Shadhujan Jeyachandran

Full Stack Software Engineer

ASP.NET Core | Next.js | TypeScript | SQL Server | Secure API Architecture | Metadata-Driven Platforms

Jeya38shadhujan@gmail.com | +94754262354 | linkedin.com/in/shadhujan | github.com/Shadhujan | shadhujan.dev | shadhujan.medium.com

Full Stack Software Engineer with **1+** years of experience building secure, scalable systems across Marketing using **ASP.NET Core, C#, Next.js, TypeScript, and SQL Server**. Experienced in developing **high-performance metadata-driven platforms**, **WYSIWYG** configuration interfaces, and advanced admin **UX frameworks**, while implementing **secure API architectures** with **HMAC-signed webhooks and MFA**. Adept at collaborating with cross-functional **Agile teams** to deliver stable, automated, and observable production releases.

Experience

Associate Software Engineer
M Data Zone SA (Pvt) Ltd.
Colombo, Sri Lanka • On-site • US Hours
September 2025 - Present

- Architected and developed a metadata-driven white-label Customer Engagement Platform using **Next.js, ASP.NET Core, and SQL Server**, enabling configurable campaign management, deep-link orchestration, analytics, and role-based administration.
- Designed and built a **metadata-driven CRUD engine** where backend-defined schemas dynamically generated **20+ admin screens** through a shared Next.js route and component architecture, reducing hard-coded UI development.
- Developed a **permission-aware shared form engine** and advanced admin UX framework powering WYSIWYG configuration for **Workflow Builder, Push, In-App Messaging, and Deep Link** modules.
- Built **50+ ASP.NET Core REST API endpoints** for EnvisionArc using structured controller, DTO, and repository layers backed by **SQL Server stored procedures**, establishing the core backend foundation for authentication, configuration, deeplinks, logging, and protected workflows.
- Designed and implemented secure backend architecture covering **JWT, cookie-based sessions, MFA/2FA, QR-based OTP onboarding, password hashing, CSRF-aware flows, and HMAC-signed webhooks** across frontend-backend integration points.
- Developed and integrated multiple **.NET microservices** within the Comm Engine platform, including the **Push API, Push Notification Worker, Event API, and Deep Link API**, connected through a **RabbitMQ-based asynchronous queue architecture** with **dead-letter queue (DLQ)** support and producer-consumer payload alignment.
- Built a custom **Serilog database sink** and improved structured logging and observability across services to support centralized troubleshooting, queue visibility, and production diagnostics.
- Supported multi-environment delivery across **DEV, STG, and UAT** using **Jenkins, Docker, and environment-driven configuration**, contributing to deployment validation, release stability, and secrets/configuration improvements

Intern Software Engineer
M Data Zone SA (Pvt) Ltd.
Colombo, Sri Lanka • On-site • US Hours
April 2025 - September 2025

- Authored 100+ T-SQL stored procedures with constraints and DB Mail notifications, improving data quality signals & reducing rework.
- Resolved backend and database issues using **T-SQL, constraints, and defensive parsing**, helping reduce operational downtime ~20%.
- Built 3 .NET Core CLI apps with Selenium, enabling group / single test runs, screenshots and logs, cutting repeat manual effort by ~70%.
- Automated 1000+ end-to-end UI scenarios with resilient selectors; enabled overnight suites and faster triage via evidence-rich artifacts.
- Performed manual QA across 6+ projects (feature/regression), improving defect detection and helping reduce escaped bugs by ~30%.
- Documented software / database designs for 3+ areas, accelerating onboarding and cross-team discussions.
- Operated as a cross-timezone partner with teams in Canada/US/Sri Lanka, aligning on release plans, QA handoffs, and DB changes.

Education

BSc (Hons) Computer Science — **First Class Honours**
University of Bedfordshire (UK) — *Jan 2025 – Oct 2025*

Higher Diploma in Information Technology — **GPA 3.3**
SLIIT City Uni — *2023 - 2024*

Skills

Languages & Fundamentals:	C, C++, C# (.NET Core), TypeScript (TS) / JavaScript (JS) (ES6+),Python, SQL (T-SQL, PostgreSQL), PHP, Java, Bash / PowerShell
Frontend Architecture:	Next.js (App Router), React, State Management (Zustand, Redux), UI Libraries (Tailwind CSS, MUI, Shadcn UI), Visuals & Animation (Framer Motion, Three.js)
Backend & System Design:	Node.js (Express), ASP.NET Core, FastAPI, Distributed Systems (RabbitMQ), Security (HMAC-signed Webhooks, OAuth2, 2FA/TOTP), RESTful APIs
Database Engineering:	MS SQL Server (Stored Procedures), PostgreSQL, MongoDB, MySQL, Caching & Persistence (Redis, SQLite, Firebase)
DevOps & Automation:	CI/CD (Jenkins, Vercel, Render, Netlify), Docker, Selenium Automation, Observability (Serilog, Database-backed Sinks)
Productivity & Design:	JIRA, Zendesk, Trello, Open Project, Android Studio, DBeaver, Adobe Products
Data Science & AI:	OpenAI API, PyTorch (NLP), Pandas, NumPy, Chart.js

