

Gracian Anton

gracian.anton@gmail.com

(613) 400-6844

EDUCATION: Carleton University – Bachelor of Computer Science Honours, Stream: Artificial Intelligence & Machine Learning (currently completing)

TECHNICAL SKILLS:

Frontend: HTML, CSS, Bootstrap, JavaScript, React.js

Backend: Laravel, SQL, PHP, Java, Python

AI/LLM: RAG systems, Agentic AI, Vector DBs, Embeddings, Semantic Search, API Tool Integration

ML: Scikit-Learn, Pandas, NumPy

Tools: Git, REST APIs, Linux, SSH/SFTP

GitHub: github.com/graciananton

Machine Learning Content Channel:

- Conducted **data analysis on large CSV datasets** using **Python, Pandas, NumPy, and Scikit-learn** to explore patterns in datasets and prepare them for ML models.
- Built and trained machine learning models using **cleaned and preprocessed datasets** generated through the data analysis pipeline (in progress).
- Developed **solid writing skills** by preparing lecture notes to **explain machine learning concepts in an easy-to-learn format**.
- Produced educational content notes explaining **machine learning pipelines, dataset preparation, and model development**.
- [Machine Learning Channel](#)

PROJECTS:

AI-Driven Hydrological Monitoring and Forecast Engine (Dec 2025 - present):

Backend:

- Implemented a **Laravel MVC backend** with **authentication, routes, controllers, and views**.
- Synchronized API-fetched JSON data into a relational **MySQL database**.

Frontend:

- Built a user-facing **React frontend** styled with **Tailwind CSS**, using backend APIs to display responsive dashboards and forecast views.

RAG Chatbot System:

- Developed a **RAG-based chatbot** using by uploading files into a vector store (chunking, vectorizing, and indexing those files) and creating a filter for a similarity search against the vector store.

Agentic Workflows:

- Designed and implemented an **agent-driven user authentication workflow** that replaces conventional signup/login forms with a conversational onboarding experience.

Machine Learning Model:

- Designed an ML model to **predict future water** levels based on projected wind, rain, and temperature.
- Used Python, Pandas, NumPy, and scikit-learn to build the model based on information **collected via a REST API and stored in a MySQL database**

API-Integration:

- **Automated API data collection** via cron-scheduled Laravel Artisan commands, **stored in a relational SQL database**.
- Developed **RESTful API** endpoints to expose forecasting results, station data, and historical trends in **JSON formats**, with enforced request validation and authorization.
- [Github](#)

AI Chatbot (Sept. 2025 – Dec 2025)

- Users complete a **registration process** that includes sign-up, account verification, and re login. Session variables track authenticated users, allowing secure movement between pages after logging in.
- Built using **object-oriented design principles**, including polymorphism, abstraction, inheritance, and encapsulation.
- Stored chats and messages in a **relational MySQL database**.
- Used JavaScript and AJAX to display new messages and chats.
- Used Langchain to **generate descriptions and responses to user messages**.
- Built with a **responsive frontend** using Bootstrap and custom CSS.
- [Github](#)

Family Tree Management System (Jan. 2025 – July 2025)

- Create a secure admin/user interface for individuals to **add, edit, update, and delete family members** and their relationships.
- Used **object-oriented principles** such as polymorphism, inheritance, and encapsulation.
- Built a persons and relations table to link persons with their relatives.
- Implemented a **dTree** JavaScript module to **display persons in a seamless and navigational way**.
- Allowed admins to **update settings** and global environment variables for the entire website.
- Created a chatbot using information from a MySQL database with LangChain.
- Designed a **responsive frontend** using Bootstrap and custom CSS.
- Built a **secure contact form** with reCAPTCHA.
- [Github](#)

Custom CMS Website for a Community Group (Apr. 2024 – Nov. 2024)

- Built a **custom content management system (CMS)** enabling administrators to manage pages, videos, announcements, and site content.
- Implemented a **secure backend interface** for content editing, publishing, and contact form handling.
- Integrated the TinyMCE rich-text editor to allow non-technical users to create formatted content.
- Developed a **custom template engine** to process reusable layouts, custom tags, and page structures.
- Implemented **multi-language support**, an integrated **Google Calendar**, and a **custom-built search engine**.
- Created custom frontend sections of the website including **sliding images, content boxes, and video listings**.
- Designed a **responsive frontend** using Bootstrap and custom CSS.
- [Github](#)

