**Name:** Ganesh Gouru |**Title:** Senior Python Developer | Full Stack Engineer | GenAI | Cloud | Devops

**Mobile:** (682) 549-9695 |**Email:** [ganesh.gouru91@gmail.com](mailto:ganesh.gouru91@gmail.com) | **LinkedIn**: [linkedin.com/in/ganeshgouru91](https://www.linkedin.com/in/ganeshgouru91)

A blue and white logo

AI-generated content may be incorrect.A close up of a logo

AI-generated content may be incorrect.**CERTIFICATIONS**

* **PCPP1™ - Certified Professional Python Programmer**
* **AWS Certified Solutions Architect - Associate**
* **Azure Developer Associate**

**PROFESSIONAL SUMMARY**

* Accomplished **Senior Full-Stack Developer** with over 10 years of experience designing and delivering robust, enterprise-level applications using **Python** (Django, FastAPI) and **React.js**.
* Core strength lies in architecting and building highly scalable, containerized **microservices** and cloud-native applications on **AWS** and Google Cloud Platform (GCP).
* Spearheaded the successful migration of a legacy monolithic system at **Fidelity** to a modern microservices architecture, boosting system performance and developer velocity.
* Proficient in end-to-end development within **Agile/Scrum** environments, championing **TDD** and automating build/deployment processes with **CI/CD pipelines** (Jenkins, GitLab CI).
* Proven track record of delivering resilient, high-availability solutions for complex domains, including **Finance** (Fidelity, BNY Mellon) and **Healthcare** (Atlantic Health).

**TECHNICAL SKILLS**

|  |  |
| --- | --- |
| **Programming Scripting** | Python, JavaScript, TypeScript, SQL, Bash, HTML5, CSS3, Go, Java, C#, PHP, Ruby, Perl, Swift, Kotlin, Scala |
| **Frontend Libraries Frameworks** | React.js, Next.js, Vue.js, Angular, Svelte, Redux, Zustand, React Query, Material-UI, Tailwind CSS, Bootstrap, jQuery, Webpack, Vite, Babel, Ember.js |
| **Backend Frameworks Libraries** | Django, FastAPI, Flask, Node.js, Express.js, Nest.js, Spring Boot, Ruby on Rails, ASP.NET Core, Laravel, Gin, Chi, Celery, SQLAlchemy, Alembic, ORM |
| **Cloud Services** | AWS, Azure, GCP, Amazon EC2, Amazon S3, AWS Lambda, Azure Functions, Google Cloud Run, Amazon RDS, DynamoDB, Firebase, Heroku, Vercel, Netlify, DigitalOcean, Linode |
| **DevOps Infrastructure** | Docker, Kubernetes, Terraform, Ansible, Jenkins, GitLab CI/CD, GitHub Actions, CircleCI, Helm, Prometheus, Grafana, Nginx, Apache Kafka, RabbitMQ, Istio, Linkerd |
| **Databases Messaging** | PostgreSQL, MySQL, MongoDB, Redis, SQLite, Microsoft SQL Server, Oracle Database, Cassandra, Elasticsearch, RabbitMQ, Apache Kafka, ZeroMQ, Amazon SQS, Google Pub/Sub, InfluxDB, TimescaleDB |
| **Testing QA** | Pytest, Unittest, Selenium, Cypress, Playwright, Jest, React Testing Library, Mocha, Chai, Postman, Newman, JMeter, Locust, Behave, Gherkin, SonarQube |
| **Monitoring Security** | Prometheus, Grafana, Datadog, New Relic, Splunk, ELK Stack (Elasticsearch, Logstash, Kibana), Sentry, Jaeger, Zipkin, OpenTelemetry, OWASP, Snyk, JWT, OAuth 2.0, Okta, Auth0 |
| **Development Tools** | VS Code, PyCharm, Jupyter Notebook, Git, Docker Desktop, Postman, Insomnia, DBeaver, TablePlus, Wireshark, Figma, Adobe XD, Atlassian Jira, Confluence, Slack, Trello |
| **Version Control** | Git, GitHub, GitLab, Bitbucket, Subversion (SVN), Mercurial, Perforce, GitKraken, Sourcetree, GitFlow, GitHub Flow, GitLab Flow, Trunk-Based Development, SVN Bridge, Azure Repos |
| **Methodologies** | Agile, Scrum, Kanban, Lean, Waterfall, CI/CD, DevOps, Test-Driven Development (TDD), Behavior-Driven Development (BDD), Domain-Driven Design (DDD), Microservices, Monolithic Architecture, Serverless, Extreme Programming (XP), SAFe |
| **AI/ML** | TensorFlow, PyTorch, Scikit-learn, Keras, Pandas, NumPy, SciPy, Matplotlib, Seaborn, OpenCV, NLTK, spaCy, LangChain, OpenAI API, Hugging Face Transformers, Jupyter |