Personal Projects

FocusBoost • github.com/Shadhujan/FocusBoost

FocusBoost is a privacy-first, web-based attention monitoring system designed to help children stay engaged during study sessions without compromising their personal data. At its core, the system uses facial expression analysis to passively track attention levels in real time, feeding that data through a machine learning pipeline to detect disengagement and trigger adaptive interventions before focus is lost. The backend is built on FastAPI with a modular architecture separating user accounts, session management, ML processing, quiz generation, and attention tracking into clearly bounded concerns. A parent dashboard provides visibility into session history and cognitive engagement trends, backed by Firebase for real-time data and the Gemini API for intelligent analysis. FocusBoost was built with the DAISEE dataset as a training foundation and with a deliberate focus on keeping sensitive facial data local and processing it responsibly, no unnecessary storage, no cloud-side video.

• *React, TypeScript, Python, FastAPI, Firebase, Gemini API, PyTorch, DAISEE*

3D Interactive Dojo • https://3d-dojo-shad.vercel.app • github.com/Shadhujan/3D-Interactive-Dojo

3D Interactive Dojo is an immersive, first-person web experience built as a foundation for interactive 3D environments on the web. The project began as a personal portfolio concept: a virtual space users could physically explore to discover content, and it later evolved into a structured template for browser-based 3D applications. Movement is handled through WASD controls with mouse-look via pointer lock, giving the experience a familiar, game-like feel without requiring any plugins or native installs. The scene is organized into modular interior and exterior sections with dynamic lighting, keeping the structure readable and straightforward to extend. A React and Tailwind UI layer sits above the 3D canvas for overlays and interface elements. GLB Inspector & Viewer was built as a companion utility specifically for this project, providing a dedicated space to inspect, validate, and debug 3D models before bringing them into the scene. Future development will focus on interactive objects, portfolio content embedded within the 3D space, collision handling, and asset optimization.

• *React, TypeScript, Three.js, React Three Fiber, react-three/drei, Vite, Tailwind CSS*

GLB Inspector & Viewer • glb-viewer-shad.vercel.app • github.com/Shadhujan/GLB-Viewer

GLB Inspector & Viewer is a professional-grade web tool for loading, inspecting, and debugging GLB (glTF 2.0 binary) 3D models directly in the browser. It was built as a dedicated utility to support 3D_Interactive_Dojo, a purpose-built companion that validates and introspects models before they are integrated into a larger 3D environment. The viewer supports drag-and-drop file loading, interactive orbit controls, and studio-quality lighting out of the box. Its inspector panel surfaces granular model statistics, including vertex and triangle counts, mesh and material breakdowns, texture listings, animation clips with durations, and a full scene hierarchy tree. A toggleable hover-inspect mode lets users identify individual mesh components by name directly within the 3D scene. The interface is dark-themed and modelled after industry-standard 3D tooling, prioritizing clarity and utility over decoration.

• *Next.js, TypeScript, Three.js, React Three Fiber, react-three/drei, Tailwind CSS*

Bunona • cis-bunona-game.netlify.app • github.com/Shadhujan/Bunona

Bunona is a browser-based number puzzle game that integrates the external Banana API as its puzzle data source, built to explore and demonstrate software design principles in a practical, interactive context. The game is structured around low coupling and high cohesion the timer, scoring, game logic, UI, and API communication layers are fully independent modules, making each concern easy to modify or scale without rippling side effects. Gameplay is driven by an event-driven architecture: losing a life, completing a daily challenge, and hitting a countdown all fire discrete events that the system responds to independently. Player identity is built up progressively through a lives system, achievement tracking, daily streaks, and personal leaderboards backed by Supabase and PostgreSQL, giving each player a persistent, evolving profile rather than a stateless session.

• *React, TypeScript, PostgreSQL, Supabase, Tailwind CSS, Banana API*

Explore more projects: shadhujan.dev/projects

References:

Pasindu Akalpa
Tech Lead
MdataZone
+94 76 459 7776
Admin@akalpa.com