

Jyothi Basu Chodavarapu

CONTACT INFO:

Email:

contact@mail.jyothibasus-a.dev

Website:

jyothibasus-a.dev

GitHub:

github.com/jyothi-basu

LinkedIn:

linkedin.com/in/jyothi-basu-chodavarapu

CAREER OBJECTIVE:

Python Backend Developer with hands-on experience building production-oriented backend applications using FastAPI, Flask, REST APIs, pytest, Docker, MongoDB, Redis, and secure authentication. Experienced in integrating LLM-powered features using LangChain, Retrieval-Augmented Generation (RAG), tool calling, and Model Context Protocol (MCP) while implementing business-rule guardrails and AI workflows. Seeking a remote Junior Python Backend Developer role focused on building scalable backend services with modern AI integration.

TECHNICAL SKILLS:

Languages: Python

Backend Frameworks: FastAPI, Flask, REST APIs, Uvicorn, Gunicorn

Databases: PostgreSQL (Learning), MySQL, SQLite, MongoDB, Redis

Testing: pytest

Authentication & Security: JWT, bcrypt, Argon2, Input Validation

DevOps & Tools: Docker, Docker Compose, Git, GitHub, Linux (WSL), Render, Cloudflare R2, Resend

AI Integration: LangChain, RAG, ChromaDB, Embeddings, MCP, Tool Calling

Domain: Custom domains and subdomains, DNS management, Custom Email,

Internships:

Software Engineering Trainee Intern FamilyShell LLP | May 2026 - Present

Worked on a production-oriented family management platform built with FastAPI, MongoDB, Redis, Docker, Cloudflare R2, and Flutter.

Key Contributions

- Investigated and resolved 27 production bugs by tracing request flows across FastAPI services, MongoDB, Redis, and Flutter clients.
- Implemented backend improvements for Health Records, Reminders, Notifications, Authentication, Background Scheduler, and Email Reminder workflows.

- Debugged and fixed API validation, timezone handling, multi-user reminder scheduling, and notification delivery issues.
- Worked with JWT authentication, Redis caching, Cloudflare R2 object storage, WebSocket notifications, and ZeptoMail integration.
- Performed regression testing and collaborated with developers to verify production-ready behavior before deployment.

PROJECTS:

Inventory Management API:

- Built and deployed a Flask REST API supporting authenticated inventory management with user accounts and CRUD operations backed by MySQL.
- Implemented JWT-based authentication, bcrypt password hashing, protected routes, request validation, and standardized HTTP error responses.
- Structured the application using a layered architecture (routes, services, storage, authentication, configuration) to improve maintainability.
- Deployed the backend on Render with environment-based configuration and a hosted MySQL database on alwaysdata.

GitHub:

github.com/jyothi-basu/inventory-management-api

Live API URL:

inventory-management-api-4ikv.onrender.com

Investment Portfolio Tracker:

- Developed a production-style **FastAPI backend** using a layered architecture (Routes → Services → Repository) with JWT authentication, multi-Demat account management, BUY/SELL transactions, holdings, portfolio summaries, and SQLite persistence.
- Built a Retrieval-Augmented Generation (RAG) pipeline using ChromaDB with configurable cloud or local embeddings, enabling users to upload financial documents (PDF, DOCX, TXT, MD) and receive source-grounded answers with citations.
- Implemented a provider-agnostic LangGraph + LangChain AI assistant that uses tool calling to combine portfolio data, uploaded document evidence, and application knowledge while keeping user and chat identity isolated through trusted backend context.
- Integrated Model Context Protocol (MCP) with Streamable HTTP, PAT authentication, and chat-scoped document search, and added an automated **pytest** test suite for authentication, portfolio, document, chat, and MCP workflows.

GitHub:

github.com/jyothi-basu/investment-portfolio-tracker

Access Tracker — Full Stack Accessibility Bug Reporting Platform

- Built **Access Tracker**, with FastAPI, MongoDB, Redis, JWT, Resend email service, Flutter. Access Tracker is a full-stack accessibility bug reporting platform that enables screen reader users to report, verify, and track accessibility issues across mobile and web applications.
- Designed and developed a **FastAPI** backend using a feature-based architecture with modules for authentication, applications, bug reports, and community verifications.
- Implemented **JWT authentication** with access and refresh tokens, secure password hashing using **Argon2**, email OTP verification, and **Redis** for temporary OTP storage and session management.
- Built a **Flutter Web** frontend with **Riverpod** for state management, **GoRouter** for authenticated navigation, and **Dio** for API communication with automatic token refresh.
- Developed complete CRUD workflows for bug reporting, community verification, application search, user profiles, and ownership-based editing and deletion.
- Implemented **Role-Based Access Control (RBAC)**, allowing administrators to moderate bug reports while restricting verification edits and deletions to their creators.
- Integrated **MongoDB** with PyMongo, optimized collections with indexes, and followed repository and service layer patterns for maintainable backend code.

GitHub:

github.com/jyothi-basu/access-tracker

VisionAid Python Programming Repository

- Maintained a public repository containing 50+ Python programs demonstrating file handling, data structures, exception handling, functions, OOP, and text processing.
- Organized exercises into reusable modules and problem sets to demonstrate consistent programming practice and problem-solving skills.

GitHub:

github.com/jyothi-basu/visionaid-python-class-assignments

Expense Tracker (Team Project):

- Developed a CLI-based Python application to manage expenses using file storage
- Implemented filtering of expenses by date and category
- Collaborated using GitHub with version control
- Contributed to project documentation by creating and improving README

GitHub:

github.com/satyamagrawal28/expanse-tracker-project

COURSES:

- Python Programming (Core concepts, Data structures OOP, file handling) – Visionaid
- SQL and NoSQL Databases (MySQL, MongoDB) – Visionaid

Agentic AI (OpenAI APIs, LangChain, Prompt Templates, Tool Calling, AI Agents, Embeddings, Vector Databases, RAG) – VisionAid

EDUCATION:

Name of course	Name of institution	Completion year
MBA (Systems)	Sathyabama institute of science and technology	2027
BBA	Sun international institute of tourism and management	2025