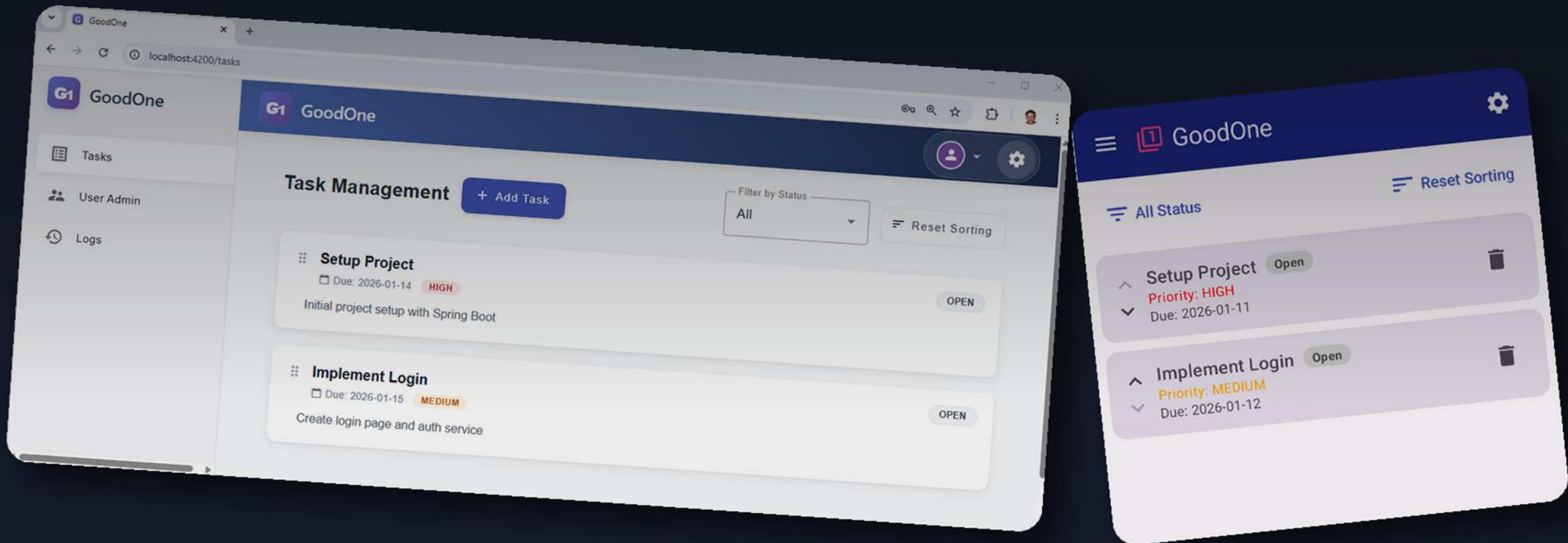


Software-Entwicklung mit AI



Agenda

Demo & Kontext

- Angular Web & Android App

AI-Werkzeuge

- Junie AI, ChatGPT

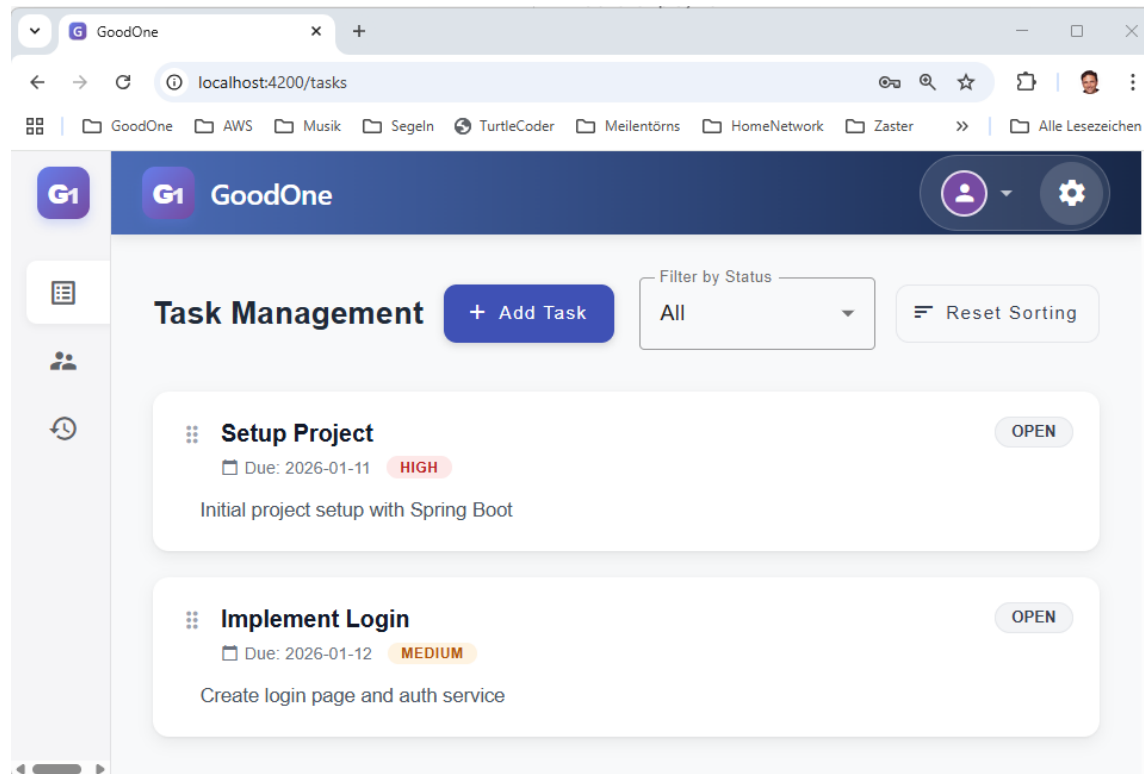
Praxis

- Vorgehensweise & Beispiele
- Wir implementierten ein Feature

