Jyothi Basu Chodavarapu

CONTACT INFO:

Email: contact@jyothibasu.com
Website: jyothibasu.com
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 Junior Python Backend Developer role focused on building scalable backend services with modern AI integration.

TECHNICAL SKILLS:

Backend: Python, FastAPI, Flask, REST APIs, SQLAlchemy ORM, Pytest, Uvicorn, Gunicorn

Databases & Data Stores: PostgreSQL, MySQL, SQLite, MongoDB, Redis

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

DevOps, Hosting & Tools: Docker, Git, GitHub, Linux (WSL), Render, Railway, Cloudflare, Resend, Zoho Mail

AI Backend Integration: LangChain, LangGraph, Retrieval-Augmented Generation (RAG), ChromaDB, Pinecone, Tool Calling, Model Context Protocol (MCP)

Engineering Practices: Testing, Debugging, API Validation, Layered Architecture, Pull Requests, Code Review

Domain & Email: Custom Domains, 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:

Investment Portfolio Tracker:

- Developed a production-style FastAPI backend using layered architecture (Routes → Services → Repository) with SQLAlchemy ORM, JWT authentication, multi-Demat account management, BUY/SELL transactions, holdings, portfolio summaries, and configurable relational database support.
- Built a RAG pipeline using LangChain with Chroma/Pinecone vector-store support and configurable local or cloud embeddings, enabling users to upload PDF, DOCX, TXT, and Markdown financial documents and receive source-grounded answers with citations.
- Implemented a provider-agnostic LangGraph/LangChain AI assistant with tool calling to combine portfolio data, uploaded-document retrieval, and application-help knowledge while enforcing user and chat isolation through trusted backend context.
- Integrated Model Context Protocol through STDIO and authenticated Streamable HTTP transports with Argon2-hashed personal access tokens and chat-scoped document search.
- Added automated pytest coverage for authentication, portfolio operations, repository contracts, RAG, document uploads, AI orchestration, and MCP security workflows.

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

Quick project demonstration:
drive.google.com/file/d/1iw9AsVXBuy3HSt64Fx9r8KR4HNdldMgj/view?usp=drive_link

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

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.jyothibasu.com

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