

RYAN WILSON

Full-Stack Web Developer

Des Moines, IA

ryan@rywils.com | rywils.com | github.com/rywils | linkedin.com/in/rywils

SUMMARY

Full-stack web developer & UI/UX designer with nearly a decade in the industry, including two years freelancing and building websites that shipped production applications for small business clients focusing on frontend, backend, databases, and cloud deployment. Fluent across the modern JavaScript/TypeScript ecosystem (React, Next.js, Node.js) as well as Laravel/PHP, CMS such as Wordpress, Joomla, Drupal, Headless Wordpress, Payload, Sanity, and Contentful. My most recent work extends into AI-integrated web applications and agentic frameworks. Regardless of the stack, I bring a security-conscious mindset and test every project I build.

EDUCATION

Western Governors University

B.S. Cybersecurity and Information Assurance

In Progress – Graduating 2028

Des Moines Area Community College

A.A.S. Information Technology – Network Administration

Graduated May 2016

TECHNICAL SKILLS

Web & Application Development: TypeScript/Javascript, CSS, Sass, Tailwind CSS, React, Next.js, Nest.js, Astro, Svelte, Vue/Nuxt, Node.js, Tanstack, HTMX, Alpine, .NET, Laravel, PHP, REST APIs, GraphQL, tRPC, gRPC, Mobile App Development

Frontend: React, React Native, Flutter, Next.js, Svelte, Vue/Nuxt, Astro, Three.js, responsive & component-based UI

Databases & ORMs: PostgreSQL, MongoDB, MariaDB, Prisma ORM, Supabase, Redis

Cloud & DevOps: AWS, Azure, GCP, Vercel, Netlify, Heroku, DigitalOcean, Docker, Kubernetes, Terraform, Ansible, CI/CD

CMS & Platforms: Headless WordPress, Sanity.io, Strapi, Payload, Contentful, Joomla, Drupal

No-code/Low-code: Wix, Framer, Webflow, Wordpress Elementor, Builder.io, Plasmic

AI & Agent Systems: LLM API integration, agentic AI development (TypeScript/Python), LangChain, LangGraph, LangSmith, PyTorch, prompt engineering

Other Languages: Rust, Go, C#, Python, C

Security-Aware Development: OWASP, authentication & IAM patterns, secret management, code analysis, web application security

Tools: Adobe Illustrator, Photoshop, Lightroom, Affinity, Blender, Figma

PROFESSIONAL EXPERIENCE

Freelance Web Developer | Self-employed

Nov 2024 – Mar 2026

- Provided full-stack web development and consulting services for small business clients, specializing in modern application development, branding, and tailored websites
- Built responsive web applications using Next.js, React Native, Nuxt.js, SvelteKit, and Laravel
- Engineered type-safe APIs with Node.js/Express, tRPC, GraphQL, gRPC, and Supabase
- Managed databases including PostgreSQL, MongoDB, and MariaDB using Prisma ORM
- Deployed applications on AWS (RDS, Lightsail, Cognito), Vercel, Heroku, Linode, Netlify, and DigitalOcean
- Implemented authentication, CI/CD, CDN distribution, and real-time caching with Redis
- Integrated project-specific CMS systems including headless WordPress, Sanity.io, Strapi, and Contentful
- Owned the full project lifecycle: client prospecting, onboarding, architecture, deployment, and ongoing support

Founding Engineer | AGIS Global

Mar 2026 – Aug 2026 (company paused operations)

- Built the company's initial technology stack solo as founding engineer, including internal platforms and production software in Rust, Go, TypeScript, and C#
- Built a custom Rust orchestration layer for zero-copy IPC, removing Python GIL bottlenecks from high-throughput AI request handling
- Implemented Infrastructure as Code (Ansible, Terraform) backed by a self-hosted observability and CI/CD stack (Grafana, Prometheus, Loki, Jaeger, GitLab)

Additional Experience

Systems Engineer, DH Pace (Jul 2023 – Aug 2024) — Administered hybrid cloud and VMware infrastructure and enterprise network/security systems (AWS, Cisco Meraki, Fortinet).

Consultant, Self-employed (Mar 2022 – Jun 2023) — Delivered IT consulting, cloud administration (Azure/AWS), and network engineering for small-to-mid-sized businesses.

Systems Administrator, BlueAlly (Oct 2019 – Dec 2021) — Managed network security and infrastructure across multiple client environments; led a year-long incident remediation effort.

Technical Implementation Specialist, Casey's General Stores (Jul 2019 – Oct 2019) — Supported rollout of a new online ordering system across Midwest retail locations.

Datacenter Technician, Microsoft (Nov 2017 – Dec 2018) — Diagnosed hardware failures and supported network configuration across datacenter clusters.

PROJECTS

A few projects worth naming, some of which are outside the scope of web development, but apart of my diverse background.

