Kaan Değirmenci

AI CV Link: https://ai-cv-kaan-degirmenci.streamlit.app

Computer Science Student | Future System Architect
Frankfurt, Germany
kaandeg@gmail.com | LinkedIn: https://www.linkedin.com/in/kaan-degirmenci-23a5a03a4/

Professional Summary

Forward-thinking Computer Science student evolving from a traditional coder into a system-oriented engineer. Strong at bridging low-level computing (C, C++, Assembly, real-time systems, embedded/IoT) with high-level software and data architecture (Java OOP, SQL, AI/ML). Known for a first-principles mindset in AI and an architectural approach to software design, focusing on scalability, correctness, and end-to-end ownership.

Education

Frankfurt University of Applied Sciences
B.Sc. in Computer Science (Informatik) — 3rd Year Student
GPA: 2.4 (German grading scale)

Selected Coursework & Grades

- Introduction to Programming with C — 1.0

- Object-Oriented Programming with C++ — 1.7

- Real-Time Systems — 1.7

- Databases (SQL) — 2.3

- Artificial Intelligence — 2.3

- Probability & Statistics (Data Analysis with R) — 2.7

- Computer Architecture (Assembly & HW/SW Interface) — 3.3

Technical Skills

Core Engineering Philosophy

- System Architecture & End-to-End Design

- Object-Oriented Design (OOP)

- First-Principles AI & Engineering Logic

Programming Languages

- Java (Advanced, OOP, GUI, concurrency)

- Python (AI/ML, data processing)

- C++ / C (Embedded systems, real-time logic)

- SQL (Relational databases)

- R (Statistical data analysis)

Artificial Intelligence & Machine Learning

- Deep Learning: CNN architectures, Conv2D & pooling layers, backpropagation

- Optimization: Adam, SGD, gradient-based learning

- Supervised Learning: Binary vs. multi-class classification (Sigmoid, Softmax), Decision Trees, Random Forests

- Unsupervised Learning: K-Means clustering, Elbow Method, Inertia, Silhouette Score

- Reinforcement Learning: Agent–environment interaction, MDPs, delayed reward logic

- Mathematical Foundations: Linear algebra (matrix operations), calculus (gradients)

Projects

https://github.com/kaanbabaa?tab=repositories

Advanced Traffic Simulation Wrapper (SUMO)

Tech Stack: Java, Object-Oriented Design, GUI (Swing/JavaFX), SUMO Engine

- Engineered a robust Java wrapper around the SUMO traffic simulation engine using strict OOP principles

- Designed a responsive GUI to visualize real-time traffic data

- Implemented multi-threading to decouple simulation logic from UI rendering

- Parsed and controlled complex XML-based simulation configurations programmatically

Machine Learning & Computer Vision Fundamentals

Tech Stack: Python, NumPy, Custom Neural Networks

- Built CNN architectures for multi-class classification (CIFAR-10 Vehicles, MNIST Digits, Kaggle)

- Applied a glass-box approach by manually reasoning about backpropagation, gradients, and matrix operations

- Used structured train/test splits to validate generalization and mitigate overfitting

Smart Trash Bin – IoT System

Tech Stack: C++, ESP8266, Firebase, Google Apps Script

- Designed an end-to-end IoT architecture connecting physical sensors to a cloud backend

- Programmed microcontroller logic to process HC-SR04 distance sensor data

- Implemented a software-based signal smoothing algorithm to reduce sensor noise

- Enabled real-time data updates via Firebase and WebSocket communication

AI-Powered CV Assistant (RAG Application)

Tech Stack: Python, Streamlit, Gemini API

- Developed a Retrieval-Augmented Generation (RAG) style application to interact with structured CV data

- Designed a multi-agent architecture (Visionary vs. Auditor) to reduce hallucinations

- Implemented session-state memory and modular prompt engineering for adaptive AI behavior

---

Internship Objective

Seeking a Summer 2026 internship that connects low-level engineering (embedded systems, IoT, real-time logic) with high-level software architecture (AI, cloud, data-driven systems). Motivated to contribute beyond task execution by participating in architectural decisions, scalable system design, and real-world engineering problem solving.