

MOHIT PANT

Full Stack Engineer · Backend Systems & Data Integrity · Cloud Infrastructure · AI/Agent Engineering

+91-7355201358 · emohitpant@gmail.com · LinkedIn · Portfolio

PROFESSIONAL SUMMARY

Full Stack Engineer with 2+ years building and validating configurable, business-logic-driven platforms in production. Delivered Tookan dispatch integrations for NEOM, Red Sea Global (RSG), and DUBZ across Saudi Arabia and the UAE, supporting enterprise workflows at 14,000+ daily deliveries. Strong hands-on experience with Node.js, Python/FastAPI, JavaScript/TypeScript, SQL/NoSQL systems, service segmentation, DB migrations, replication, and production reliability. Built a production C++ vehicle-routing optimisation engine and, within the engineering team, converted the Tookan API surface into Python MCP tools and implemented an LLM-powered shared memory layer using fact extraction and Qdrant so multiple coding agents can share organisational context, skills, and workflows. Currently building a multi-agent AI SRE system for automated, safety-gated incident response. Twice awarded Employee of the Quarter.

TECHNICAL SKILLS

Languages & Backend: JavaScript / TypeScript (expert), Node.js, C++17, Python (FastAPI, hands-on), SQL, REST APIs, gRPC

AI / Agent Engineering: Model Context Protocol (MCP), LLM agent workflows, LLM fact extraction, shared agent memory, Qdrant, tool-driven automation

Databases & Data Integrity: MySQL (master-slave replication, query optimisation), PostgreSQL, MongoDB, Redis, DB migrations & schema validation

Frontend: React.js, Angular v5–v18, Micro Frontends, AngularJS, HTML5, CSS3, REST API integration

Architecture & Validation: Service segmentation, monolith-to-microservices migration, configurable-platform customisation, edge-case/failure-point analysis, ACL/RBAC, root-cause analysis, workflow testing

Cloud & DevOps: AWS (EC2, RDS, S3), Docker, Kubernetes, Traefik, HTTPD, HAProxy, Celery (Beat & Workers), CI/CD

Testing & Security: Mocha.js (automation workflow validation), Jest (platform testing), GoogleTest/gtest, Semgrep (SAST), OWASP ZAP (DAST), dependency auditing

Tools & Practices: Postman, Jira, Git, production monitoring & alerting

WORK EXPERIENCE

Software Engineer — Jungleworks (Tookan Ecosystem) Jun 2024 – Present

Remote · Enterprise clients across Saudi Arabia, UAE & Europe

- Validated & customised configurable business logic for NEOM, RSG, and DUBZ, translating client requirements into dispatch configurations, technical specs, validation criteria, and production-ready workflows.
- Stakeholder collaboration & delivery ownership: worked directly with CEOs, CTOs, product stakeholders, QA, operations, and 13+ engineers through full-cycle enterprise delivery.
- Data integrity across MySQL, MongoDB & Redis: implemented replication/read replicas and production DB migrations/schema updates with near-zero downtime — reducing API response times by 61.3%.
- System resilience & failure-point analysis: built automated monitoring and alerting pipelines with Google Chat integration, cutting incident response time by 89%.
- Architecture & service segmentation: migrated monolithic platform capabilities into independently deployable, containerised services — including a dedicated VRP engine — improving isolation and maintainability of high-complexity workflows.
- Mapping cost reduction: every directions, routing and places call ran through Google. Re-architected and debugged the provider layer onto GraphHopper for route geometry and point-to-point routing and Pelias for open-source geocoding and places, cutting Google mapping spend to roughly a quarter — with provider selection kept configurable per client, since regions such as the UAE lack accurate open map data and still route through Google.
- Agentic platform enablement: converted the Tookan API surface into Python-based MCP tools so Claude/LLM agents can securely interact with Tookan workflows, APIs, and client operations programmatically.
- Shared agent memory: implemented an engineering-team memory layer where an LLM extracts durable facts from agent conversations, stores vectorised knowledge in Qdrant, and makes shared context available across Claude and Kiro workflows.
- Workflow verification: maintained Mocha.js automation and integration tests for critical Tookan flows, and Jest coverage on platform code, so service changes and agent-driven actions can be validated against real business behaviour.
- Modernised the Angular frontend from v5 → v18 into scalable microfrontends and improved operational efficiency by 45% and data accessibility by 60%.

