

Dev Patel

Boston, MA • +1 (617) 415-3674 • patel.devani@northeastern.edu • [linkedin.com/in/devpatel122003](https://www.linkedin.com/in/devpatel122003) • github.com/devpatel122003

EDUCATION

Master of Science in Computer Science, Northeastern University (GPA 3.4/4.0) Sep 2025 – May 2027
B.Tech. in Computer Engineering, Pandit Deendayal Energy University (GPA 3.69/4.0) Sep 2021 – May 2025

WORK EXPERIENCE

Applied AI/ML Engineer Intern Cambridge, MA
Julia AI, LLM services and agent workflows Sep 2026 – Present

- Building [LLM agent workflows](#) on LangChain and FastAPI that automate decision steps people handled by hand in 3 internal processes; the hardest, [request triage](#), has to route messy, inconsistent requests without relying on a fixed input format.
- Shipped [4 LLM services](#) on shared components; defined the API contracts for 3 and built their React/Next.js client.

Computer Vision and Machine Learning Engineer Intern Gandhinagar, India
SPAN Inspection Systems, automated inspection of pharmaceutical products and packaging Jan 2025 – May 2025

- Productionized an [EfficientNet-B0 defect detector](#) trained on 50,000+ images that catches broken tablets, missing pills, and damaged packaging, reaching [98.2% production accuracy](#) and taking over 40.9% of manual inspection across 3 lines.
- Cut false positives [32.2%](#) by teaching the model to ignore glare: reflections on foil and blister packs looked like real damage and were being flagged as defects, so added training augmentations that [simulate that glare](#).
- Converted the model to [TensorRT FP16](#) for NVIDIA Jetson: 42.8% lower latency, 27.3% more throughput, within 0.3 pts of FP32.
- Automated [retraining and rollout](#) with S3, Docker, and CI/CD, ending the 6 hours a week operators spent on manual updates.

Full-Stack Software Engineer Intern Remote
Octopus Technologies, business web portal on React and Node.js microservices May 2024 – Jul 2024

- Cut page load [38.3%](#) for 5,000+ monthly users: the portal shipped as a [single JavaScript bundle](#), so every page loaded the code for all 12 routes even when only one was opened; split it into route-level, lazy-loaded chunks.
- Rewrote slow queries behind [compound indexes and a Redis cache](#), raising API throughput 35.6%.
- Added JWT auth for 8 user roles and moved deploys to GitHub Actions and AWS ECS (about 25 minutes down to 10).

Backend and Quantitative Systems Engineer Intern Ahmedabad, India
WealthAffinity Softech, trading platform software May 2023 – Nov 2023

- Rebuilt the C++ order-matching core on [lock-free queues and async I/O](#): 2.3 ms round trips, 1.4x capacity at 10,000+ orders.
- Some crashes in the new core only showed up sometimes, so a test could pass several times in a row and then fail; tracked them down, fixing 3 memory leaks with [Valgrind](#) along the way.
- Served an [XGBoost risk model](#) over REST that checks orders before they execute; it blocked [23% of erroneous trades](#), an estimated \$50K a month in avoided write-offs.
- Wrote 150+ unit tests (6 modules at 95% coverage) and cut CPU use 28% after JMeter load testing.

PROJECTS

Computer-Use Automation System | *TypeScript, Node.js, Playwright, Gemini API, Zod, Vitest*

- Built a browser agent for legacy web apps with no test IDs that records each successful run as a [typed, versioned Zod capability](#) (a replayable script of the steps), so repeat tasks run deterministically with [zero model calls](#).
- Human oversight for runs on a mock bank and a credit-union console: checkpoints, a pause/takeover/resume path, and a [circuit breaker](#) that stops for approval when the agent's confidence drops.

SentiMind (Team Project, HackPrinceton Fall 2025) | *Python, MCP, Gemini 2.0 Flash, WebSocket*

- Co-built a [Reddit MCP server](#) (PRAW) and an agent that orchestrates its tools with Gemini 2.0 Flash to detect sentiment, toxicity, and manipulation patterns in live threads, [streaming results over WebSocket](#).
- Co-built a [Chrome extension](#) that shows the same analysis inside WhatsApp Web, Slack, Discord, and Teams.

FlightRoute: Aerial Route Planning Engine | *Python, Flask, MongoDB*

- Finds [multi-leg flight routes](#) between any two airports under a 6,000 km range limit using a [greedy layover search](#) (IJIMS paper).
- Computes [great-circle distances](#) with the spherical law of cosines and prunes the global airport set to a source-destination bounding box, ranking candidate stops by how far they detour from the direct path.

TECHNICAL SKILLS

Languages: Python, TypeScript, JavaScript, C++, SQL, Bash

AI and Machine Learning: PyTorch, LangChain, LLM Agents, MCP, Gemini API, TensorRT, EfficientNet, XGBoost, OpenCV

Backend and Data: FastAPI, Flask, Node.js, Express.js, React, Next.js, REST, WebSocket, PostgreSQL, MongoDB, Redis

Cloud and DevOps: AWS (S3, EC2, ECS), Docker, GitHub Actions, CI/CD, NVIDIA Jetson, Linux, Valgrind

Certifications: Machine Learning Specialization (Stanford); AWS Cloud Technical Essentials; Advanced C Programming

PUBLICATIONS

[An Integrated Framework for Real-Time Object Detection and Speech-Driven Interaction](#)

ICDAM 2025, Springer

[A Pioneering and Robust Technique for Optimizing Aerial Route Planning](#)

IJIMS 2026, Inderscience

[Cipherring the Modern World: Analysis of DES, AES, RSA and DHKE](#)

INDIACom 2024, IEEE