

Aviral Yadav

Phone: 7007934595 | aviy2402@gmail.com | [LinkedIn](#) | [Github](#)

Education

Mount Litera Zee School, Gorakhpur | 10th – 77% | 12th – 89%

2019 - 2021

Vellore Institute of Technology, Bhopal – *B Tech in Computer Science and Engineering (Cyber Security and Digital Forensics)* |
CGPA – 7.84

August 2022 – June 2026

Skills

Technologies & Tools: Excel (Advanced Formulas, Lookup Functions, Pivot Table), AWS, Git, Blender

Programming Languages: Java, Python (Pandas, NumPy, Matplotlib, Seaborn), React

Experience

QuantAI, Auckland, New Zealand - *Data Scientist Intern*

May 2025 – July 2025

Scrapeazon [Python | Requests | BeautifulSoup | Flask | Flask-CORS]

May 2025

- Developed a Python web scraper to automate data collection from Amazon, successfully extracting up to **100** product listings from a single search query.
- Engineered the tool to navigate and parse up to **20** consecutive search result pages, significantly expanding the scope of data aggregation.
- Utilized libraries like **Requests** and **BeautifulSoup** to transform unstructured HTML into organized, actionable data for market analysis.

Projects

Car Finder [Python | Requests | BeautifulSoup | Flask | Flask-CORS | JSON]

June 2025

- Developed a Python scraper to aggregate data from **2** car dealership websites (**Andrew Simms** and **NZ Cheap Cars**), delivering up to **100** combined results per search.
- Aggregated a comprehensive dataset of up to **100** vehicles per query by parsing **8** key attributes—including price and odometer reading—from as many as **6** pages of search results.
- Ensured high reliability by implementing a **3-attempt** retry mechanism with exponential backoff and rotating through **3** unique user-agents to prevent blocking.

Word Cloud Generator - [Flask | Python | WordCloud | PyMuPDF | Wikipedia-API]

July 2025

- Developed a full-stack application using **Flask** that allows users to generate word clouds from **2** distinct sources: direct file uploads or by searching for any Wikipedia article. The application can parse **3** different document types (.txt, .docx, and .pdf) to extract text for analysis.
- Utilized **5** core Python libraries, including **Flask**, **WordCloud**, **PyMuPDF**, **python-docx**, and **Wikipedia-API**. The accuracy of the visualization is enhanced by automatically filtering out a predefined set of common STOPWORDS, which ensures the final image highlights the most significant and relevant terms from the source text.

Academic Research Assistant – [Python | Flask | AI & Machine Learning]

July 2025

- Developed a user-friendly, responsive web application using **Flask** that allows users to upload PDF files and engage in a real-time conversational chat about the documents.
- Constructed a RAG chain using the **LangChain** framework. When a user asks a question, the system first retrieves the most relevant document chunks from the **FAISS** index. This context is then fed to a Large Language Model to generate a precise and well-informed answer.
- Integrated the **Llama3-8B** model running on the high-speed **Groq LPU™ Inference Engine**. This ensures that answers are not only accurate but also generated with extremely low latency, providing a seamless user experience.

Certifications

- Bits and Bytes of Computer Networking** – Coursera
- Cyber Physical System** – NPTEL
- VLSI Design** – Maven Silicon