

# Oskari Pessinen

oskari.pessinen@gmail.com | +358 447802010

[linkedin](#) | [github](#) | [portfolio](#)

## EDUCATION

### M.Sc. (Tech) in Computer Science

2023 - 2028

University of Turku

## SKILLS

**Languages** Python | Java | JavaScript | TypeScript | SQL

**Technologies** React.js | React Native | Next.js | Node.js | PostgreSQL

## PROJECTS

### Since Ai Hackathon - Top 5

#### AI-Powered Funding Advisor

- The application analyzes basic company information and suggests suitable funding instruments (such as grants, loans, and equity investments) based on the company's size, stage, and funding needs.
- Technologies: Next.js 16 (App Router), Prisma + PostgreSQL for the database, and Supabase Auth (Email OTP) for authentication.
- Utilizes the Qwen 3 80B Instruct language model via an OpenAI-compatible API to generate tailored and justified funding recommendations.

### Junction 2025 Hackathon - 2nd Place (Espoo Challenge)

#### AI-Based Business Advisor

- Developed an AI-assisted advisory tool for the City of Espoo to help entrepreneurs prepare before meeting with business advisors.
- Responsible for UI design and full stack development (Next.js, Vercel). Our project placed second in the challenge.

### Compkit

#### AI for Component Generation

- An AI-powered platform for generating, publishing, and installing UI component libraries.
- Frontend: React + TypeScript, Vite, Tailwind, shadcn/ui; Backend: Node.js + Express; AI via OpenRouter and authentication/database via Supabase.
- Generates ready-made components with previews from descriptions. Libraries can be published instantly with a unique identifier.
- CLI: `npx compkit install <id>` — downloads components to the `src/components/ui` folder, installs dependencies, and configures Tailwind automatically.
- Structure: Monorepo (apps/frontend, apps/backend, packages/types).

### Navi

#### AI CLI Assistant

- An interactive command-line tool that provides autocomplete, AI suggestions, and allows inserting commands directly into the Zsh prompt.
- User interface built with Python using the `prompt_toolkit` library (autocomplete, dynamic suggestion updates).
- Integrated into the Zsh environment: the selected command is written to a temporary file (NAVI\_OUT) and fed into the command line buffer using the `print -z` command.
- Focused on a smooth CLI experience: debouncing AI queries, instant autocomplete, and a visual bottom toolbar. Developed with Python (3.12), tested in a Linux environment.

## WORK EXPERIENCE

### ISS

06/2025 - 08/2025

Cleaner

Summer Job

### ISS

06/2023 - 08/2023

Cleaner

Summer Job

### Burger King

06/2022 - 08/2022

Fast food worker

Summer Job