Siim Liimand

Software Developer · AI Automation & Web Platforms

Tallinn · Born 10 May 1987

Updated: November 2026

siim.liimand@gmail.com
+372 5647 1230
github.com/siimliimand
seoweb.ee

SUMMARY

I am a software developer with **15+ years** of experience and broad knowledge across IT services, e-commerce and a wide range of business processes. I build solutions end to end — from problem analysis and technical decision-making through testing, deployment and further development.

My strengths are independent work, analytical thinking and finding practical solutions. I am interested in how technology and artificial intelligence help solve real problems, automate workflows and make work more efficient. I use AI tools in my development work and work with LLM-based solutions, agents and automated workflows.

WORK EXPERIENCE

Information Systems Administrator — Amazina OÜ 05.2026 – 09.2026

Day-to-day management and development of the company's e-shops, including building two new shops from scratch. Also built a support-desk tool that classifies incoming emails and drafts reply messages.

Developer — Dolm IT 08.2017 – 05.2026

Full-spectrum web development: from simple company websites and e-shops to large public-procurement projects. Alongside server-side work, direct client communication — from clarifying requirements to delivering the finished solution.

Software Developer — ERGO Insurance SE 05.2017 – 08.2017

Software development for an insurance group's core systems — building and maintaining business logic for policy and claims handling in a large organisation's complex IT environment.

PHP Developer — Web Expert 10.2015 – 10.2016

Website development across larger and smaller projects; most of the time spent on back-end development in PHP.

PHP Developer — Concise Systems 08.2012 – 10.2015

Development on the company's in-house PHP framework: continued development and maintenance of existing client websites.

PHP Developer — Indium OÜ 10.2010 – 05.2011

Updating websites through a configuration-based management system — content and structure changes made via the system's settings.

SELECTED PROJECTS

All projects are my own, built by me from start to finish with AI assistance (not all of them are finished).

Full portfolio: seoweb.ee

Subcontractor Marketplace

CONCIERGE MARKETPLACE

WHY

Construction project managers in Estonia still find subcontractors by phone and personal networks, and availability stays unknown until someone replies — if the crew shows up at all. The platform seeds verified supply from Estonian public registers in one narrow segment (interior fit-out, Tallinn/Harju, six trades), matches requests and books real crew capacity instead of trusting a calendar. It is a proof-of-concept pilot where request mediation is initially handled manually under the team's lead. Marketplace automation enters development only after real demand has been validated and the pre-defined success metrics met — this avoids premature investment in complex technical infrastructure and grounds automation work in proven market need.

FOR WHOM

Construction project managers who need a qualified crew fast, and trade contractors who want relevant jobs without paying to be listed.

TECH

The whole product runs under a single Cloudflare Worker: a Hono-based TypeScript API, a React 18 + Vite SPA served as static assets and a D1 database behind Drizzle ORM, managed by eight versioned SQL migrations.

Staff authenticate through Cloudflare Access (JWT validated against the team JWKS with an allowlisted-email check), while providers respond to requests via single-use email tokens without creating an account. Cloudflare brings compute, database, email, queues, cron jobs and AI inference under a single deployment; pilot data resides in the EU region.

Data integrity rests on an immutable audit log from which the dashboard metrics are also computed. Every change is logged within the same D1 transaction, and the first-confirmed-booking-wins rule is enforced by a database trigger. The email state machine combines three producers: a secret-protected webhook, RFC 3464 bounce-report parsing and a cron-based sweep for stale sends. All state transitions are idempotent, and unconfirmed deliveries are explicitly marked as inferred.

Release safety is provided by 320 integration tests that run against a real D1 database in Miniflare.

Tech stack: Cloudflare Workers, Hono, React 18, Vite, D1, Drizzle ORM, Cloudflare Access, Queues, Cron and Miniflare.

Cloudflare Workers · Hono · D1 + Drizzle · React 18 + Vite · Cloudflare Access

github.com/siimliimand/subcontractor-marketplace

Girder

AI AUTOMATION · FLAGSHIP

WHY

Girder exists because AI agents are useful right up until you have to let them touch production code — nobody can guarantee the agent stays in bounds. Girder makes autonomous development safe: describe a need, approve the spec, and sandboxed agents implement, test and deliver it to main, while guardrails — test-integrity locks, budget caps, audit-gated merges — are enforced outside the agent's reach.

BIGGEST LESSON

A constraint that keeps an agent in line must live in the architecture, not in the prompt.

