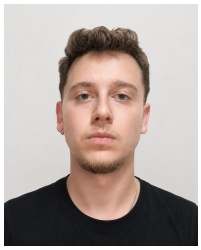
ALP YILDIZ

Software & Data Engineer | Python, SQL, and products that are live with real users

alpyildiz59@gmail.com

linkedin.com/in/alpyaldaz | github.com/alpyaldaz | alpyildiz.online

Turkish citizen, also EU citizen (Bulgarian passport)



PROFESSIONAL SUMMARY

IT engineer (BEng) who builds data-backed products and then keeps them running. I own Yolio end to end, live on the App Store, Google Play and the web: the ingestion pipelines, the PostgreSQL model, the FastAPI backend, the dashboards, and the part nobody volunteers for, which is being the only person on call when a scheduled job fails at six in the morning. I work across the stack, Python and SQL on the data side, React, Next.js and Flutter on the front, and I ship work tested and deployed through CI rather than left on a laptop.

TECHNICAL SKILLS

Languages: Python, SQL, JavaScript, TypeScript, Dart, C#, C++
Data and analytics: PostgreSQL, BigQuery, SQLite, Supabase, Pandas, NumPy, ETL pipelines, data modelling, data quality control, Power BI, Matplotlib, Excel and Google Sheets
Backend: FastAPI, Node.js, REST APIs, webhooks, Prisma, authentication and authorisation
Frontend and mobile: React, Next.js, Flutter, Dart, React Native (Expo), responsive UI
Cloud and DevOps: Google Cloud, Microsoft Azure (IoT Hub, Functions), Docker, Linux, Git, GitHub, GitHub Actions CI/CD
AI: LLM features shipped to production on the Claude API, automated evaluation of model output, AI coding assistants in daily professional use
Practices: Agile, Scrum, code review, automated testing, technical documentation, requirement gathering from non-technical stakeholders

PROFESSIONAL EXPERIENCE

Founder & Lead Developer | Yolio (yolio.com.tr) April 2026 - Present

Vehicle journey cost calculator, live on the App Store, Google Play and the web

- Own the product end to end: around 18,000 lines of Python behind 810 automated tests, a PostgreSQL model, a FastAPI backend, Docker deployment and GitHub Actions CI. I coordinate a contract developer on the mobile client and review the work before it ships.
- Built the scheduled ingestion layer: national fuel providers that each publish differently, some as HTML that changes without notice and some as REST, normalised into one canonical table. Every record keeps its source and fetch timestamp, so a disputed number can be traced rather than argued about.
- Audited the 19,671-row vehicle reference database into six documented defect classes: a hybrid flag that was wrong on every row (19,230 corrected), 6,498 malformed engine fields rebuilt, and the table de-duplicated. Ran through an idempotent pipeline that was adversarially reviewed and signed off before a single row changed.
- Built the accuracy monitor comparing predicted cost against user-reported actuals. Overall error came back near zero, which turned out to be two large opposing biases cancelling out, visible only after splitting by segment.

Full-Stack Developer Intern | Math Europe May 2024 - October 2024

Wroclaw & Poznan, Poland | 740 hours | 5-person Agile team

- Built the automated ETL path from Azure IoT Hub sensor APIs into a PostgreSQL backend, processing 10,000+ device telemetry events a day and replacing a workflow the operators had been running by hand.
- Wrote the SQL behind weekly KPI reporting and root cause analysis, and presented the results to non-technical colleagues as dashboards showing trends and the anomalies worth acting on.

PROJECTS

- **Malin Cloud, accounting platform for certified public accountants.** Next.js, TypeScript, PostgreSQL, Prisma, 426 automated tests. Specified the workflow with the accountants themselves through five defined user profiles, then held the release until every critical finding was closed.
- **Automated triage pipeline.** Python, external APIs and IMAP, LLM scoring, scheduled on GitHub Actions. Replaced a daily manual review of 60 to 70 records with a pipeline that deduplicates, scores each record against a defined profile and delivers only what passes. When the scoring environment turned out to be network restricted, I split it into two stages so the outbound call runs where it is permitted.
- **NeeDo, real-time marketplace app.** Flutter, Firebase, BLoC, Clean Architecture. Transactional flows tested under concurrent updates, with race conditions documented and verified against the expected state transitions.
- **Dukkan Defter, offline-first stock and sales tracker.** Flutter, Dart, SQLite (Drift), Riverpod, Supabase. 15 modules modelled on the shop owner's existing paper ledger rather than an idealised process, demonstrated on site.
- **Review Insight, AI review analysis app.** React Native (Expo), Supabase, Claude API. Pulls app-store reviews and uses an LLM to summarise sentiment and surface recurring issues.
- **Brazilian E-Commerce Analysis.** Olist dataset, roughly 100,000 orders. Python, Pandas and SQL across five joined tables: monthly order trends, demand concentration and revenue by category. Traced a sharp fall at the end of the series to an incomplete final period rather than a real decline, and said so in the write-up instead of reporting the drop.

EDUCATION

Engineer's Degree (BEng) in Information Technology October 2022 - March 2026

Collegium Da Vinci, Poznan, Poland | Taught in English

Preceded by an English preparation course at Politechnika Wroclawska, Wroclaw (2021-22). Coursework: Algorithms and Data Structures, Databases, Big Data, IoT and Cloud Computing, Software Engineering, Advanced Artificial Intelligence (Machine Learning, Neural Networks).

LANGUAGES

English: C1, the language of my degree, my internship and my working life. **Turkish:** Native.