

PULICHARLA GOPI KRISHNA

Email: Pulicharalagopikrishna@gmail.com · Phone: +91 81795 64931

LinkedIn: linkedin.com/in/gopikrishna-pulicharla · GitHub: github.com/gopikrishna-p

PROFESSIONAL SUMMARY

Software Engineer with 1.5+ year of experience at Deepgrid Semi, specializing in building scalable backend systems, enterprise applications, and real-time platforms. Proven ability to independently design and deliver production-grade solutions, including HRMS and POS systems using Frappe Framework and full-stack technologies.

Experienced in REST API development, asynchronous programming, and system design, with hands-on expertise in real-time communication (WebSockets), caching (Redis), and ERP-style architectures. Built advanced systems integrating AI/LLMs, data pipelines, and hybrid search, demonstrating strong capability in solving complex data and engineering problems.

Focused on writing high-performance, secure, and maintainable code, with a strong foundation in backend engineering and scalable system design.

EMPLOYMENT

DEEPGRID (Startup)

Software Development Engineer, Full Time Employee

April 2025 - Present

- Led development of a production-grade HRMS platform, managing employee lifecycle, attendance, payroll, and project workflows
- Designed and implemented scalable REST APIs for mobile and web integration
- Built geo-location-based attendance system with real-time tracking and synchronization
- Developed and maintained backend services using Python, ensuring high performance and reliability
- Implemented role-based access control (RBAC) and authentication for secure enterprise usage
- Designed database schemas and optimized queries for efficient data handling using MariaDB
- Worked with real-time technologies (WebSockets / API sync) to enable seamless user experience
- Automated HR processes, improving operational efficiency and reducing manual workload
- Took complete ownership of system design, development, and deployment as a sole developer

SKILL SUMMARY

- **Languages:** Python (advanced), Java, JavaScript (ES6+).
- **Frontend Development:** HTML5, CSS3, Bootstrap, ReactJS (Hooks, State Management, Component-Based Architecture), Responsive UI Design.
- **Backend & Frameworks:** Django, FastAPI, RESTful API Development, Microservices Architecture, JWT Authentication, Role-Based Access Control (RBAC)
- **Real-Time & Integration Technologies:** WebSockets (real-time communication), Firebase Integration (Authentication, Firestore / Realtime DB), REST API Integration, Third-Party API Integration
- **Databases & Caching:** MySQL, MariaDB, SQLite, Database Design, Query Optimization, Transactions, ORM (Django ORM), Redis (Caching, Performance Optimization).
- **Async & Performance Engineering:** Asynchronous Programming (async/await in Python), Concurrent Request Handling, API Performance Optimization.
- **DevOps & Deployment:** Git, GitHub (branching, version control), Docker (basics), CI/CD fundamentals, Application Deployment.
- **Cloud & Backend Services:** Firebase, Cloud Concepts (Hosting, Storage, Authentication Services)
- **Software Engineering Fundamentals:** Object-Oriented Programming (OOP), System Design Basics, Debugging, Code Optimization.
- **Platforms:** Code Blocks, PyCharm, Jupyter Notebook, Google Colab, Anaconda, Visual Studio Code.

PROJECTS

HRMS (Human Resource Management System) – hr.deepgrid.in

Role: Project Lead - Full-Stack Developer / Android App Developer - Sole Developer

Tech: Python, JavaScript, React Native, REST API, MariaDB

- Developed an end-to-end HRMS solution to manage employee lifecycle, attendance, leave, and shift operations
- Designed and implemented key modules including Employee, Attendance, Leave Management, Holiday List, Shift Management, HR Settings, Payroll, Projects.
- Built a mobile application with geo-location-based attendance and real-time employee services
- Designed and integrated REST APIs for seamless communication between mobile and backend systems.
- Integrated the app with the backend via REST APIs, ensuring real-time sync between app and ERP system.
- Implemented role-based access control, validations, and user dashboards to enhance security and usability.
- Automated HR workflows and improved operational efficiency through centralized system design.