FOR WHOM

Development teams that want to delegate routine development to AI without losing control.

TECH

A Python `asyncio` daemon with a SQLite database in WAL mode manages persistent state for this single-user, local tool. Crash-safety comes from a single writer process, with no separate database server.

Agents run in rootless Podman sandboxes with no network access and all unnecessary capabilities dropped, so security constraints stay entirely outside the agent's reach. The runtime is implemented by hand, without LangChain, to guarantee deterministic logging and full control over context compaction. Role-based model routing is handled by an OpenAI-compatible gateway, which helps keep model-related costs under control.

Python · FastAPI · Podman · SQLite · LLM orchestration

github.com/siimliimand/Girder

Rajahinta.fi

DATA PLATFORM

WHY

Beverage prices differ several-fold between Finland and Estonia, and a cross-border shopper can't easily see what is actually cheaper after duties and transport. Rajahinta is a live price index and landed-cost calculator where every result is traceable to the exact inputs, the excise-dataset version and the timestamp that produced it.

BIGGEST LESSON

Every e-shop's data is different.

FOR WHOM

Cross-border shoppers and importers who must justify their cost calculations.

TECH

The whole system is built on the Cloudflare Workers platform in TypeScript. The API layer is served by Hono, and the web application uses Next.js with OpenNext. Persistent data lives in a D1 database with Drizzle ORM, where excise datasets are stored as append-only, versioned records in EU jurisdiction. Observation logs go to R2, and background processing is orchestrated by Cloudflare Queues, Workflows and cron-based workflows.

The architecture was deliberately migrated from the original combination of NestJS, PostgreSQL, Redis and Kubernetes to Cloudflare-managed services. The goal is to minimize infrastructure overhead so that a single person can keep the entire system running without operating servers. Durable Objects take over Redis's role, enforcing rate limits and guaranteeing idempotent operations.

TypeScript · Cloudflare Workers · Hono · D1 + Drizzle · Next.js

rajahinta.fi · github.com/siimliimand/rajahinta-fi

PainSignal

AI PRODUCT

WHY

PainSignal starts from an observation: the best product ideas are not born in a meeting room but online — on Reddit, thousands of people repeat the same problems. The service monitors those pain signals, converts them into validated micro-SaaS ideas and coaches teams with an AI coach that suggests — never oversteps.

BIGGEST LESSON

What shipping a real AI product takes: six cooperating microservices over gRPC, and the insight that AI delivers value when you give it a concrete task, not free rein — and then validate the result against strict rules.

FOR WHOM

Founders and product teams who want to validate ideas before writing code.

TECH

TypeScript end to end: the AI coach runs as six cooperating microservices over gRPC — typed, low-latency contracts that double as guardrails on AI behaviour, backed by deterministic rules and human-approval gates. Managed services (Supabase Postgres, Lemon Squeezy billing, Vercel, Cloudflare) keep the solo-built multi-platform product — web, iOS, Android, Chrome — running without an ops team.

TypeScript · gRPC · Microservices · Multi-platform · Live product

painsignal.app · github.com/PainSignal

Unejutumaa

AI + AUDIO

WHY

My wife and I publish free Estonian bedtime stories that show neurodifferences (ASD, ADHD, dyslexia) as gifts — every story ships as text and as multi-voice audio. Publishing regularly is impossible by hand without days of writing and recording, so I built the whole pipeline as a WordPress plugin: pick a character and a topic, and AI generates unique ideas (deduplicated against existing stories), writes an 800–1500-word child-safe tale, converts it into a line-by-line audio script, synthesizes every line with Estonian voices, merges the audio and designs the cover art — publishing is one click and every stage stays editable.

BIGGEST LESSON

In AI content production the human checkpoint between stages matters more than full automation.

FOR WHOM

Estonian families get a free story library; our two-person team gets to publish a new story every week.

TECH

PHP 8 and WordPress — the author already publishes stories in WordPress, so the plugin fits the existing workflow with no new platform. Ideas, stories and audio scripts are generated by openai/gpt-luna-latest, which writes Estonian well and is cheap; jutusta.ee provides natural Estonian voices that the bigger TTS vendors do not offer.

PHP 8 · WordPress · OpenAI GPT Luna · jutusta.ee TTS

unejutumaa.ee · github.com/siimliimand/ai-bedtime-stories

Erametsad

MARKETPLACE

WHY

Estonian forest transactions are fragmented and opaque: owners don't know what their forest is worth and buyers have few trustworthy places to bid. Erametsad creates the missing marketplace — cutting-right and forest-property auctions where the owner compares bids and buyers are vetted, with Estonian eID login.

