

# PIYUSH SATTI

+1 (514) 451-1479 • Montreal, Quebec, Canada • [piyushsatti@gmail.com](mailto:piyushsatti@gmail.com)  
[linkedin.com/in/piyush-satti](https://linkedin.com/in/piyush-satti) • [github.com/piyushsatti](https://github.com/piyushsatti) • [piyushsatti.github.io](https://piyushsatti.github.io)

## SKILLS

<b>Languages</b>	Python, JavaScript/TypeScript, Java, Go, SQL, C/C++, Ruby, HTML/CSS, MATLAB
<b>Frameworks</b>	FastAPI, Flask, Express/Node, Spring Boot, Kafka/RabbitMQ, Nginx
<b>Agents</b>	LLM APIs, RAG, Prompting, LangChain, MCP, Tavily
<b>Data</b>	PostgreSQL, Redis, MongoDB, Vector DB, SQLite
<b>Cloud</b>	AWS, Docker, Azure, DigitalOcean, Supabase, Vercel
<b>Tools</b>	Git/GitHub, Pytest, JUnit, Postman, Selenium, Jira/Trello, Figma
<b>Practices</b>	TDD, CI/CD (GitHub Actions), Event-driven, DDD

## EDUCATION

<b>Concordia University</b> (GPA: 3.62/4.3) <i>Master of Science, Applied Computer Science</i>	Montreal, Quebec, Canada <i>Sept. 2023 – Aug. 2025</i>
---	---

<b>Thapar Institute of Engineering and Technology</b> (GPA: 8.96/10) <i>Bachelor of Engineering, Electronics and Computer Engineering</i>	Patiala, Punjab, India <i>Jun. 2017 – Jun. 2021</i>
--	--

## EXPERIENCE

<b>Teaching Assistant: OOP</b> <i>Concordia University</i>	Montreal, Quebec, Canada <i>Jan. 2025 – Apr. 2025</i>
---	--

- Programmer on Duty (Java) for Object-Oriented Programming II with more than 400 students. Guided students to understand OOP concepts and their implementation. Helped with course projects and program debugging.
- Conducted classes for roughly 20 students each week, and a revision lecture for 40 students covering the following topics: File I/O, Polymorphism, Recursion, Exception Handling, Abstract Classes and Interfaces, Inheritance.

<b>Research Assistant: Algorithms</b> <i>Thapar Institute of Engineering and Technology</i>	Patiala, Punjab, India <i>Aug. 2019 – Apr. 2023</i>
--	--

- Researched algorithm-based techniques for restoring corrupted images. Published 4 academic papers and achieved state-of-the-art performance. Implemented 40 cutting edge research papers in the process.
- Achieved exceptional improvement in the signal-to-noise ratio, resulting in the several papers being published in peer-reviewed journals, including the esteemed IEEE:SPL (*70+ citations*). DOI: **10.1109/LSP.2020.3016868**.

<b>Freelance Software Engineer: Hacking Forums</b> <i>Independent Contractor</i>	Remote <i>Jun. 2023 – Present</i>
---	--------------------------------------

- Delivered rapid prototypes for early-stage clients, aligning scope and timelines while iterating on requirements and UI feedback.
- Built and deployed a lightweight analytics dashboard using React and FastAPI, integrating third-party APIs and automated reporting.

## PROJECTS

<b>GameOps Suite: RESTful API + ETL Pipeline</b> <i>Python, FastAPI, Discord API, Redis, MongoDB, Docker</i>	<a href="#">piyushsatti/nonagon</a> <i>Mar. 2025 – Present</i>
---	---

- Designed and now maintaining an end-to-end telemetry and sign-up platform for *500 Indian Board Gamers* - dashboard for serving player statistics, logic for managing game sign-ups, and Discord interface for in-text AI support.
- Developed an ETL pipeline to convert discord text-based data into MongoDB documents for persistence, inference, and downstream business logic. The pipeline ingests data via Discord's event-driven APIs. Serves it via the API.
- Created RESTful FastAPI endpoints with authentication and role-based authorization for CRUD operations. A dashboard for serving player statistics and displaying real-time game announcements.
- Deployed using containers (API + ETL) over DigitalOcean droplets. The project uses the principles of domain-driven design to keep API logic and the ETL pipeline maintainable. Delivered 200 ms end-to-end latency for typical actions.

<b>Risk-Emulated: Spring Boot + GraphQL</b> <i>Java, PostgreSQL, DDD, GoF Patterns, Docker</i>	<a href="#">piyushsatti/risk-emulated</a> <i>Aug. 2024 – Present</i>
---	---

- Built a Spring Boot **GraphQL** backend for a playable *Risk* clone with clean Domain-Driven Design: Entities (Game, Player, Territory, Continent, Card, MoveLog), GraphQL types (input/output), **resolvers** for map & turn actions, and Services enforcing rules.
- Designed a graph-based map builder/validator that guarantees connected, legal maps and continent bonuses; exposed a typed **schema** with **queries/mutations** (*createGame, reinforce, attack, fortify, gameStatus*).
- Implemented a phase/state engine and rules using GoF patterns (*State* for phases, *Strategy* for combat/reinforcement policies), enabling pluggable variants and targeted test scenarios.
- Dockerized app; GitHub Actions for build/test/coverage; style gates (Checkstyle/Spotless); structured audit logging (MoveLog) for reproducible turns and troubleshooting.