






Abdullah Naeem

Python Backend Developer at Meissasoft

Passionate Software Engineer with expertise in both Frontend and Backend development. Experienced in building high-performance APIs using FastAPI and creating dynamic, user-friendly interfaces with React.js and Tailwind CSS. Skilled in developing scalable, efficient web applications and automation solutions. Adept at solving complex challenges in fast-paced environments, always eager to explore new technologies and drive innovation.

Contact Info

 **Nishtar Colony**, Lahore, Pakistan
 Abdullahnaeemgill1724@gmail.com
 [LinkedIn](#)
 [GitHub](#)
 **+923161762533**

Skills & Strengths

- FastAPI
- Django
- AI / LLMs
- Celery
- Python
- DRF
- Docker
- Redis
- PostgreSQL
- Third Party APIs
- Git / GitHub
- CI / CD

Education

Lahore Garrison University
Lahore / BS
Computer Science (2025)

Languages

- Punjabi - Native
- Urdu - Native
- English - Medium

Projects

Urban SDK | Meissasoft, Lahore | July 2024 - June 2025

Links:

- <https://dev.api.platform.urbansdk.com/v2/admin/docs#/>
- <https://www.urbansdk.com/>

Tools: FastAPI, Flask, Postgresql, github, swagger UI, jwt auth, Docker, Jira, Postman / Swagger, Rabbit MQ, Microservices Architecture, Pydantic

- Worked on a live project for a US client, transitioning from **Flask** to **FastAPI**, and enhancing backend services.
- Developed and optimized **FastAPI**-based **RESTful APIs**, improving backend performance, scalability, and maintainability.
- Led the migration of the **Flask** application to **FastAPI**, ensuring seamless integration with pre-existing services and data structures.
- Contributed to a **microservices-based architecture**, utilizing and integrating existing **microservices** to maintain modularity and scalability across the project.
- Utilized and integrated existing **microservices**, including models and utilities, ensuring reusability and consistency across the project.
- Applied fixes and enhancements to the **models** microservice repository to improve functionality and address client-specific requirements.
- Implemented **reverse engineering** techniques to analyze and refactor legacy code, increasing efficiency and code maintainability.
- Built reusable worker functions to interact with **APIs**, automating repetitive tasks and streamlining development workflows.
- Managed **PostgreSQL** database migrations, merging legacy tables with newly created **schemas** to ensure data integrity.
- Set up and maintain **Dockerized** environments for consistent development, testing, and deployment processes.

Hobbies

- Watching Cricket
- Table Tennis
- Social Media

- Collaborated closely with **frontend developers** to resolve technical challenges and ensure smooth API integration.
 - Actively addressed client and frontend queries, ensuring timely resolution and maintaining high client satisfaction.
 - Utilized **Git** for version control, managing codebase efficiently with multiple branches.
 - Tracked and documented daily work progress using **Jira**, ensuring timely updates and alignment with project goals.
 - Maintained consistent productivity by actively running the **Upwork** time tracker for 8 hours daily during remote engagements.
 - Collaborated with a **Scrum team** to design, execute, and maintain comprehensive test cases, ensuring high software quality and alignment with **Agile development practices**.
-

Human RL AI | Meissasoft, Lahore | **June 2025 - Dec 2025**

Links:

- <https://hrl-ai-dashboard-query-api-staging-74ye3o2uoa-uc.a.run.app/docs#/>

Tools: Python, FastAPI, SQLAlchemy, PostgreSQL, OAuth 2.0, JWT, Docker, Google Cloud Services, Artifact Registry, GitHub Actions, Async Programming, CI/CD, Microservices Architecture

Broker Portfolio Microservice (FastAPI):

- Built a production-grade microservice for automated broker portfolio ingestion from **Alpaca Trading API** into **PostgreSQL**.
- Designed an **asynchronous FastAPI architecture** with background tasks for real-time portfolio tracking.
- Implemented **OAuth 2.0 token flow** with automatic refresh and subscription-based access controls.
- Developed robust **error handling, retry logic, and connection pooling** using **SQLAlchemy ORM**.
- Created a **dual-database ingestion pipeline** optimized for time-series storage and historical snapshot queries.
- Built **modular service layers** for ingestion, account discovery, token management, and data enrichment.
- Containerized the service with **Docker** and deployed on **Google Cloud Run**, using **Cloud Scheduler** for cron jobs and **Artifact Registry** for images.
- Set up **CI/CD pipelines** with **GitHub Actions** for automated Docker builds and environment-specific deployments.

Broker Portfolio API Microservice:

- Developed a separate microservice to **fetch stored broker portfolio data** efficiently from the database.
- Implemented **JWT-based authentication** with a dedicated authentication microservice to secure API access.
- Designed scalable and modular API endpoints for **data retrieval, filtering, and aggregation**.
- Ensured smooth integration with the ingestion microservice while maintaining data consistency and performance.
- Applied **FastAPI async features** to support high-performance concurrent requests.
- Containerized and deployed using **Docker** with CI/CD pipelines for automated delivery.