

Arjun Singh

Full Stack Developer Noida, India

arjunsingh0335639@gmail.com

+91-8744018060 aarjun8060.vercel.app | github.com/aarjun8060 | linkedin.com/in/aarjun8060

Professional Summary

Full Stack Developer who ships products in public: two published Shopify App Store apps, a VS Code Marketplace extension, and Split2Ship, the merchant payment app I lead at DesignOmate. Works across React, Next.js, Remix, and TypeScript with Node.js, Express, PostgreSQL, and MongoDB, covering REST API design, JWT/RBAC authorization, payment integrations, and AWS/Vercel deployments behind CI/CD. Owns features from schema design through production support.

Technical Skills

Frontend: React, Next.js, Remix, TypeScript, JavaScript (ES6+), HTML5, CSS3, Tailwind CSS, Shadcn UI, MUI, Redux Toolkit, Zustand, Recoil

Backend & Data: Node.js, Express.js, REST APIs, WebSockets, Socket.io, PostgreSQL, MongoDB, Prisma ORM, Redis, SQL, NoSQL

Cloud & DevOps: AWS (EC2, S3), Vercel, Docker, Linux, Git, GitHub, CI/CD (GitHub Actions), JWT, Role-Based Access Control (RBAC)

Shopify & Payments: Polaris, App Bridge, Theme App Extensions, Admin GraphQL API, Billing API; Razorpay, Axis Bank API, Fino Bank API

Other: VS Code Extension API, Solidity, OpenZeppelin, Hardhat, Ethers.js, Web3.js

Work Experience

Full Stack Developer | DesignOmate – Noida, India Dec 2025 – Present

- Developed verification-related pop-up flows on product pages using React, Remix, and TypeScript to streamline user verification and improve checkout completion rates
- Designed and integrated multiple payment options (full, partial, COD) into the Split2Ship Shopify app using Remix, Prisma, and PostgreSQL, owning each feature from schema design through independent production deployment
- Drove 50% customer growth by owning the product roadmap and resolving customer-reported production issues end-to-end as the primary point of contact

Full Stack Developer | Lesus Innovations – Delhi, India May 2025 – Sep 2025

- Developed and maintained full-stack features for ZapNow and ZapHub, serving 500+ daily active users; built scalable backend APIs and admin dashboards with 99.8% uptime
- Integrated secure payment workflows in Node.js and Express using Axis and Fino Bank APIs, processing 100+ monthly transactions at a 99.9% success rate; improved scalability 25% and cut production bugs 30%

Software Engineer | TechPyro – Haridwar, India Jun 2023 – Nov 2023

- Built React and Next.js interfaces with MUI and Shadcn UI, and scaled the Node.js services behind them to support 10,000+ concurrent users
- Reduced page load time by 45% through code splitting and lazy loading; optimized Node.js REST APIs to cut response time by 40% and add 2,000 requests/minute of throughput

Projects

WishVault – Published Shopify App Store wishlist app

- Designed, built, and published a save-for-later wishlist app to the Shopify App Store; delivers the wishlist drawer through a Theme App Extension so it installs on any merchant theme without code changes, with guest wishlists that persist saves without an account
- Shipped a merchant analytics dashboard surfacing most-wishlisted products as buying-intent signal, plus automated reminder emails that bring shoppers back to saved items
- Implemented credit-metered subscription billing across three tiers (free, \$5, \$10, with annual plans) on the Shopify Billing API, gating customization and email features per plan
- **Tech:** Remix, TypeScript, Prisma, PostgreSQL, Polaris, Theme App Extensions, Admin GraphQL API, Billing API

EnvCompare – Published VS Code Marketplace extension

- Published a developer tool that keeps environment-variable keys in sync by diffing keys across `.env` and every `.env.*` variant in a workspace, reporting exactly which keys each file is missing
- Built an inline completion provider that suggests missing keys as Tab-to-accept snippets, backed by debounced file watchers that warn the moment a workspace goes out of sync; compares key names only and never transmits values, so secrets stay on the local filesystem
- **Tech:** TypeScript, VS Code Extension API, Node.js

Notion CMS Blog – Notion-backed publishing platform

- Built a blogging platform that uses a Notion database as its CMS: posts are authored in Notion and go live the moment their status flips to Published, with the filter applied server-side in the Notion API query so drafts never reach the client
- Statically generated listing and post pages with ISR revalidation, so published edits appear without a redeploy while pages keep serving from cache and requests stay inside Notion's API rate limits
- Mapped Notion's nested block tree and loosely typed database properties into typed React components to render rich text, headings, and embeds; supports a post-type taxonomy across trending and recent listings
- **Tech:** Next.js, TypeScript, Notion API, ISR, Vercel

Jotion – Real-time collaborative Notion clone

- Built real-time collaborative document editing on Convex live queries, supporting 100+ concurrent editors with changes propagating to every client without a refresh
- Implemented a nested document hierarchy with dynamic routing and Clerk-backed authentication, scoping each user's workspace tree so every document is individually addressable
- **Tech:** Next.js 13, TypeScript, Clerk, Convex

Additional Projects

- **PulseBar Announcement Bar:** No-code Shopify app for customizable store announcement bars, integrated with Shopify Emails, Bravo, and Klaviyo. **Tech:** React Router v7, PostgreSQL, Upstash Redis
- **GangaClean:** Volunteer-management dApp with three on-chain roles via OpenZeppelin AccessControl and a pull-payment reward flow. **Tech:** Solidity, OpenZeppelin, Hardhat, Ethers.js

Open Source

- **tweakcn:** Implemented enhancements to a Tailwind theme generator (e.g., PR #113 merged: color palette & picker UI); reduced configuration time by 50% for 200+ developers
- **Hacktoberfest 2023:** 5 merged PRs across repositories; improved code quality, documentation, and added 8 features

Education

B.Tech in Computer Science
Gurukula Kangri University, Haridwar

2021 – 2025 | GPA: 8.1