INDEPENDENT PROJECTS

Engineering Memory Layer — Shared Agent Memory over MCP github.com/mohitpant05/agent-memory

Python, mem0, Qdrant, Ollama, Model Context Protocol (MCP), Docker

- Built a local, self-hosted semantic memory layer shared between coding agents (Claude Code, Kiro) over MCP, so context established with one agent is recallable by another weeks later in a different repository — facts, not transcripts, with no API key, no cloud and no network egress.
- An LLM extracts durable facts from agent conversations, a local model embeds them, and Qdrant stores the vectors scoped by user, agent and run, with ADD/UPDATE/DELETE reconciliation rather than blind appends.
- Diagnosed and fixed three upstream defects that each failed silently — tools reporting success while storing nothing: reasoning models returning empty content, a fact-extraction prompt that discarded technical statements, and a delete guard whose scope check was dead code.
- Applied the fixes at runtime via a .pth shim so the pinned upstream commit stays byte-identical to what was reviewed, with CI asserting the dependency ceilings that keep the wrapper importable.
- Shipped a one-command installer with verify, doctor and status subcommands; the installer refuses to bind 0.0.0.0 because the HTTP transport has no inbound authentication.

AIOrchestra — Agentic Orchestrator for 24/7 Server Availability github.com/mohitpant05/AIServerOrchestra

Python, FastAPI, PostgreSQL (pgvector), Redis, SSH, Prometheus, Grafana, Loki, Docker Compose

- Building a multi-agent incident-response system that ingests server metrics every 30s, diagnoses alerts with an LLM, and either remediates automatically or escalates to a human — 24/7 availability management without a standing on-call rotation.
- Designed a safety-gated decision layer: an action executes automatically only when it is whitelisted, model confidence is ≥ 0.90 , and a Redis rate limit of 3 actions per service per hour is clear; anything else becomes a Telegram approval request with inline approve/deny.
- Implemented incident memory with pgvector similarity search so root-cause analysis is grounded in past incidents, plus hourly trend analysis for risk-window and scaling predictions.
- Built agent-based discovery over SSH across systemd services, Docker containers, Kubernetes pods and open ports, with a watchdog that confirms unreachable hosts by SSH probe after 90s of collector silence.
- Abstracted the LLM behind a provider interface (Ollama, OpenAI, Anthropic, OpenRouter) and instrumented the stack with Prometheus, Alertmanager, Grafana and Loki.

Vehicle Routing Engine — Pickup-and-Delivery Route Optimisation [vrp.mepant.com](https://mepant.com/vrp)

C++17, adaptive large neighbourhood search, gRPC, REST APIs, GraphHopper, Pelias, Docker, Kubernetes

- Built a production pickup-and-delivery routing engine in C++ planning 2,000-stop routes under 12 hard constraints including capacity, time windows, shifts, vehicle compatibility, precedence, and zones.
- Deployed into the Jungleworks stack to cut mapping spend: every directions, routing and places call previously went to Google. Re-architected the provider layer onto GraphHopper for route geometry and point-to-point routing and Pelias for open-source geocoding and places, reducing Google cost to roughly a quarter.
- Kept provider selection configurable per client rather than global, because open map data is not uniformly accurate — regions such as the UAE still route through Google where GraphHopper coverage is too weak to trust.
- Reduced end-to-end solve time from 143s → 30s for typical 60–70 task/point workloads through routing and optimisation pipeline improvements.
- Cut distance-matrix acquisition 235s → 52s by replacing barrier-synchronised fetching with a bounded worker pool and Hilbert-ordered spatial pruning, reducing provider calls 1,225 → 174 per request.
- Reworked an unbounded construction phase to commit per-route batches within the request time limit — 1,720-stop requests went from returning zero routes to routing every stop in 13s.
- Hardened with 257 unit, integration, and constraint-fuzz tests running under ctest in CI; shipped via multi-stage Docker so the runtime image carries the binary rather than the build toolchain.

Rental Manager — Multi-Tenant Resource Rental Management Platform [rm.mepant.com](https://mepant.com/rm)

Python, FastAPI, PostgreSQL, Celery (Beat & Workers), MinIO, Meilisearch, Traefik, Docker, Kubernetes

- Designing and building a multi-tenant SaaS platform from the ground up for the resource rental industry, with tenant-level data isolation for inventory, bookings, and operations.
- Architecting backend services in Python/FastAPI with PostgreSQL and configurable tenant-specific business rules.
- Implementing asynchronous and scheduled processing with Celery Beat/Workers for booking reminders, inventory sync, and background operations.
- Setting up MinIO for tenant document/file storage and Meilisearch for fast full-text resource discovery.
- Containerising and orchestrating services with Docker and Kubernetes, using Traefik for reverse-proxy routing and load balancing.

