEV26 | linkedin.com/in/dev-patel-66363b257

SUMMARY

3rd-year CSE student at Parul University moving toward AI engineering, backed by a habit of actually shipping things. Over the past year I've independently built six full-stack projects end-to-end — including CineMatch, a content-based recommender using cosine similarity across genre, cast and director features. I'm now deliberately building out my AI/ML foundation through hands-on coursework in generative AI and prompt engineering, and I'm looking for an internship where I can apply both my software engineering base and growing AI skillset from day one.

TECHNICAL SKILLS

AI / ML & Data: Generative AI Concepts, Prompt Engineering, Recommender Systems (cosine similarity), Pandas, NumPy, Matplotlib, Web Scraping

Languages: Python, JavaScript (ES6+), C, C++, HTML5, CSS3

Frontend: React.js, Tailwind CSS, Vanilla JS, Anime.js, Canvas API

Backend & APIs: Node.js, Express.js, REST APIs, Socket.io, WebSockets

Databases: MongoDB, SQL (basics)

Tools & Platforms: Git, GitHub, VS Code, Figma, Linux CLI, npm, AWS (Prompt Engineering)

CS Fundamentals: DSA, OOP, OS, DBMS, Computer Networks, System Design basics

CERTIFICATIONS

- **Foundations of Prompt Engineering** — AWS Training & Certification, 2026
- **What Is Generative AI?** — LinkedIn Learning, 2026
- **Kaggle — Python** — Colin Morris & Alexis Cook, Kaggle, May 2024
- **Kaggle — Intro to Programming** — Alexis Cook, Head of Kaggle Learn, May 2024

Currently completing additional coursework in AI/ML — Google AI, IBM AI Fundamentals, Microsoft Azure AI Fundamentals

PROJECTS

CineMatch github.com/PATELDEV26/cinematch

Stack: Python, Pandas, TMDb API

- Content-based movie recommender using cosine similarity across genre, cast and director features to generate personalized suggestions
- Pulled live metadata from the TMDb API; validated recommendation quality through informal testing with friends

VisualDSA github.com/PATELDEV26/VisualDSA

Stack: JavaScript, HTML5 Canvas, CSS3

- Interactive visualizer for 10+ sorting and graph algorithms (bubble, merge, quick sort, BFS, DFS, Dijkstra) with adjustable animation speed
- Built using the Canvas API, optimized to stay smooth on lower-end machines; used personally for exam prep

DevBoard github.com/PATELDEV26/Devboard

Stack: React.js, GitHub REST API, Chart.js

- Personal developer dashboard visualizing GitHub activity — commits, repos, and languages — in one view
- Implemented OAuth token auth for API access; first project with a properly componentized React structure

NeonTalk github.com/PATELDEV26/NeonTalk

Stack: Node.js, Express, Socket.io

- Full-stack real-time chat application with multi-room support, built to understand bi-directional communication
- Handled concurrent users, room join/leave events, and message broadcasting with consistently low local latency

RetroScrape github.com/PATELDEV26/RetroScrape

Stack: Python, BeautifulSoup, Requests

- CLI web scraping tool automating configuration, parsing, and export to JSON/CSV in a single command

Litera github.com/PATELDEV26/Litera

Stack: JavaScript, Open Library API

- Framework-free book discovery platform for searching titles and managing a personal reading list, under 400 lines of vanilla JS

EXPERIENCE

Freelance Web Developer

2024 – Present

- Delivered small client projects — landing pages, portfolio sites, simple tools — owning the process from initial brief through deployment
- Developed strong client communication and iteration skills, translating vague requirements into working products

EDUCATION

B.Tech — Computer Science Engineering

2023 – 2027

Parul University, Vadodara, Gujarat | CGPA: 7+ / 10

- Relevant coursework: Data Structures & Algorithms, DBMS, Operating Systems, Computer Networks, OOP

ADDITIONAL NOTES

- All projects above were built independently — no step-by-step tutorials, no starter templates
- Comfortable reading documentation and problem-solving independently; most skills were learned by building, breaking, and fixing
- Actively exploring system design and competitive programming to strengthen core problem-solving