Pau Kraft

Next.js expert and full-stack engineer with a passion for building AI-driven applications. Core developer on Teampilot AI, where I built the core multi-agent system and the web connection feature, comparable to Perplexity.ai in its ability to synthesize information from across the web. Leveraged LLMs (OpenAI, Gemini, Claude) and the Vercel AI SDK. Developed a data enrichment platform, showcasing proficiency in Next.js server components, server functions, and modern UI development (shadcn). Experience in RAG, web crawling, and prompt engineering for optimal AI performance.

Personal details

Address	Freiburg
Country	Germany
E-mail	me@paukraft.com

Educational background

Fachabitur in Software Development 2018 - 2020
Berufsbildende Schule Südliche Weinstraße

Professional experience

Fullstack Developer 2022 - Present
SODEFA

- Developed 5+ full-stack web applications (Next.js, tRPC, PostgreSQL) for diverse clients, including a real estate platform and a tour planning software with route optimization, collaborating directly with clients in regular meetings.
- Gained experience with Microsoft Dynamics 365 Business Central and worked effectively in small agile teams (2-5 developers).
- Core developer of Teampilot AI, a generative AI platform, designing and implementing its core multi-agent system and the Perplexity.ai-like web connection feature (search engine API, Cheerio crawling, custom summarization).
- Integrated OpenAI models (via API) for NLP tasks within Teampilot AI and implemented RAG using Qdrant (vector database) and OpenAI embeddings.
- Onboarded new Teampilot AI customers, conducting training sessions and providing ongoing customer success support.
- Solely developed a data enrichment platform (Next.js: App Router, Server Components, Server Functions; PostgreSQL; shadcn/ui), leveraging Teampilot AI's multi-agent capabilities and featuring a user-friendly, animated interface.

Fullstack Developer (Part-Time) 2020 - 2022

it-Plan

- Led the design, development, and successful deployment of Pegel-Dashboard, a web platform (React, Node.js/Express.js) for internal monitoring and management of a large-scale project digitizing physical water level records for German government agencies.
- Built a Node.js resource management system that optimizes digitization by dynamically scaling Hetzner servers based on processing demands, preventing database overload and ensuring efficient resource utilization.
- Developed an interactive viewer (Next.js, tRPC) for overlaying digitized water level data onto original record images, enabling visual verification and quality control.
- Engineered a custom, high-performance canvas-based rendering solution to handle large image files and complex coordinate system transformations (non-standard origins, varying aspect ratios) within the interactive viewer.
- Collaborated with a team of 3 developers working on the core digitization process, providing them with tools for monitoring and managing their work.

Skills

Next.js
React
JavaScript (ES6+)
TypeScript
Node.js
tRPC
PostgreSQL
Qdrant
LLMs
OpenAI API
RAG
Web Crawling
Cheerio
Full-Stack Development
AI Integration
Data Enrichment
Serverless
Hetzner
AWS
GCP
Vercel
Agile