

Jose David Ibanez

jdavidip2022@gmail.com | +1 (548) 390-4453 | github.com/jdavidIP | [linkedin/ibanez](https://linkedin.com/in/ibanez) | Mississauga, ON

PROFESSIONAL SUMMARY

Full-stack software developer with a Computer Science degree (major in Cybersecurity), combining hands-on experience across React, Node.js, Python, C#, and PHP with a strong foundation in testing, databases, and cloud deployment. Comfortable owning features end-to-end, adapting quickly to new stacks and codebases, and using AI tools like Claude and GitHub Copilot daily as part of the development workflow.

PROFESSIONAL EXPERIENCE

Junior Software Developer — Kraken Sense

May 2026 – July 2026

Contract, Full Time | Oakville, ON, Canada

- Took ownership of developing new features and fixing bugs across 3 existing internal full-stack applications built with React (JavaScript/TypeScript), Node.js, FastAPI, and PostgreSQL, supporting internal operational workflows.
- Built and enhanced backend APIs with FastAPI, implementing business logic, database interactions, and integrations across application services.
- Built responsive frontend features in React alongside backend APIs, collaborating across the full stack on the same builds.
- Collaborated closely with engineers and stakeholders in an iterative, Agile development cycle — gathering feedback, prioritizing requirements, and refining features through short feedback loops.
- Leveraged AI-assisted development tools such as Claude and GitHub Copilot to accelerate testing, debugging, and development workflows, including MCP-based integrations connecting Claude Code to GitHub.
- Performed QA testing and debugged issues across frontend, backend, and database systems, identifying and resolving problems prior to release.
- Managed PostgreSQL databases and supported deployments on Google Cloud Platform using Cloud Run and Cloud SQL, contributing to CI/CD pipelines.
- Worked with Redis as part of application infrastructure to support caching and performance, and used Bitbucket for version control.

Junior Software Developer Intern — Agro-Costa SAS

May 2024 – December 2024

Internship | Barranquilla, Colombia

- Took ownership of features across web, backend, automation, and database systems, delivering solutions from design through deployment in a production environment.
- Practiced test-driven development to validate business logic before implementation, improving code reliability and reducing regressions in production workflows.
- Implemented role-based access control (RBAC) to manage multiple user types, improving administrative workflows and increasing platform engagement by 30%.
- Developed and maintained backend services using PHP, ASP.NET, and Node.js, implementing business logic, API integrations, and database-driven workflows — including learning PHP on the job within an existing production codebase, under the mentorship of a senior developer.
- Identified, investigated, and resolved production issues across frontend, backend, and data systems, tracing bugs to their root cause across the stack.
- Organized and tracked work using Azure DevOps throughout the development lifecycle.
- Built a C# desktop automation tool to process inventory and pricing updates via CSV parsing and API integration, reducing manual effort by 90%.
- Designed and optimized SQL Server and MySQL queries and resolved production issues across frontend, backend, and data systems.

PROJECTS

Marquee — Distributed Social Movie-Discovery App

July 2026 – Present

- Building a full-stack app around a synchronous "clap to unlock" mechanic: an ASP.NET Core (.NET 9) API and background worker, an Angular 20 SPA (standalone components/signals), PostgreSQL, Redis, and RabbitMQ (MassTransit) for event-driven fan-out; github.com/jdavidIP/Marquee.

- Implemented atomic clap counting in Redis via Lua scripts, an outbox pattern for exactly-once event delivery between the API and worker, and Quartz.NET for time-triggered scheduling.
- Built real-time updates over SignalR with batched (not per-clap) broadcasts, and a permission-based JWT authorization model with Redis-backed block-list checks.
- Added observability: OpenTelemetry distributed tracing across API → Redis → RabbitMQ → worker, structured logging with Serilog, and an admin dashboard for live queue depth and connection metrics.
- Verified correctness under load with Testcontainers-based integration tests against real Postgres/Redis and k6 load tests confirming no lost claps or duplicate opens under concurrent traffic.