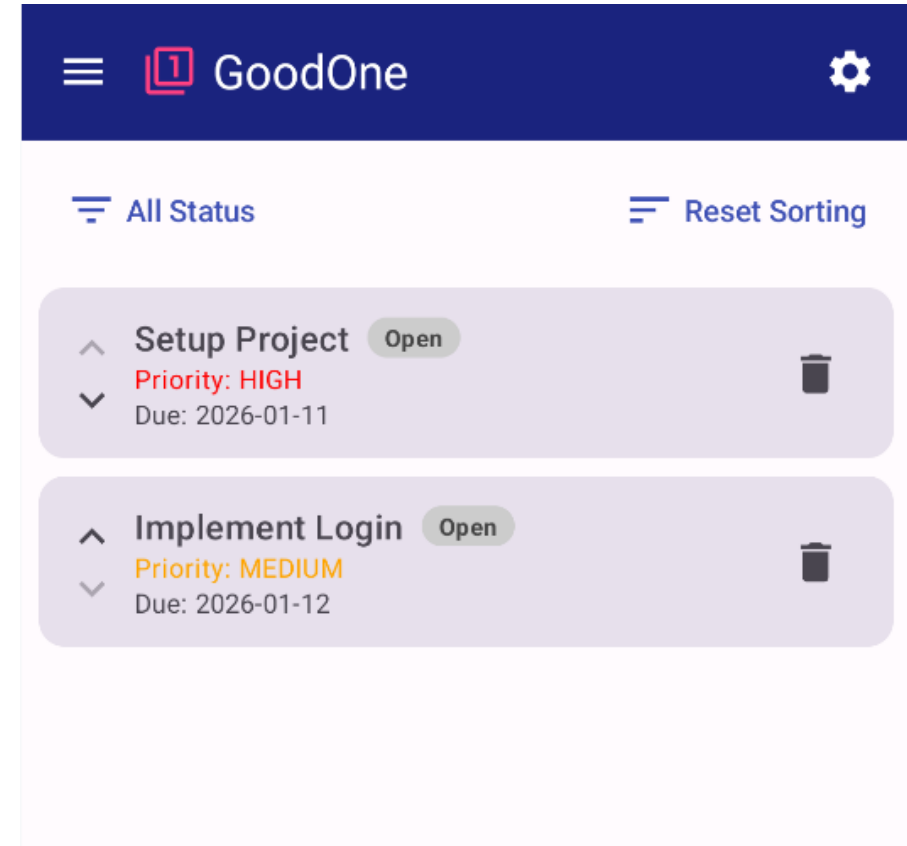
Einordnung

- Fazit & Ausblick

Anwendungsübersicht

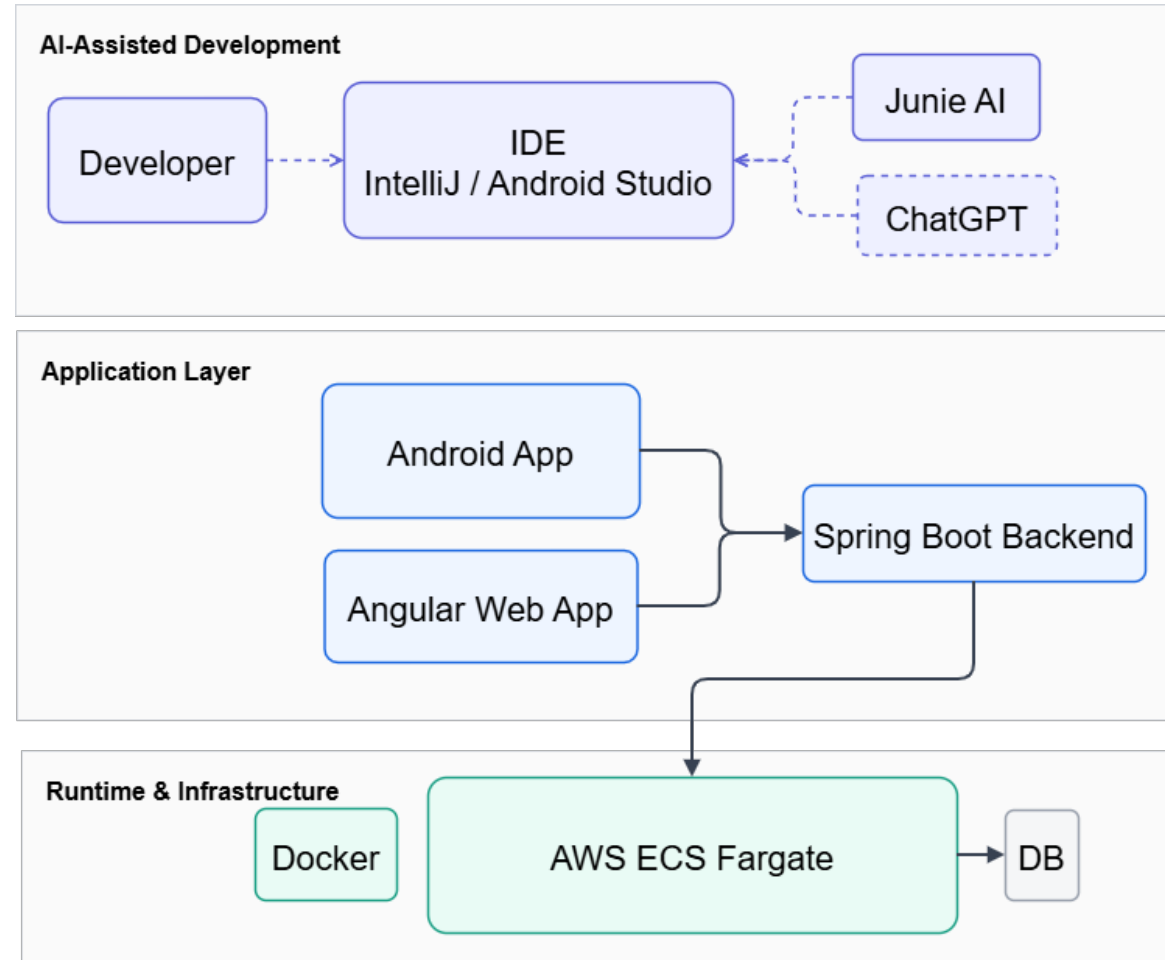


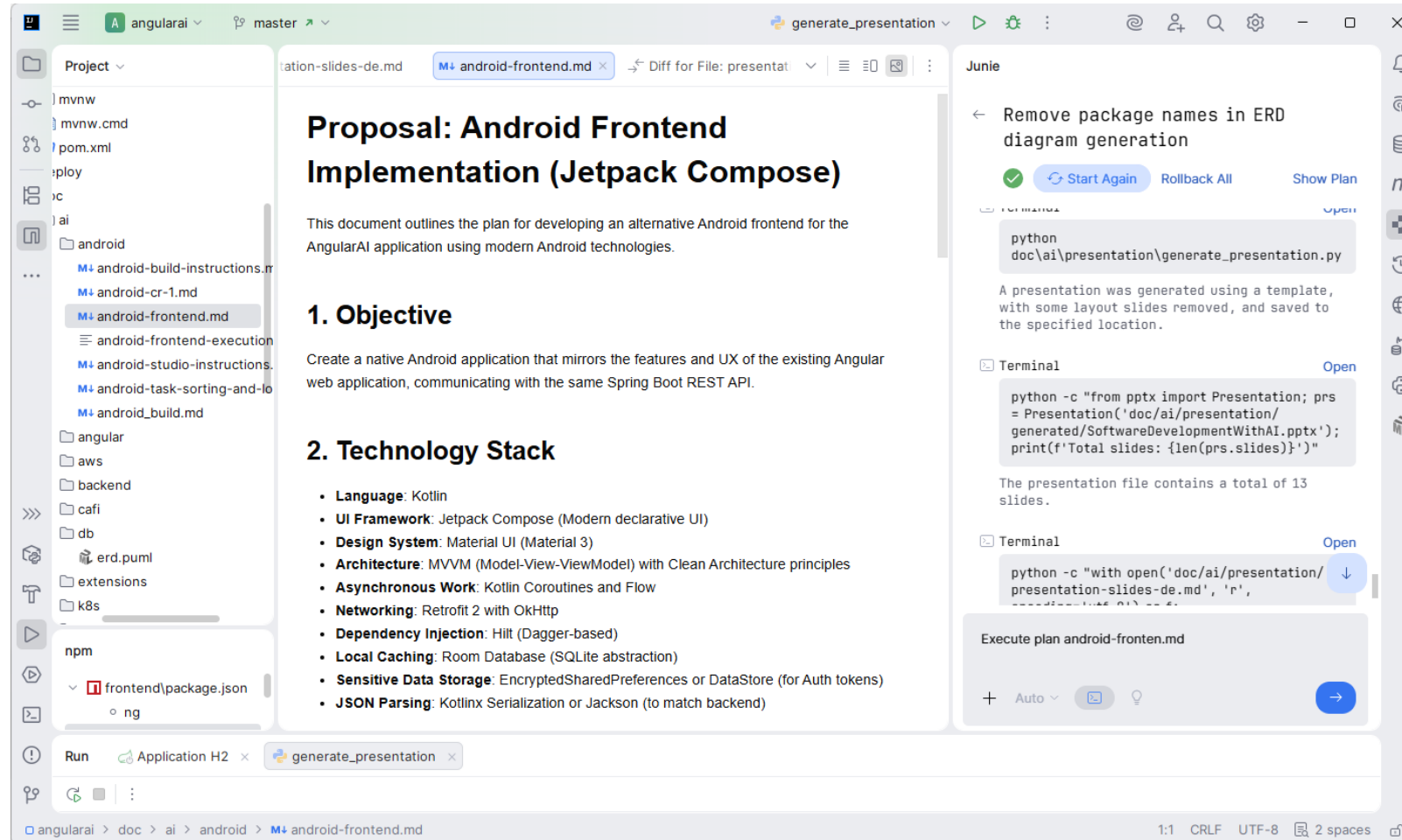
Angular Web-UI www.goodone.ch



Android App

Architekturübersicht – Infrastruktur





The screenshot displays an IDE interface with the following components:

- Project Explorer (Left):** Shows a file tree with folders like `mvnw`, `android`, `angular`, `aws`, `backend`, `cafi`, `db`, `erd.puml`, `extensions`, and `k8s`. The `android` folder is expanded, showing files like `android-build-instructions.m`, `android-cr-1.md`, `android-frontent.md` (selected), `android-frontent-execution`, `android-studio-instructions`, `android-task-sorting-and-lo`, and `android_build.md`.
- Main Editor (Center):** Displays the content of `android-frontent.md`.
 - Title:**