Evoq Antalia — Real Estate Website Rework [evoq-antalia.mepant.com](https://mepant.com/evoq-antalia)

HTML5, CSS3, JavaScript, client-side routing, responsive UI

- Rebuilt a broken multi-page real estate marketing site as a single-page application with client-side routing across nine views.
- Built an interactive UI including floor-plan tabs, image-gallery lightbox, mobile drawer navigation, and scroll-triggered animations.
- Designed a custom SVG radial distance diagram and implemented lead-capture validation with a honeypot field for spam protection.
- Shipped as a self-contained responsive HTML/CSS/JS site with accessible focus states and reduced-motion support.

KEY ENTERPRISE PROJECTS

NEOM — Smart City Traveller & Fleet Dispatch Platform Saudi Arabia

Node.js, MySQL, Redis, Angular, AWS, Tookan · Vision 2030 Gigaproject

- Validated and integrated Tookan as the dispatch engine for traveller and fleet logistics across NEOM zones and Neom Bay Airport.
- Built and validated real-time fleet tracking, automated task assignment, route-optimisation flows, and custom APIs/webhooks for zero-manual-intervention workflows.
- Delivered production monitoring with automated alerting to catch SLA-threshold breaches before they impacted operations.

Red Sea Global (RSG) — Tourism Destination Logistics Platform Saudi Arabia

Node.js, MySQL, Angular, Tookan, AWS · PIF-owned Vision 2030 Gigaproject

- Delivered and validated Tookan dispatch integration across RSG's 90+ islands and luxury resorts, targeting 1M+ annual visitors at full buildout.
- Designed and validated ACL/RBAC for hierarchical teams and built scheduling/dispatch automation that reduced manual intervention.
- Built custom reporting and data-export tooling, performing gap analysis to support multi-destination operational decisions.

DUBZ (powered by dnata) — Emirates Home Baggage Check-in Platform Dubai, UAE

Node.js, MySQL, GPS APIs, Angular, Tookan · Official Emirates Airline Partner

- Architected and validated the Tookan dispatch platform against Emirates flight schedules and SLA requirements for doorstep baggage check-in and delivery.
- Built and stress-tested intelligent driver-assignment and route-clustering logic under peak-load conditions, ensuring zero missed SLAs during peak travel periods.
- Integrated GPS tracking APIs and optimised DB queries/Redis caching for peak-load reliability across 10+ airlines at Dubai International Airport.

ADDITIONAL CLIENT ENGAGEMENTS

- Peddlers (Amsterdam, ~14,000 deliveries/day): owned and validated a Tookan-based delivery platform with MySQL master-slave architecture, Redis caching, and near-zero-downtime DB migrations.
- Poona Cabs (India): customised and validated automated employee transport scheduling with route-distance optimisation and GPS tracker integration.
- McDonald's India: built and validated a Tookan fork for outlet maintenance scheduling with hierarchical ACL/RBAC.

EDUCATION

B.Tech — Computer Science & Engineering — GNA University, Phagwara 2021 – 2025

GPA: 8.4 / 10

INTERNSHIP

Full Stack Intern — ShapeMySkills 2023

- Built a full-stack URL shortener with MongoDB analytics and MySQL for structured data, focused on performance and scalability.

ACHIEVEMENTS

- Employee of the Quarter — awarded twice at Jungleworks for exceptional delivery and ownership.
- Production System Owner — advanced from intern-level to owning production systems within months of joining.
- Delivered enterprise Tookan integrations for NEOM and Red Sea Global (Saudi Vision 2030 gigaprojects) and DUBZ, the official Emirates airline baggage partner in Dubai.
- Built and integrated an agentic Tookan layer using Python MCP tools plus shared LLM-extracted memory with Qdrant for cross-agent engineering workflows.
- Independently built a production C++ pickup-and-delivery vehicle routing engine handling 2,000+ stop routes with adaptive large neighbourhood search.
- Independently designing and building Rental Manager, a multi-tenant SaaS platform using FastAPI, PostgreSQL, Docker, and Kubernetes.
- Building AIOrchestra, a multi-agent AI SRE system pairing LLM root-cause analysis with confidence-gated, rate-limited automated remediation and Telegram human-in-the-loop approvals.