Appllyt — Self-Hosted Job Application Tracker

June 2026 – August 2026

- Built a local-first, self-hosted job application tracker with a Chrome extension for automatic capture and AI-powered resume tailoring; github.com/jdavidIP/Appllyt.
- Built the backend with Node.js, Fastify, and better-sqlite3 (TypeScript), and the dashboard with React and Vite (TypeScript).
- Built a Chrome Manifest V3 extension (TypeScript content scripts) that auto-detects submitted job applications on Indeed, LinkedIn, and Glassdoor without filling out or submitting anything on the user's behalf.
- Integrated direct REST calls to the Anthropic and OpenAI APIs for AI-based resume tailoring, including per-run token/cost tracking and pre-generate cost estimates.
- Designed a REST API with CRUD endpoints, CSV export, and lifecycle management features (bulk status updates, stale-application detection, response-rate stats).
- Wrote automated tests with node:test (backend) and Vitest (dashboard), and packaged the app with Docker Compose for one-command setup.

Simple Rentals – Full Stack Web App

January 2025 – August 2025

- Built with a small team; personally responsible for the backend, ML recommendation system, and testing/CI-CD infrastructure.
- Built the backend using Django REST Framework and PostgreSQL, and developed a responsive frontend with React.
- Built a personalized listing recommendation system using scikit-learn, based on user interaction data.
- Built the testing and CI/CD infrastructure: unit and end-to-end tests with Cypress, unit tests with Vitest, and automated pipelines with GitHub Actions running on every push and pull request.
- Applied test-driven development, writing tests ahead of implementation to catch regressions early.
- Managed project tasks using Azure DevOps, and containerized deployment using Docker.

Real-Time Object Detection & Tracking System

2023 – 2024

- Built a computer vision application in Python using OpenCV and YOLO to detect and track everyday objects in real time from live video input.
- Implemented the detection pipeline end-to-end — video capture, preprocessing, inference, and bounding-box/tracking overlay — and tuned model parameters to improve detection accuracy and frame-rate performance.

TECHNICAL SKILLS

Programming Languages: Python, JavaScript, TypeScript, C#, C, C++, PHP, SQL, HTML5, CSS3

Web & Frontend: React, Angular, Vue.js, Vite, Tailwind CSS, responsive UI development

Backend & APIs: Node.js, Fastify, FastAPI, ASP.NET Core, Django, ASP.NET, PHP, REST API design & integration, SignalR

Database Technologies: PostgreSQL, MySQL, Microsoft SQL Server, MongoDB, SQLite (incl. better-sqlite3), Redis

Messaging & Distributed Systems: RabbitMQ, MassTransit, event-driven architecture, outbox pattern, Quartz.NET scheduling

Observability: OpenTelemetry (distributed tracing), Serilog, Jaeger

Browser Extensions: Chrome Manifest V3, content scripts, background service workers

Computer Vision & AI: OpenCV, YOLO, object detection & tracking

Testing & QA: Test-driven development (TDD), Cypress (unit, integration & end-to-end testing), Vitest, node:test, xUnit, Testcontainers, k6 load testing, manual QA testing, bug identification & tracking, Postman (API testing, coursework)

Cloud & DevOps: Docker, Docker Compose, Google Cloud Platform (Cloud Run, Cloud SQL), Microsoft Azure (coursework level), GitHub Actions, Azure DevOps, Git, Bitbucket, CI/CD

In Progress: Studying for AWS Certified Cloud Practitioner

AI-Assisted Development: Claude, GitHub Copilot — used throughout the development lifecycle; MCP integrations (Claude Code to GitHub); direct Anthropic/OpenAI API integration for AI-powered features; LangChain fundamentals (self-directed coursework)

Operating Systems: Windows, Linux (Ubuntu, Kali, Debian, CAINE)

Security Foundations: Access control & identity management (RBAC), secure coding fundamentals, applied cryptography, network security fundamentals, vulnerability & risk concepts (Cybersecurity coursework)

Practices: Agile/iterative development, pair collaboration, full-stack ownership, code review

EDUCATION

Bachelor of Computer Science – Honours (Co-op), With Distinction

2021 – 2025

Major in Cybersecurity — Conestoga College, Waterloo

Relevant Coursework: Introduction to Software Security · Network Security · Information Security Management · Rootkits and Hacking · Privacy in Computing · Forensics and Cyber Incident Management · Secure Enterprise Architecture · Applied Cryptography · Software Design Techniques · Data Structures and Algorithms · Software Development Life Cycle · Software Quality · Object Oriented Programming · Database Systems