Proposal: Android Frontend Implementation (Jetpack Compose)
 - Text:** "This document outlines the plan for developing an alternative Android frontend for the AngularAI application using modern Android technologies."
 - Section 1: Objective**

Create a native Android application that mirrors the features and UX of the existing Angular web application, communicating with the same Spring Boot REST API.
 - Section 2: Technology Stack**
 - Language:** Kotlin
 - UI Framework:** Jetpack Compose (Modern declarative UI)
 - Design System:** Material UI (Material 3)
 - Architecture:** MVVM (Model-View-ViewModel) with Clean Architecture principles
 - Asynchronous Work:** Kotlin Coroutines and Flow
 - Networking:** Retrofit 2 with OkHttp
 - Dependency Injection:** Hilt (Dagger-based)
 - Local Caching:** Room Database (SQLite abstraction)
 - Sensitive Data Storage:** EncryptedSharedPreferences or DataStore (for Auth tokens)
 - JSON Parsing:** Kotlinx Serialization or Jackson (to match backend)
- Run Bar (Bottom):** Shows `Application H2` and `generate_presentation` tabs.
- AI Assistant Panel (Right):** Labeled "Junie", it shows a task: "Remove package names in ERD diagram generation". It includes a "Start Again" button, a "Rollback All" button, and a "Show Plan" button. Below this, it displays a code snippet for a Python script: `python doc\ai\presentation\generate_presentation.py`. It also shows a terminal output: `python -c "from pptx import Presentation; prs = Presentation('doc/ai/presentation/generated/SoftwareDevelopmentWithAI.pptx'); print(f'Total slides: {len(prs.slides)}')"`. At the bottom, it says "The presentation file contains a total of 13 slides." and "Execute plan android-fronten.md" with a blue arrow button.

Clients

- Angular Web-UI
- Android App (Jetpack Compose)
- Test-Daten-Generator

Backend

- Java, Spring Boot REST-API

Persistenz

- PostgreSQL (AWS RDS)
- Lokal: H2

Junie AI (JetBrains)

- Code-Generierung, Refactoring, Tests

ChatGPT (OpenAI)

- UX-Design, Architektur-Reviews, Konzeptarbeit

IDE-Integration

- IntelliJ IDEA, Android Studio

Vorgehen bei Design durch AI

1. Prompting

- Ziel und Kontext definieren

2. Struktur

- Entitäten, Attribute, Menüs

3. Planung

- Detaillierter Markdown-Plan

4. Review

- Entwickler prüft & ergänzt

5. Implementierung

- Code + Tests durch AI

6. Qualität

- Entwickler bleibt verantwortlich

Implementiere ...

- Android App mit identischen Features wie Web
- Read-only Admin Access
- Behebe Defekt XYZ
- AWS Cloud Deployment
- UI-Integrationstests mit Cypress
- Architektur-Diagramm
- Confluence-Dokumentation
- Powerpoint-Päsentation

KI-gestützt mit Junie (IDE-zentriert)

- Direkte IDE-Integration
- Kontext aus Projekt & Code
- Schnelle Iterationen
- Fokus auf Implementierung

ChatGPT (extern & beratend)

- Architektur- & UX-Feedback
- Alternative Lösungsansätze
- Formulierung & Dokumentation
- Kein direkter Code-Zugriff

Einschränkungen im Firmenumfeld

Security & Compliance

- Keine Kundendaten extern

Tool-Vorgaben

- Zentrale AI-Freigaben

Technologie-Gap

- Verzögerter Zugang zu aktuellen Modellen

Know-how

- Prompting-Kompetenz nötig

Demo bestätigt: AI ist produktionsreif

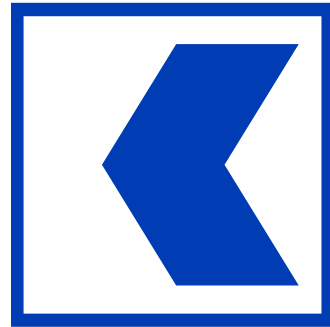
Werkzeuge: Junie & ChatGPT ergänzen sich ideal

Praxis: Weniger Code, mehr Qualität

Ausblick: AI-gestützte Entwicklung wird Standard

Schneller bauen. Besser verstehen. Qualität sichern.

Mit Junie AI & ChatGPT



Zürcher
Kantonalbank