Patryk Jaseniuk

Full Stack TypeScript Developer

React · Next.js · Astro · PostgreSQL/Supabase · testing & CI/CD

Wrocław, Poland · remote or hybrid
patryk.jaseniuk@gmail.com
github.com/PatrykJaseniuk

PROFILE

Full Stack TypeScript Developer: React, Next.js, Astro and PostgreSQL/Supabase. Favourite paradigm: functional programming — immutability, function composition and controlled side effects. Since 2022 I have been building end-to-end applications with tests and CI/CD; with LLM agents, I work spec-driven and verify changes through typing and tests.

SELECTED PERSONAL PROJECTS

parys — gym website and print materials generated from a single source

2026 · Commercial project · Astro 5 · TypeScript · Tailwind CSS 4 · Pages CMS · GitHub Actions · site · printables · Code

- Defined a structured business model as the single source of truth: zones, memberships, classes, trainers and services stored as JSON; website and print materials are generated from it.
- Content edited via Pages CMS — a git- and GitHub Actions-based CMS.
- No server: fully static, lightweight and fast site; zero running costs; CI validation. Lighthouse: 100/100/100/100.

ts-fullstack-template — fullstack app in a business domain: rental-property management

2026 · React 18 · TypeScript · TanStack Router/Query · PostgreSQL RLS · Vitest · Playwright · Live · Code

- CI/CD: GitHub Actions on push and PR to `main`, 7 jobs (lint, types, `tsc`, Vitest, deploy) — frontend on GitHub Pages, backend on Supabase; local loop via `Makefile`.
- Backend: Supabase (Postgres + PostgREST): 7 tables, 12 foreign keys, 7 enum types, 12 CHECK/UNIQUE constraints, 13 functions, 15 triggers, 30 RLS policies. `user_role` claim from `custom_access_token_hook`; admin/landlord/tenant roles; 10 `security_invoker` views (balances and statements on window functions); typed API `createClient<Database>`.
- Frontend: React 18 + Vite, TanStack Router (hash history) and Query; components divided into dependency layers. List/CRUD hooks: 300 ms debounce, sorting, PostgREST pagination (`.range()`, 20 rows); cache `staleTime` 30 s, invalidation after mutations.
- LLM: rules from `AGENTS.md` enforced by ESLint (`functional/*`, no `if` / `switch` / `enum` / `let` / `throw`, import boundaries, `no-floating-promises`) and strict TS (`noUncheckedIndexedAccess`).
- 245 Vitest tests (29 files), 8 E2E, 13 pgTAP, 15 Storybook (21 `play`, a11y).

pkg-builder — an LLM agent as a package manager building from source

2026 · AGENTS.md · LLM agents · spec-driven development · DAG · AGENTS.md

- The agent is defined entirely in `AGENTS.md`.
- The user submits a request to install given software, and the agent carries out the task — from cloning the git repo, through building and recursively installing dependencies, to a launch script that runs the software with the right configuration.
- The whole installation is split into layers; the agent fills successive layers iteratively until done, generates reports and records dependencies as knowledge for the future.
- Building packages from source gives enormous room for modification.

sailing-sim — physics engine and 2D sailing simulator

2022–2024 · math · physics · TypeScript · Next.js · Three.js · Jest · Code · Live

- Implemented a custom soft-body dynamics model: springs, aerodynamic and hydrodynamic forces, rotational motion and a fixed integration step. Built the view with Three.js and collision detection with detect-collisions/SAT.
- The model is separated from the view; 13 Jest suites and 13 chapters of technical documentation.

SKILLS

Frontend	React 18, Next.js, Astro 5, TanStack Router and Query, Tailwind CSS, HTML/CSS, Three.js, Storybook
Backend & data	Node.js, PostgreSQL, Supabase, REST API, PostgREST, RLS, views, triggers, pgTAP
Testing & CI/CD	Vitest, Jest, React Testing Library, Playwright, Storybook a11y, GitHub Actions, Lighthouse
LLM agents	spec-driven development, AGENTS.md, MCP, SKILL.md, local LLM, llama.cpp
Formal languages & tools	TypeScript, JavaScript (ES2022+), SQL, Bash, Python, C, C++, Java · Git, Docker, Vite, ESLint, Linux, Make

Natural languages: Polish — native · English — B2 (comfortable with technical documentation)

EDUCATION

Computer Science — University of Applied Sciences in Nysa (PANS) · *2019–2023*