(Currently in production) Unnamed - 2026 - 2026

A multi-tenant SaaS platform for independent restaurants and businesses in the food industry that combines a self-hosted CMS and centralized management platform, a customer-facing storefront, POS-synced menus and inventory, vendor tracking, stock management, sales analytics, customer rewards/loyalty program, online ordering, built-in hiring and employee onboarding pipeline, AI-assisted marketing automation, and a per-business permissions and feature-switch system, each feature switchable per business, and each feature integrated as a simple adapter. Originally, the client came to me for a website redesign. After learning the scope and direction the manager and owners wanted to go with it, I knew the solution was much larger than what they had anticipated. With the original concept, the client mentioned they may be moving away from their current POS system to a new one, so I wanted to build a modular system to where simply writing an adapter is all that is needed to integrate a brand new technology into the ecosystem to pull directly from the database all menu items, loyalty information, and other relevant information. This project not only tested my frontend design abilities and backend development knowledge, but it forced me to pay close attention to the way data is exchanged and handled to reduce risk, data manipulation, or parameter tampering and build the most secure solution I possibly can. The frontend and the decoupled CMS were both built on Next.js, using almost entirely Typescript throughout the project, with Python where it was suited best. To confirm this and enhance development with modern tools, I utilized Claude Fable to write a total of 606 test cases across 132 test files throughout various states of the project, review my source code, make suggestions, improvements, refactors and spot mistakes that mitigated multiple critical vulnerabilities. You can read the case study for this by [clicking this link](#).

- Designed template storefront driven by the CMS, with catalog-aware menu sections, locations, hours, reviews and a careers page.
- Block library of designed sections with live preview at mobile, tablet and desktop.
- Drag-and-drop page canvas embedded in the admin, writing the same page data as the block editor.
- Pre-built themes selectable per business, each with its own design tokens.
- Business-scoped website settings and domains so every tenant runs its own site from the same platform.
- Menu blurbs and section descriptions generated from the published catalog when left blank.
- Every user scoped to one or more businesses and locations, with a business-scoped tech admin role.
- Business-defined roles with a capability matrix, per-user overrides, and a Permissions admin screen.
- Field-level masking so staff see a single order's contact but never browse customer lists.
- Per-business switches for every module: online ordering, loyalty, inventory and vendors, website, chatbot, AI and automation, employee discount, hiring, onboarding, marketplace.
- Categories, items, modifiers and images synced to the POS through an outbox with retries and health reporting.
- Inventory levels, movements, vendors, purchase orders and reorder alerts with vendor ordering links.
- Bring-your-own POS with per-business connections and a catalog mode of platform-led, POS-led or payments-only.

- Additional POS adapters and vendor API ordering behind the same adapter interfaces.
- Customer accounts, orders with pickup and fulfillment settings, loyalty accounts and reward tiers.
- Online checkout with payment authorization.
- Employee discount applied in-store through the POS and online for linked staff accounts.
- Live dashboard widgets for sales, orders, loyalty, stock, revenue, locations and integration health, with an editable draggable layout.
- Public careers page with an application form, optional resume and cover letter upload, and resume auto-fill.
- Applicant pipeline with stages, bulk stage changes, unread badges, and interview, offer and hiring notes.
- Onboarding role with a welcome dashboard, tax and state forms, handbook, policies, optional training, and typed or drawn signature.
- Employees directory with onboarded status and links to signed documents.
- Email delivery for application receipts and onboarding reminders.
- AI asset store, generated-content records with approval states, and scheduled automations.
- Social posting adapters and an internal knowledge-base assistant.
- Isolated AI website chatbot reading only the published site snapshot.
- Layered upload validation, signed cross-origin tokens, restricted framing, rate limits and audit events.
- Idempotent migrations, one-script local setup, shared generated types, and test suites on both apps.
- Production deployment, custom domains per tenant, and secrets management.

Stygian UI 2026 - Ongoing

A react UI component library based on shadcn/radix, three.js, GSAP, and framer motion

Rusticize - Online platform to learn Rust 2025 - Ongoing

- Built a platform to learn the Rust programming language and integrate the official book.
- Pulls official book from the Github repo, parses & styles it and renders it within a canvas.
- Canvas has the ability to read as intended, or to freefloat any highlighted element and move text around the screen.
- Sticky note system, where text is chosen to be hidden and a indicator above the last word of the selected text indicates there's a note, or, render the note on the page either married to the textblock of the parent, or set to freefloat and can drag around the canvas.
- Highlighting, Underlining, bookmarking system
- Rust Playground integration, and building an interactive course with in-browser terminal simulation to pair with the platform.
- Designed to co-exist with already made solutions such as Rustlings

Skevor - AI Orchestration Framework 2026 – Ongoing

TypeScript framework for building multi-tenant AI agent products (Next.js apps or terminal chat products) from a single config file, without hand-rolling the same infrastructure for every client build.

- Architected tenant isolation by default — every data store (conversation memory, credentials, rate limits, assets) scoped by tenantId at the query level
- Built a connector library (Gmail, Stripe, Slack, Zoom, Microsoft Teams, and more) behind a single factory pattern for fast extension
- Implemented per-tenant RAG knowledge bases for grounded, retrieval-augmented answers from client-provided docs
- Built 'create-skevor-app,' a CLI scaffolding tool (inspired by create-react-app) that generates a full app from one config file in a single command
- Supported multi-provider chat models (Anthropic, OpenAI, Google), configurable per tenant

BitSentry - Security Assessment Platform 2025 – Ongoing

Built a modular CLI-based security assessment suite (TypeScript) covering attack-surface discovery, web vulnerability scanning, and multi-format reporting (JSON, Markdown, PDF, HTML)

- Designed plugin-based scanning pipelines and CI/CD- and Docker-compatible execution paths for portable automation

CERTIFICATIONS

ISC2 Certified in Cybersecurity (CC) • Google Cybersecurity Professional • Datacamp AI Fundamentals • CompTIA A+