

# AYUSH JUVEKAR

ayushjuvekar@gmail.com | +1 (906) 299-4314 | [LinkedIn](#) | [GitHub](#) | [Portfolio](#)

Full-Stack Engineer

## PROFILE

---

Full-stack engineer building production web products with Next.js, TypeScript, databases, auth, payments, analytics, and deployment ownership.

## EDUCATION

---

**Michigan Technological University**, Houghton, MI  
*M.S. Computer Science*

**CGPA: 3.9/4.0**  
Graduated Apr 24, 2026

**Pune Institute of Computer Technology**, Pune, India  
*B.E. Computer Engineering*

**GPA: 8.14/10**  
Jun 2024

## EXPERIENCE

---

**Freelance Sole Developer** | The Mind Point - Mental Health Education Platform Aug 2025 - Present  
Pune, India

- Built a production course-commerce platform from scratch for a mental health education business, supporting real paid enrollments through Razorpay and Clerk-authenticated accounts.
- Owned storefront, cart, checkout, enrollment, admin tracking, analytics, email confirmations, and Vercel deployment across the full product workflow.

**Teaching Assistant - Physics Department** | Michigan Technological University Sep 2024 - Apr 24, 2026  
Houghton, MI

- Supervised and graded physics labs for 100+ students across four semesters, guiding experiments, explaining concepts, and providing clear feedback across five lab sections per semester.

## SELECTED PROJECTS

---

**YayCamp - Campground Discovery Platform** | *Next.js, TypeScript, Prisma, PostgreSQL, Clerk, React Leaflet* | [Live](#) | [GitHub](#)

- Built and deployed a full-stack campground discovery app with Next.js, TypeScript, Clerk authentication, Prisma, PostgreSQL, and React Leaflet maps.
- Connected searchable listings, relational data models, auth flows, map-based discovery, and public deployment into one end-to-end personal product.

**Ranking App - Preference Ranking Tool** | *TypeScript, React, Next.js, Redis* | [Live](#) | [GitHub](#)

- Designed and built a novel preference-ranking app after finding bracket tools ignored choice confidence; used decision latency to weight close vs. decisive matchups.
- Persisted user-defined item sets with Redis and normalized final rankings to a 100-point score for clearer comparison.

**MonkeyLang-CS - Interpreter Port** | *C#, .NET, TDD* | [GitHub](#)

- Ported Thorsten Ball's Monkey interpreter from Go to idiomatic C#, implementing lexer, parser, AST, evaluator, object system, REPL, functions, closures, arrays, hashes, and built-ins.
- Followed the book's test-first workflow to practice interpreter internals, language runtime design, and disciplined implementation across a translated architecture.

## TECHNICAL SKILLS

---

<b>Languages</b>	TypeScript, Python, Java, C#
<b>Frontend</b>	React, Next.js, Tailwind CSS, shadcn/ui
<b>Backend</b>	Node.js, Express.js, REST APIs, Prisma ORM
<b>Databases</b>	PostgreSQL, Redis, MongoDB
<b>Product Integrations</b>	Clerk, Razorpay, Resend, PostHog
<b>DevOps</b>	Docker, Git, Vercel, Linux