Umair Talib

Applied AI Engineer | LLM and speech systems

Siegen, Germany | umair.talib99@gmail.com | linkedin.com/in/umairtalib | github.com/UmairTalib

PROFILE

Applied AI engineer focused on LLM and speech systems whose results can be checked. After over 1.5 years as a software engineer, I'm developing AI products part-time: an AI platform for public-funding workflows at ATTEC, as its only developer, and an AI-assisted planning tool for municipal citizen participation at the University of Siegen. I make the design decisions and write the specs, AI coding agents implement, and I review and test every change. M.Sc. Computer Science student and co-author of a conference paper on removing interviewer speech from clinical interview recordings, published by IEEE (CoDIT 2026).

WORK EXPERIENCE

Software Engineer, AI Product (part-time)

ATTEC UG (haftungsbeschränkt), Siegen (hybrid) | 06/2026 – present

- Develop, as its only developer, an AI platform that aims to automate public-funding workflows for German SMEs and the funding experts who advise them, from finding programmes to the application and managing the funded project. The first stage already works: it shows which programmes from a curated, growing catalogue fit a company, with every checked condition, its reason and the official source (Python, FastAPI, React 19, TypeScript, LLM API).
- Built a deterministic Python rules engine: code decides and the LLM only reads language, so the same input always gives the same result, and each result replays from its data version, engine version and date.
- Designed grounded LLM extraction: the model must cite the words it used, code verifies them in the input, and the user confirms. On three evaluation sets used for tuning (113 cases, no unseen data) it gave 0 invented or wrong answers, with a tag recall of 98.5%.
- Model each programme's conditions as yes/no facts and checks in a pydantic-validated YAML model. Every fact needs a verbatim quote from a saved official text, and data tests check the quotes, numbers and dates.
- Set up verification: about 3,700 automated tests (pytest, Vitest, Playwright) with property, golden and mutation tests, and GitHub Actions CI with secret scanning and dependency audits.
- Run an AI-native workflow for each feature: spec, build with AI coding agents, three independent AI reviews and fixes, my own review and tests, and written decision records.

Student Research Assistant (WHB), Full-Stack Development (part-time)

University of Siegen, research project FUSION, Siegen (hybrid) | 07/2026 – present

- Develop the FUSION toolbox (Werkzeugkasten), an AI-assisted planning tool for municipal citizen participation: it derives a project profile from a description and documents, recommends participation formats and turns the choice into a roadmap with budget, tasks and calendar.
- Hybrid recommendation: deterministic filters and a fit score select candidates, an LLM rates them against format profiles with quoted reasons, and fixed rules check its answer. Every AI task has a fallback without the LLM, and a collection of test cases checks the recommendations.
- Modular monolith with 15 domain modules: FastAPI (Python 3.13), SQLAlchemy 2, Alembic, PostgreSQL 16 as the only data store (jobs and events in tables) and a background worker; front end in Next.js 16, React 19 and TypeScript with a client generated from OpenAPI.
- Built with AI coding agents in a spec, plan, build and review cycle per sub-project. About 9,000 automated tests (pytest, Vitest, Playwright), including a permission matrix that fails when a route has no rights entry, and GitHub Actions CI.
- Earlier, built the first version (07–08/2026) with an AI coding agent (Antigravity): accounts, roles and a Kanban board on FastAPI, PostgreSQL and Next.js; fixed four of five critical findings of an AI-assisted security audit. Reviewed and approved the architecture of the current version (09/2026).

Software Engineer

ZweiDevs | 02/2022 – 09/2023

- Built React and Next.js front ends and Node.js and Express back ends for several projects, including an EdTech platform and a Web3 game (React front end, game mechanics in Unity).

- Implemented Stripe payments, JWT-based login and role-based access for the EdTech platform, deployed services on AWS (EC2, S3) with PostgreSQL and MongoDB, and tested with Jest and Cypress.

EDUCATION

M.Sc. Computer Science
University of Siegen | 10/2023 – present

- Current average 2.1 (German scale, 1.0 is best)

B.Sc. (Hons) Computer Science
University of Central Punjab, Lahore, Pakistan | 10/2017 – 03/2022

- Final-year project: multi-level classification of hate speech in Urdu and Roman Urdu, in a team of four

RESEARCH AND PUBLICATIONS

Publication
U. Oleh, U. Talib, M. A. Alam, and R. Obermaisser, "Leakage-Resilient Diarization and Feature Extraction Framework for Robust Multimodal Mental Health Corpora," in *Proc. 12th Int. Conf. Control, Decision and Information Technologies (CoDIT)*, Bari, Italy, Jul. 2026, pp. 2508–2512, doi: 10.1109/CoDIT70676.2026.11630977.

Research project: An Audio-Dominant Framework for Robust Depression Detection and Clinical Explainability
University of Siegen, Chair for Embedded Systems, module Project Work (grade 1.0) | 04/2025 – 03/2026

- Built a speaker-verified diarization and transcription pipeline (pyannote.audio, Faster-Whisper) with a text filter for interviewer prompts, and transcribed 34.1 hours of interview audio in about 2.5 hours on 10 GPU nodes with SLURM. In the published paper, interviewer contamination in participant transcripts fell from 14.18% to 1.09%.
- Trained a gated fusion model on WavLM and Mental-RoBERTa embeddings with a session-level Transformer on top, for screening of elevated depressive symptoms (PHQ-8 of 10 or more). Held-out test on 56 participants: ROC-AUC 0.614, recall 88.2% at precision 41.7%.
- Built an explanation pipeline that turns attention-based evidence into text explanations with an LLM, and a research dashboard in Streamlit that shows the results.

Course project: NAS-Bench-201 transfer to cosmic-ray detection
University of Siegen, with one co-author | 08/2026

- Transferred NAS-Bench-201 architectures to cosmic-ray detection in telescope images. Thirty fine-tuned networks reached best validation AUROCs between 0.976 and 0.989.

SKILLS

- **AI and ML:** PyTorch, Hugging Face Transformers, WavLM, Whisper, pyannote.audio, LLM APIs with structured output, evaluation sets
- **Backend:** Python, FastAPI, pydantic, SQLAlchemy, Alembic, PostgreSQL, REST, JWT authentication
- **Front end:** TypeScript, React, Next.js, Tailwind CSS
- **Engineering practice:** pytest, Vitest, Playwright, GitHub Actions, Docker Compose, Ruff, mypy, Git
- **Compute:** SLURM on an HPC cluster with A100 GPUs
- **AI coding agents:** Claude Code and Antigravity, controlled by tests, evaluation sets and decision records

LANGUAGES

English: C1, fluent (M.Sc. programme taught in English)

ADDITIONAL INFORMATION

- **Further training:** Artificial Intelligence course, Samsung Innovation Campus, Lahore, 10/2021 – 12/2021