Resort Management System — Resort Operations & Hospitality ERP (api.cholabeachresort.com)

Role: Project Lead - Full-Stack Developer / Android App Developer - Sole Developer

Tech: Python, Frappe Framework v15, MariaDB, Redis, React Native (TypeScript), REST API, WebSocket's, Firebase Cloud Messaging

- Designed and built an end-to-end resort management platform covering the full guest lifecycle — booking, check-in, cottage allocation, in-stay services, billing and checkout — for a 14-cottage beach resort running live in production.
- Architected a versioned REST API (18 modules: bookings, availability, pricing, billing, orders, service, housekeeping, maintenance, reports) with a single response envelope and error-code taxonomy, so web and mobile clients share one transport layer.
- Built a React Native staff application (39 screens) with role-specific workflows for Reception, Guest Service, Kitchen, Housekeeping and Management, using React Query, Zustand and TypeScript.
- Implemented an immutable folio ledger where bills are always computed as the sum of submitted entries and corrections are posted as reversals — giving a fully auditable, dispute-proof billing trail with GST tax invoicing and invoice series.
- Developed real-time order and notification delivery over WebSockets (Frappe SocketIO) with user-scoped rooms, plus FCM push notifications (OAuth2 / FCM v1 API) for delivery to closed apps, with a queued fall-back when realtime is down.
- Built ESC/POS thermal printing via a custom React Native native module (TurboModule) for kitchen order tickets (KOT), food bills and consolidated stay bills over Bluetooth.
- Designed a secure guest QR portal (unauthenticated web entry point) using per-stay access tokens, PIN verification, rate limiting and a mandatory session guard on every endpoint — enabling guests to view bills, order food and raise service requests without exposing personal or cross-stay data.
- Implemented fine-grained RBAC across 10 roles using Frappe role permissions, field-level permlevels and permission query conditions (e.g. "a GSA sees only their own assigned stays"), enforced at the API layer rather than the UI.
- Added idempotency-key handling on all mutating endpoints to make order placement, payments and check-ins safe against retries on unreliable mobile networks.
- Engineered availability and stay-lifecycle logic including cottage blocking, cleaning buffers, occupancy rules, extra-bed pricing, cancellation slabs and rate-plan/floor-price validation.
- Created a reproducible deployment pipeline using Frappe fixtures and seed scripts, allowing a new server to be provisioned with full master data (cottages, tariffs, 100-item menu, service SLAs) without any data-base copy.
- Wrote an automated test suite (Python unit + integration tests, Jest tests for the mobile app) covering billing arithmetic, portal security and permission boundaries.

Chola Beach Resort — Resort Website & Enquiry Platform (www.cholabeachresort.com)

Role: Full-Stack Developer / Frontend Architect - Sole Developer

Tech: TypeScript, Next.js 16 (App Router), React 19, Tailwind CSS v4, Zod, Vitest, Playwright, Vercel

