NISHIT D B

nishit.bilimagga@gmail.com
github.com/Crossbow2560 | linkedin.com/in/nishit-db | nishit-db.com | Bangalore, IN

B.Tech CS undergraduate building full-stack platforms, embedded/IoT systems, and GenAI/agentic applications.

Education

B.Tech Computer Science
PES University Bangalore
CGPA: 8.91 | 2024 – 2028

Coursework: Operating Systems, Computer Networks, Microprocessor and Computer Architecture, Computer Organization

Skills

Languages: Python, C, C++, JavaScript, TypeScript
Web Frameworks: Node.js, Fastify, Next.js, React, TanStack Query, Recharts, Express, HTML5, CSS
Data Structures & Databases: DSA, SQL, PostgreSQL, MongoDB, Supabase
DevOps/Infra: Git, Docker, Linux, Cloudflare, MinIO, MQTT
AI/ML: GenAI, Prompt Engineering, Local LLM inference (llama.cpp), faster-whisper, pydantic
Embedded: Raspberry Pi, Arduino, ESP32

Experience

Research Intern – PES Innovation Lab (Akamai-funded) *Jun – Jul 2026*
Project: Grimoire – LLM-based Knowledge Wiki

- Built an LLM-based knowledge wiki (Grimoire) that converts scattered Jira issues, Confluence docs, PDFs, and markdown files into a persistent, structured, interlinked knowledge base, with automated Atlassian ingestion and tag-based indexing
- Implemented rolling summarization and multimodal image interpretation for large documents, and built an interactive chatbot that answers relational queries by cross-referencing wiki files – version-controlled with Git throughout

Club Head – The Alcoding Club

- Hosted club events: CTFs, hackathons, and competitive programming events
- Attended multiple hackathons, CTFs, and competitive programming challenges

IEEE RAS Member / Remote Robotics Lab – Center for Robotics, Automation and Intelligent Systems (cRAIS)

- Contributed to a remote robotics initiative at cRAIS by building a line-follower bot and a custom web dashboard, enabling users to write, upload, and run code on the physical bot without direct lab access.

Projects

DBase – Self-Hosted Cloud File Storage Platform *Solo project*
Tech: *Node.js, TypeScript, Fastify, PostgreSQL (Supabase), Next.js, TanStack Query, MinIO, Docker*

- Built a full-stack, self-hosted, multi-user cloud file storage platform (Dropbox/Nextcloud-style) with hierarchical folders, direct and public-link sharing, per-user storage quotas, and an admin role for system-wide file and usage visibility
- Modeled a relational schema in PostgreSQL with polymorphic file/folder permissions and cycle-safe concurrent folder moves via Postgres advisory locks; built a Fastify + TypeScript REST API with JSON-schema validation and asynchronous, retryable job queues, and a Next.js/TypeScript client with TanStack Query, deployed via Docker Compose

Agent Security Testing Framework – PES Innovation Lab
Tech: *Python, OpenAI/Anthropic/Gemini/Ollama/llama.cpp adapters, pydantic, JSONL event logging*

- Built a Python test-automation framework for evaluating LLM-backed agents against memory poisoning: a hot-swappable LLM adapter layer, risk-tiered tool registry, full observability, and an external memory store
- Designed and ran a 200-case memory-poisoning benchmark across 5 local LLMs (7%–93% per-domain attack success rates), plus a prompt-injection test harness sweeping authority-override, role-spoofing, and jailbreak variants

Agritech Platform – Farm Management & GIS Dashboard *Team project, in production*
Tech: *Next.js, TypeScript, Supabase, MinIO, Recharts, custom GIS microservices*

- Contributing the majority of the frontend and backend implementation for a full-stack farm-management platform, as part of a team building the product, now in production for a Malaysia-based client
- Built dashboard modules (farm overview, crop health, NDVI/NDRE vegetation indices across 14,000+ tracked plants) with Next.js/TypeScript/Recharts, and backend microservices (Tiler, Modifier) serving geospatial map tiles from MinIO imagery, with Supabase-backed auth and data persistence

AI Home Assistant
Tech: *Local LLM (Qwen3.5 0.8B via local Llama server), RealtimeSTT (faster-whisper), Python, MQTT (Mosquitto)*

- Built a fully local voice assistant pipeline: real-time speech-to-text (RealtimeSTT) feeding a locally hosted LLM, with no cloud dependency, and a tool-calling interface so the LLM emits structured function calls for device actions from voice commands
- Implemented an MQTT (Mosquitto) control layer to execute tool calls on ESP32-based custom smart switches, closing the loop from spoken command to physical actuation entirely on local hardware

Remotely Programmable Line Follower – Center for Robotics and Intelligent Systems
Tech: *ESP32, Raspberry Pi, Sensors & Actuators, Next.js, Python, Supabase, PostgreSQL*

- Built a line-following robot that supports remote code upload and execution over a wireless interface, with a custom Next.js web dashboard for programming and control
- Implemented Python backend services for device communication and code deployment, and integrated ESP32 sensor-actuator control with remote firmware updates