Aniket Gupta

+91 9324399314 aniketgupta0295@gmail.com Thane, Mumbai GitHub LinkedIn
Portfolio LeetCode HackerRank

PROFESSIONAL SUMMARY

Backend-focused developer with hands-on experience building two full production-style systems from scratch — including JWT auth flows, Redis-backed rate limiting and caching, async SQLAlchemy schemas, and Docker deployment. Strong problem-solving foundation with 200+ LeetCode problems solved. Currently pursuing a B.tech in AI & Data Science (CGPA 9.28) and looking for a backend or full-stack internship.

EDUCATION

Btech in Artificial intelligence and Data science, *Thakur College Of Engineering and Technology* 2024 – present
CGPA: **9.28** Kandivali, India

Higher Secondary School, *KV Pendharkar college* 2022 – 2024
Score **: 71.83%** Dombivli, India

SKILLS

- **Language** — Python, Java, SQL, JavaScript
- **Frontend** — HTML5, CSS3, JavaScript , ReactJs, TailwindCSS
- **Backend** — FastAPI, Flask, REST APIs Design, Asynchronous Programming, JWT, Pydantic
- **Database & Caching** — PostgreSQL, MySQL, Redis, SQLAlchemy 2.0 (async), Alembic
- **Data/ML** — Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- **Tools & DevOps** — Docker, Git/GitHub, Vercel, Netlify, Render, Neon Postgres, Upstash Redis, Supabase

PROJECTS

Datalab — Full-Stack Data Analysis Platform ,
FastAPI · PostgreSQL · Redis · Docker · Supabase Storage · SQLAlchemy (async) · Alembic · Pandas · NumPy · JWT · SlowAPI

- Built a browser-based workspace for Data Analysis. CSV/XLSX/JSON upload, column-wise and dataset-wide cleaning, 5-chart-type visualization, and streaming download.
- Implemented JWT auth (access token only) via HTTP-only secure cookies, with server-side logout invalidation (logged_out_at), email verification, and magic-link password reset.
- Offloaded the Pandas/NumPy data engine to a thread pool (run_in_threadpool) so cleaning and chart-aggregation work never blocks the async event loop.
- Designed a two-tier cache — in-memory TTLCache (20 slots/900s) backed by Supabase Storage/local disk — with atomic invalidation on every mutating operation.
- Engineered graceful degradation when Redis is unavailable so cache and session reads keep working, while intentionally keeping rate limiting and email delivery fail-closed for safety.
- Applied SlowAPI rate limiting (5/min login, 3/hr password reset, 10-20/min mutations) with a custom exception hierarchy returning JSON or styled HTML error pages (404/429/500) by Accept header.
- Containerized with Docker Compose (PostgreSQL 18 + Redis 7 + web); auto-runs Alembic migrations on startup and switches between local-disk and Supabase Storage via a single ENV variable.

ExpenseFlow — Take Back Control of Your Expense ,
FastAPI · ReactJs · Async SQLAlchemy 2.0 · PostgreSQL · Redis · Alembic · JWT · Pydantic · FastAPI-Mail · Vercel

- Re-architected a Flask/MySQL prototype into an async FastAPI service covering expense tracking, settlements, and tag management, with a 4-table relational schema managed through Alembic migrations across 3+ schema revisions.
- Implemented JWT authentication (access token only) with email verification and self-service password reset via signed links (FastAPI-Mail), removing manual account-recovery support.
- Applied SlowAPI rate limiting backed by Upstash Redis to cap abusive traffic, with CORS configured for secure cross-origin access; this path fails closed if Redis is unreachable.
- Validated all request/response I/O with Pydantic schemas and published interactive API docs (Scalar) covering 100% of endpoints.
- Wrote a 22+ test pytest suite (pytest-asyncio, isolated test database) covering auth, expense/settlement CRUD, and tag management for repeatable runs.
- Dockerized the stack (FastAPI + PostgreSQL + Redis) with docker-compose; deployed serverlessly on Vercel (API) + Vercel (frontend) with managed Neon Postgres and Upstash Redis.

CERTIFICATES

Intermediate Backend Development with FastAPI — Packt

PostgreSQL — Infosys Springboard