PUSPENDRA CHAWLA

puspendrachawlax@gmail.com | +91 9828219407 | GitHub | LinkedIn | Rajasthan, India

SUMMARY

Backend software engineer (fresher, 2025) specializing in Go, Python, and TypeScript, with hands-on experience building production-grade distributed systems, real-time applications, and secure APIs. Developed a Redis-backed webhook delivery platform sustaining ~600 req/s, a hardened authentication API, and a real-time multiplayer application — demonstrating strong command of concurrency, event-driven architecture, and scalable backend design.

TECHNICAL SKILLS

Languages: Go (primary), Python, JavaScript, TypeScript, Bash, SQL, C++
Backend: REST API design, Event-driven architecture, Concurrency, Goroutines, Microservices
Databases: PostgreSQL, Redis, MongoDB, SQLite
DevOps: Docker, Docker Compose, GitHub Actions, Terraform
Cloud: AWS (EC2, S3, ECS, Lambda, VPC, IAM, EKS)
Other: Linux, NGINX, CI/CD pipelines, JWT, OAuth, System design, Unit testing

PROJECTS

Webhook Delivery Platform *Go, Gin, PostgreSQL, Redis, Docker*
GitHub

- Built a production-grade webhook delivery system inspired by Svix; ingestion API sustains ~600 req/s under load with avg response latency under 15ms.
- Designed a durable queue using Redis (BRPOPLPUSH) with AOF persistence and a goroutine worker pool of 20 concurrent workers for crash-safe parallel HTTP delivery.
- Implemented Ack/Nack handling and automatic recovery of stalled jobs on restart, achieving zero message loss across simulated crash scenarios.
- Built event ingestion API with Gin and PostgreSQL (JSONB) for immutable event storage; indexed delivery states reduce retry-query time by ~60% vs. full-table scans.
- Decoupled ingestion and delivery via async processing, improving throughput by ~3× compared to synchronous delivery under identical load.
- Containerized with Docker Compose including health checks, environment-based config, and reproducible local dev setup.

Authentication System *Go, PostgreSQL, Redis, NGINX*
GitHub | **Live Demo**

- Built secure user flows — signup, login, email verification, password reset — with short-lived JWT access tokens (15 min) and rotating refresh tokens (7 days).
- Login endpoint handles ~400 req/s at under 20ms avg latency; rate limiter enforces 100 req/min per IP, reducing brute-force surface by design.
- Hardened with bcrypt (cost 12), Redis token blacklist, and account lockout after 5 failed attempts to prevent credential-stuffing attacks.
- Mitigated XSS and SQL injection via input validation, CORS enforcement, structured logging, and parameterized queries throughout.
- Deployed with Docker and NGINX using SSL/TLS termination for production-ready network security.

BriWorld *Go, React 18, TypeScript, WebSocket, PostgreSQL, Docker*
GitHub | **Live Demo**

- Built a real-time multiplayer geography quiz supporting up to 6 concurrent players via WebSockets with server-side synchronized timers; avg round-trip latency under 80ms on LAN.
- Implemented Levenshtein fuzzy matching (threshold: edit distance ≤ 2) to accept near-correct answers, reducing false rejections by ~30% vs. exact-match scoring.
- Secured sessions with JWT authentication, bcrypt hashing, and optimized pgx queries averaging under 5ms per database read.
- Full-stack Go backend + React 18 + TypeScript frontend; deployed on Render with Neon PostgreSQL via Docker with zero-downtime configuration.

EDUCATION

B.Tech in Computer Science 2021–2025
Modern Institute of Technology & Research Centre, Rajasthan CGPA: 7.8/10

Senior Secondary (Science) 2019–2021
Yashwant Senior Secondary School, RBSE 82%

CERTIFICATIONS & CONTINUOUS LEARNING

IBM DevOps, Cloud, and Agile Foundations (Coursera, 2024)
DevOps: Beginner to Advanced with Projects (Udemy, 2025)
143-day learning streak on daily.dev covering backend, cloud, and DevOps topics.

INTERESTS & HOBBIES

Gaming | Music | Cooking | Linux Customization