FOR WHOM

Estonian forest owners and buyers of forest property.

TECH

The application runs on the Cloudflare Workers platform using Next.js 15 and OpenNext, consolidating all state management into a single infrastructure. Instead of an external database it uses D1 with Drizzle ORM. Each auction gets its own single-threaded Durable Object, which serializes bid admission, automatic auction extensions and real-time SSE updates. This eliminates race conditions caused by concurrent requests and keeps the auction correct even under heavy load.

Buyers authenticate through Estonian eID solutions: Smart-ID, Mobile-ID or ID-card. The originally planned combination of Payload CMS and PostgreSQL was partially replaced with a custom Estonian-language admin interface, whose development is still in progress.

Next.js 15 · Cloudflare Workers · D1 + Drizzle · Durable Objects · eID

github.com/siimliimand/erametsad

My Stash

AI + PWA

WHY

My Stash came from a frustration: microlearning dies under the weight of accounts and tracking — people want small daily ideas, not another signup. A Cloudflare Worker delivers a daily batch of AI-generated idea cards that accumulate on your device and work offline: no accounts, no tracking.

FOR WHOM

Busy, privacy-conscious people who want bite-sized learning.

TECH

The application uses a Next.js static export and runs on Cloudflare Pages, so there is no need for a dedicated server. All user data is stored locally on the device in `localStorage` and validated with Zod. The absence of accounts, tracking and central data storage is therefore guaranteed by the architecture itself, not merely by a privacy promise.

An optional Cloudflare Worker uses a D1 database, KV and cron scheduling to generate the daily batch of AI cards server-side via OpenRouter. A per-day run lock guarantees idempotency and prevents the same batch from being generated twice. If the Worker or the AI service is unavailable, the application keeps working in a fully local mode without losing its core functionality.

Next.js · TypeScript · PWA · Cloudflare Workers

my-stash.pages.dev · github.com/siimliimand/my-stash

TabSpace — developers juggle many projects; every new tab becomes a workspace for links, tasks and notes Chrome MV3 Zustand Cloudflare

Web Downloader — saving web content and media in one click, without screenshots and re-downloading React TypeScript Vite

Storefront Zero — slow checkouts cost e-shops sales; a lean WooCommerce theme keeps them fast PHP 8.3 Flight PHP HTMX

NovaTools SEO — WordPress sites need meta, schema, sitemaps and redirects; one plugin covers them PHP React 18

NovaTools Polyglot — selling into several language markets from one WordPress install PHP 8.1 WooCommerce

PDF Editor — quick PDF fixes without uploading documents to unknown web services Python FastAPI

TECHNICAL SKILLS

Languages	TypeScript, JavaScript, Python, PHP, SQL, HTML, CSS
Frontend	Next.js, React, Tailwind CSS, Radix UI, Framer Motion
Backend	NestJS, FastAPI, Node.js, Payload CMS, WordPress / WooCommerce
AI / LLM	OpenCode (daily driver), OpenAI, Anthropic, OpenRouter, agent orchestration, AGENTS.md / OpenSpec, human-in-the-loop pipelines
Data & infrastructure	PostgreSQL, Drizzle, Prisma, Redis, BullMQ, Cloudflare (Workers / Pages / D1 / Durable Objects), Linux, Bash, Docker, Kubernetes, GitHub Actions
Quality	Vitest, Jest, Playwright, Pest, PHPStan, golden-dataset & compliance testing
Also worked with	SAP ABAP, gRPC, HTMX, Flight PHP, Zustand, Vite, Podman

EDUCATION

Tallinn University of Technology (TalTech)
BSc in Business and Information Technology · 2008 – 2012

Tallinn Polytechnic
Heat and Industrial Automation · 2004 – 2008

LANGUAGES

Estonian
Native

English
Listening C1, reading C1, speaking B1, writing B1

CERTIFICATIONS

AWS Certified Cloud Practitioner
In progress · 2026 — cloud-fundamentals certification, first step on the AWS architect track

Linux Professional Certification (LPI / Linux Foundation track)
Planned · 2026 — server Linux administration, formalizing daily-driver Linux, Bash and Docker/Podman skills

PREFERENCES

Desired roles

IT architect, systems analyst, software developer

Employment type

Permanent or contract · full-time, part-time or freelance

Location

Tallinn or remote