Faraz Saeidi

Software Engineer · Full-Stack Developer

- ✉ farazsaeidii.2000@gmail.com
- ☏ +39 344 697 0811
- ⌂ Turin, Italy
- in linkedin.com/in/farazsaeidi
- ⌥ github.com/Farazsaeidi
- ◖ farazsaeidi.dev

▸ TECHNICAL SKILLS

LANGUAGES

Python · TypeScript · JavaScript · C · C++

FRONTEND

HTML · CSS · React · Next.js · Tailwind

BACKEND

FastAPI · REST APIs

DATABASES

PostgreSQL · MongoDB · SQL · NoSQL

DEVOPS & TOOLS

Docker · Git · GitHub · GitLab CI/CD · Linux

▸ SOFT SKILLS

Problem Solving · Team Collaboration · Critical Thinking · Adaptability · Time Management

▸ LANGUAGES

Italian	A2
English	B2 · IELTS
Persian	Native
Turkish	Native
Azeri	Native

▸ EDUCATION

Politecnico di Torino
B.Sc. Computer Engineering
2024 — Present · Turin, Italy

Algorithms & Data Structures · Operating Systems · Computer Networks · Databases · Computer Architecture

PROFILE

Computer Engineering student at **Politecnico di Torino** shipping production full-stack platforms end-to-end. Strong foundation in **Python / FastAPI**, **PostgreSQL** with PL/pgSQL business rules, and **Next.js / React / TypeScript**. Comfortable across system design, async API architecture, relational DB design, Docker, and GitLab CI/CD. Drawn to low-level, performance-oriented engineering; happiest building reliable, scalable systems on Linux.

EXPERIENCE

StarFrame / PolitoTA — Software Engineer
03/2026 — 05/2026
Turin, Italy

Full-stack educational SaaS platform · bilingual Persian/English (RTL)

- Owned a bilingual (Persian/English, full **RTL**) educational SaaS platform end-to-end — from domain model to deployed Linux server — applying the **Unified Process** methodology across Inception, Elaboration, and Construction.
- Designed a **PostgreSQL 17** schema derived from the domain model; enforced business rules at the database layer with enums, partial unique indexes, CHECK constraints, and **PL/pgSQL triggers** (one-active-license-per-course, installment-due-before-exam).
- Built the backend in **Python 3.12 / FastAPI** with async **SQLAlchemy 2.0 + asyncpg** and **Pydantic v2** — **16 routers** under `/api/v1` with auto-generated OpenAPI docs.
- Built the frontend in **Next.js 15 / React 19 / TypeScript** with a Tailwind brand-token design system, **TanStack Query** (server state), **Zustand** (client state), and **React Hook Form + Zod** (typed forms).
- Shipped **Persian/English i18n + full RTL** with **next-intl** across all layouts and components; verified parity in both directions.
- Containerised with **Docker Compose** and built a **GitLab CI/CD** pipeline (lint → build → push → SSH-deploy on merge to main); coordinated a **7-branch** team Git workflow reconciled into a single source-of-truth on main.

Danesh Bartar Software Development Co. — Software Development Intern
02/2024 — 08/2024
Tabriz, Iran

- Developed backend **REST API** endpoints in **Python / FastAPI** under senior code review.
- Contributed to **PostgreSQL** schema design and query tuning on existing tables.
- Implemented frontend features in **React / Next.js** against design specs; reproduced and closed bug tickets, participated in manual QA.
- Authored unit and integration tests with **pytest** to cover new endpoints and guard against regressions when fixing bugs.
- Participated in agile workflows — daily standups, sprint planning, and merge-request reviews — using **Jira** for task tracking.
- Worked in a **Dockerised, Linux-based** dev environment using a Git feature-branch workflow.

PROJECTS

Starview — NatCat Damage-Assessment Platform `github.com/Farazsaeidi/Starview`

FastAPI · SQLAlchemy 2 async · PostgreSQL · Next.js 15 · three.js + R3F · Leaflet

Full-stack platform turning disaster events into actionable decisions within 72 hours. Implemented **9 use cases** (UC-01 … UC-09) following the Unified Process, with audit-triggered **PostgreSQL** schema, JWT-secured editor APIs, and a marketing front-end featuring a Leaflet events map and a custom **three.js + React-Three-Fiber 3D globe with GLSL shaders**.

ShieldPlay — DRM-Protected Educational Content Platform `github.com/Farazsaeidi/Shieldplay`

Next.js · Python / FastAPI · PostgreSQL · Alembic · libVLC · Docker Compose

License-management platform for instructors distributing protected video, PDF, and audio coursework — conceptually similar to SpotPlayer, rebuilt with a modern scalable architecture. Designed a **cancellation & reactivation** licensing model backed by an immutable audit trail, abuse-prevention cooldowns, watermarking, device fingerprinting, and short-lived signed URLs from object storage.

PolitoTA — Bilingual Educational Platform · Unified Process Case Study `github.com/Farazsaeidi/PolitoTA`

FastAPI · SQLAlchemy 2 async · PostgreSQL · Next.js 15 · React · TypeScript · GitLab CI/CD

Persian/English (RTL) educational SaaS platform built under the **Larman / Unified Process** methodology, with the complete artifact set committed alongside the implementation: Inception (Vision, Use-Case Model, Supplementary Spec, Glossary, Risk List) and Elaboration (Detailed Use-Case Model, Domain Model, SSDs, Operation Contracts). **PostgreSQL** schema derived from the Domain Model, with **PL/pgSQL triggers** enforcing business rules and an append-only audit log. Coordinated as a 7-branch team workflow.

ENGINEERING PRACTICES

- **Unified Process / Larman methodology** — drove Inception (Vision, Use-Case Model, Supplementary Spec, Glossary, Risk List) and Elaboration (Detailed Use-Case Model, Domain Model, SSDs, Operation Contracts) deliverables before Construction.
- **Business rules at the database layer** — encoded BRs as PL/pgSQL triggers, CHECK constraints, and partial unique indexes so invariants survive any client; immutable append-only `audit_log` behind every state-mutating contract.
- **Schema-first, contract-first design** — every API route maps to a documented use case; every table traces back to a Domain Model attribute or Glossary entry.
- **Team Git workflow** — coordinated a 7-branch contributor model with disciplined merging into a single source-of-truth on `main`, including cross-branch reconciliation against raw input sources.
- **PII discipline** — load-bearing `.gitignore` to keep raw exports and re-identification mappings out of the index; only anonymised mirrors committed.

CERTIFICATES & COURSES

- **CCNA Routing & Switching** — TCP/IP, subnetting, routing, switching, troubleshooting
- **Cisco Packet Tracer** — VLANs, routing config, network topology design
- **SQL Certificate** · HackerRank — joins, aggregations, subqueries
- **Python for Everybody Specialization** · University of Michigan
- **Machine Learning Course** · Politecnico di Milano — supervised learning, classification, regression
- **Anthropic Academy — Claude Code** · Claude API · MCP · Agent Skills

Interests — *volleyball, tennis, skating, and outdoor activities.*