Luis Fernando Villalon

(760) 234-2481| lvillalon1179@sdsu.edu | LinkedIn | GitHub | Web Portfolio

EDUCATION

San Diego State University **Graduation**: May 2027

Bachelor of Science: *Computer Science*

Relevant Coursework: Object-Oriented Programming, Data Structures, Algorithms, Software Systems, Operating Systems, AI

Academic Clubs: ColorStack, STEM Advantage, Society of Hispanic Professional Engineers, Association of Computing Machinery

TECHNICAL SKILLS

Programming Languages: *Python, C++, Java, JavaScript, TypeScript*
AI & LLM Integration*: RAG Systems, LLM APIs, LangGraph*
Backend & APIs: *Node.js, FastAPI, Pydantic, Restful APIs*
Databases*: MongoDB, Firebase, PostgreSQL, Supabase*
Frontend: *HTML, CSS, React.js, React Native, Next.js, Tailwind CSS*

PROFESSIONAL EXPERIENCE

Computer Science Intern

Second Course - San Diego, CA *(Hybrid)* **Nov. 2025 - Jan 2026**

● Refactored and documented a rapidly AI-generated React Native codebase (TypeScript, CSS, Firebase), reducing technical debt by 30%+ and improving maintainability to accelerate the app's launch timeline.

● Enhanced app stability and user flow by debugging core components and optimizing front-end logic, contributing to a smoother MVP experience aimed at reducing food insecurity for 150+ college students.

RESEARCH

Undergraduate Researcher - Adversarial Attack & Defense in Deep Learning

AI4Business Lab - San Diego, CA *(Remote)* **Jan. 2026 - May 2026**

● Conduct literature-driven research on AI agents, focusing on trust, safety, and adversarial robustness in deep learning systems.

● Contribute to an undergraduate research paper proposing a three-layer trust-centered AI safety framework integrating psychological trust, technical reliability, and security threats.

BOOTCAMPS

NxP Semiconductors: Agentic AI in the Cloud

ACEE (Agentic Abstraction for Chip Engineering | GitHub Repo |- San Diego, CA *(Hybrid)* **Feb. - May 2026**

● Designed and implemented a multi-agent AI pipeline that converts SystemVerilog RTL into architecture diagrams using specialized LLM agents (Architect, Auditor, Stylist, and DOT Compiler) orchestrated through LangGraph workflows.

● Built an automated validation and self-correction framework with audit-driven feedback loops, structured JSON schemas, and retry mechanisms that improved reliability and consistency of LLM-generated hardware representations.

● Developed a full-stack application using FastAPI, Next.js, and Graphviz integrations to generate and serve SVG architecture diagrams, supporting iterative user-guided refinement and multiple LLM backends (AWS Bedrock, OpenAI, Gemini).

PROJECTS

NutriNav AI | GitHub Repo | *Academic AI/LLM Project* **Mar. 2025 - Present**

● Designed and built an autonomous multi-agent workflow that coordinates deterministic and LLM-based agents to generate nutritionally compliant meal plans from user biometrics and dietary preferences.

●Engineered a self-correcting feedback architecture where validation agents evaluate outputs and automatically guide subsequent LLM generations until compliance requirements are satisfied.

kanso | Front-End GitHub Repo | Back-End GitHub Repo | AI GitHub Repo | *Personal Full-stack Project* **Jan. 2026 - Present**

● Built a frontend with Next.js, React.js, TypeScript, and Tailwind CSS that connects to a FastAPI backend with PostgreSQL to power an organizational web app, implementing RESTful CRUD endpoints to track and visualize user data with real-time statistics and intuitive UI.

● Deployed a FastAPI AI microservice that converts structured notes into curated learning resources using LLM-generated search plans, web search, paywall and dead-link filtering, credibility scoring, and deterministic ranking, with support for OpenAI and Gemini models.

LLM-Powered Bible Tutor | GitHub Repo | *(Personal AI/LLM Project)* **Nov. 2025 - Present**

● Designed a hallucination-resistant RAG system in Python using OpenAI's GPT-4o-mini LLM API to answer Bible questions exclusively from semantically retrieved verses, enforcing citation of 5 sources and explicit *"I don't know"* responses when no grounded answer exists.

● Built a Bible-only vector database by embedding structured CSV verse data and indexing it in ChromaDB, enabling efficient semantic search through vector similarity matching between user queries and source text.

● Orchestrated a modular LangChain-based workflow for embedding generation, retrieval, and prompt-constrained reasoning.