RITIKA GUPTA

Backend / Platform / Applied AI Engineer

Email: ritikag5533@gmail.com | **GitHub:** github.com/RitikaxG | **Portfolio:** www.ritikaxg.co.in

SUMMARY

Backend/platform/applied AI engineer focused on production-style systems across Go, TypeScript, Next.js, Postgres, Redis Streams, Docker, Kubernetes, GitOps, AWS, RAG, agents, memory, evals, and observability.

EDUCATION

B.Tech in Data Science and Engineering 2021 - 2025
Manipal University Jaipur

SKILLS

Languages: Go, TypeScript, JavaScript, SQL
Backend: REST APIs, worker pipelines, state machines, background jobs, auth/RBAC, validation, durable workflow state
AI Systems: RAG, LLM agents, tool calling, guardrails, memory, evals, traces, LangChain, Gemini, pgvector
Data / Infra: Postgres, Redis, Redis Streams, Docker, Kubernetes, ArgoCD, External Secrets, Prometheus, HPA, AWS EC2, ASG, S3
Frontend: Next.js, React, Tailwind CSS

PROJECT EXPERIENCE

ClaimFlow AI - Governed Agentic AI Workflow for Insurance Claims May 2026 - Jun 2026
Independent full-stack AI systems project
Tech: Next.js, TypeScript, Postgres, pgvector, LangChain, Gemini, Docker
Links: GitHub | Live

- Built an end-to-end AI workflow that converts claim PDFs/emails into structured, policy-grounded, human-reviewed cases with extraction, validation, policy RAG, memory, guarded agent actions, evals, and trace dashboards.
- Implemented clause-level policy RAG with Postgres/pgvector so coverage reasoning is grounded in retrieved policy evidence instead of unsupported model output.
- Designed a single-step agent workflow where the LLM proposes one registered action while backend guardrails block unsafe actions such as approval, rejection, deletion, or bypassing review.
- Added workflow memory with safe-use rules so trusted prior outcomes can guide routing without copying old claim facts into a new claim.

RunState - Go Uptime Monitoring Platform with GitOps Deployment Jan 2026 - Mar 2026
Independent backend/platform project
Tech: Go, Postgres, Redis Streams, Docker, Kubernetes, ArgoCD, Prometheus
Links: GitHub | GitOps

- Built an uptime monitoring platform with API server, monitoring pusher, monitoring worker, status-change worker, notification worker, user/admin dashboards, and persisted operational history.
- Used Redis Streams and consumer groups to decouple check scheduling, HTTP monitoring, incident transitions, and notification logging.
- Modeled users, websites, ticks, incidents, and notification logs in Postgres so uptime, response times, downtime periods, and alert attempts are auditable.
- Created a GitOps deployment path using Kubernetes manifests, ArgoCD, External Secrets, ingress/TLS, Prometheus metrics, HPA, and image automation.

SpinUp - Cloud Workspace Control Plane Nov 2025 - Dec 2025
Independent cloud platform project
Tech: Next.js, TypeScript, Postgres, Redis, AWS EC2/ASG, S3, Docker, code-server, Clerk
Links: GitHub

- Built a control-plane workflow that turns browser project creation into an EC2-backed code-server workspace with visible lifecycle states.
- Stored durable project state in Postgres and used Redis locks/runtime mirrors to coordinate long-running workspace creation safely.
- Implemented EC2 Auto Scaling Group allocation, VM agent health checks, Docker container boot, S3 restore/sync, readiness polling, and cleanup flows.
- Surfaced truthful runtime state in the UI, including lifecycle status, instance ID, public IP, container name, heartbeat, timestamps, and workspace actions.