- Designed and built a production-grade marketing and lead-generation website for a 14-cottage beachfront resort, covering 18+ routes including dynamic cottage detail pages, celebrations/venues, offers, gallery, and enquiry flows.
- Architected the codebase on ports-and-adapters (hexagonal) principles — a framework-free domain core (domain, ports, use-cases) with infrastructure adapters at the edges — and enforced the import boundaries with custom ESLint rules in CI.
- Built a content-as-data layer of 65+ typed JSON files validated at load time with Zod schemas, allowing non-technical content updates without code changes while guaranteeing type safety and fail-fast validation.
- Engineered a single-source design token system: all colour, type and spacing values generated from one JSON source into CSS variables and Tailwind theme config, with a custom lint rule that fails the build on any raw hex value outside the token file.
- Implemented seasonal theming — five calendar-driven themes (peak, summer, monsoon, festive, new year) that re-skin the entire site, including hero imagery and featured offers, by remapping semantic tokens on a single root attribute.
- Designed an enquiry pipeline with validation, deduplication, and a pre-filled WhatsApp handoff, structured so the guest always retains a contact path even if notification delivery fails.
- Defined IAvailabilityProvider and IPaymentGateway interface ports backed by stubs, so real booking and Razorpay UPI integration becomes an adapter drop-in with zero page changes.
- Built an automated image optimisation pipeline using Sharp — responsive resizing plus base64 blur placeholders — to meet a strict performance budget (LCP < 2.0s, CLS < 0.05, JS < 120KB gzipped) on mid-range Android over 4G.
- Achieved WCAG 2.1 AA compliance, verified by an automated contrast test across every seasonal theme plus axe-core accessibility scans at desktop and 360px viewports.
- Implemented schema.org structured data (Resort, LodgingBusiness with geo, Place, FAQPage) generated from the same content the page renders, preventing markup and visible text from ever drifting.
- Wrote 27 test files across unit, contract, component (Testing Library) and end-to-end (Playwright) suites, wired into a GitHub Actions CI pipeline gating types, lint, formatting, token sync, tests and build on every push.

POS (Point of Sale System) — Restaurant Management Platform

Role: Full-Stack Developer - Sole Developer

Tech: Python, Frappe Framework, ERPNext Concepts, MariaDB, JavaScript

- Designed and developed a custom POS system using Frappe Framework, tailored for small restaurant operations
- Built and customized DocTypes (data models) for orders, invoices, menu items, inventory, and customer management
- Implemented real-time order and billing workflows supporting dine-in, takeaway, and payment processing
- Developed server-side business logic using Frappe controllers and Python, ensuring accurate billing and transaction handling
- Customized Frappe UI forms, dashboards, and reports for efficient staff operations and management insights
- Integrated role-based access control (RBAC) using Frappe's permission system for admin and staff users
- Implemented inventory management with stock updates and validations, reducing manual tracking errors
- Built sales reports and analytics dashboards using Frappe reporting tools for daily revenue tracking

StatSignal — Data Impact & Research Monitoring Platform

Role: Data Engineer / NLP Pipeline Developer – Team Member

Tech: Python, FastAPI, PostgreSQL, Dagster, NLP Models, OpenAI/Claude APIs

- Developed a multi-source data monitoring platform to track dataset usage across news, research papers, blogs, and policy documents.
- Designed a layered ELT pipeline (L0–L3) ensuring data lineage, reproducibility, and auditability.

- Implemented NLP-based enrichment pipeline with: Named Entity Recognition (NER), Sentiment analysis, Claims extraction.
- Built a campaign-based tracking system to monitor specific organizations and datasets automatically.
- Created a weighted scoring model to measure dataset impact across multiple channels (news, citations, policy).
- Integrated multi-provider adapters to unify heterogeneous APIs into a single schema.

StatLens — AI-Powered Statistical Data Query Engine

Role: Backend Developer / Full-Stack Data Engineer – Team Member

Tech: Python, FastAPI, PostgreSQL, pgvector, React, Next.js, OpenAI GPT

- Architected a natural language query engine for government statistics, enabling users to ask questions like “GDP by state” and receive structured, cited answers.
- Built a multi-stage data ingestion pipeline (10 phases) to collect and normalize datasets from 10+ global data sources (Census, WHO, World Bank, etc.).
- Implemented hybrid search (semantic + keyword) using pgvector + BM25, enabling fast and accurate dataset discovery across millions of records.
- Developed an LLM-powered query pipeline that decomposes user queries, selects relevant datasets, fetches real-time data, and generates human-readable insights.
- Designed precomputed query “slices” to reduce latency and optimize performance for frequently asked statistical queries.
- Integrated OpenTelemetry & monitoring tools for tracking LLM usage, latency, and system performance.