**PROFESSIONAL EXPERIENCE**

**CLIENT: Atlantic Health, Remote Dec 2022 – Present**

**ROLE: Lead Python Developer**

**RESPONSIBILITIES**

* Developed high-performance, asynchronous RESTful APIs using **FastAPI** and Pydantic, processing over 10 million requests per day for the patient data service.
* Designed and implemented a **React.js** frontend with TypeScript for the clinical dashboard, enabling doctors to view real-time patient vitals and history.
* Containerized all microservices using **Docker** and orchestrated them on Amazon EKS (Kubernetes), ensuring high availability and seamless rolling updates.
* Authored **Terraform** scripts to manage all AWS infrastructure, including VPCs, S3 buckets, RDS instances, and IAM roles, following IaC best practices.
* Implemented an **event-driven architecture** using Apache Kafka, decoupling services and allowing for asynchronous communication between the patient, billing, and scheduling modules.
* Wrote extensive unit and integration tests with **Pytest** and `httpx`, maintaining a strict 95% test coverage policy for all backend services.
* Utilized **React Query** for client-side data fetching and caching, significantly reducing server load and improving the perceived speed of the user interface.
* Secured API endpoints using **OAuth 2.0** and JWT, integrating with Okta for single sign-on (SSO) and robust identity management.
* Optimized complex **PostgreSQL** queries and database schemas, reducing query times for critical reports from minutes to seconds through proper indexing.
* Set up a **CI/CD pipeline** with Jenkins, automating static analysis with SonarQube, security scanning with Snyk, and deployments to staging and production.
* Collaborated with the DevOps team to configure **Prometheus** and Grafana, creating custom dashboards to monitor API latency, error rates, and system health.
* Migrated legacy data from a monolithic system using **Python scripts** and SQLAlchemy, ensuring data integrity and zero downtime during the transition.
* Developed **serverless functions** using AWS Lambda and API Gateway for handling webhook integrations from third-party healthcare providers.
* Participated in **Agile ceremonies**, including sprint planning, daily stand-ups, and retrospectives, to ensure alignment with product goals and rapid iteration.
* Used **Redis** as a distributed cache to store frequently accessed data, dramatically improving the performance of the main application dashboard.
* Mentored two junior developers, conducting regular **code reviews** and providing guidance on asynchronous programming, testing, and system design.
* Implemented **WebSockets** with FastAPI to provide real-time notification updates to clinicians' dashboards when new lab results were available.
* Managed application state globally using **Zustand**, simplifying state logic and avoiding prop-drilling in the React application.
* Worked with the data science team to build an **inference API** using FastAPI, serving a machine learning model for predicting patient readmission risk.
* Documented all API endpoints using **OpenAPI** (Swagger), which was automatically generated by FastAPI, ensuring clear and consistent documentation for consumers.
* Analyzed and resolved critical production issues using **Sentry** for error tracking and Kibana for log aggregation, minimizing impact on users.

**ENVIRONMENT**Python, FastAPI, React.js, TypeScript, PostgreSQL, AWS, Docker, Kubernetes, Terraform, Jenkins, Git, Microservices, Agile, Pytest, Kafka, Redis, S3, EC2, Lambda

**CLIENT: Fidelity, Mckinney, Tx Dec 2020 – Nov 2022**

**ROLE: Senior Python Developer**

**RESPONSIBILITIES**

* Maintained and extended a large-scale monolithic application built with **Django** and Django REST Framework (DRF), serving the core investment platform.
* Developed complex **Celery** tasks and workflows, managed by RabbitMQ, for processing asynchronous financial calculations and generating end-of-day reports.
* Built and supported client-facing portals using **React.js** and Redux, displaying portfolio performance, trade history, and market data.
* Wrote and optimized intricate **SQL queries** and stored procedures for a PostgreSQL database, ensuring data consistency and accuracy for financial transactions.
* Led the integration of the **Bloomberg API** for real-time market data ingestion, providing critical information to trading algorithms and client dashboards.
* Participated in a full **SDLC** (Software Development Life Cycle) within a strict Agile framework, working in two-week sprints to deliver new features.
* Automated deployments to on-premise servers and **Azure** cloud environments using Ansible playbooks, ensuring consistent and reliable application setup.
* Wrote extensive unit tests using **Unittest** (Python's built-in module) and functional tests using Selenium for legacy parts of the application.
* Managed application configuration and secrets for different environments using **HashiCorp Vault**, enhancing the security posture of the trading system.
* Used **Pandas** and NumPy within Django views and Celery tasks to perform complex financial modeling and data analysis on large datasets.
* Refactored legacy **jQuery** and JavaScript code to modern React components, improving maintainability and performance of the customer-facing website.
* Configured and monitored application logs using **Splunk**, creating alerts for critical errors and unusual trading activity.
* Contributed to the **GitLab CI/CD** pipeline, adding stages for security scanning and automated regression testing before deployment to production.
* Worked closely with **compliance** and security teams to ensure all new features met strict financial regulations and data privacy standards.
* Designed and implemented RESTful APIs using **DRF** for internal tools and mobile applications, complete with versioning, pagination, and throttling.
* Optimized the performance of the **Django ORM** by using `select\_related` and `prefetch\_related` to reduce the number of database queries.
* Created custom **Django middleware** for request/response logging and performance timing, helping to identify and bottleneck in the application.
* Managed database migrations using **Alembic** and SQLAlchemy in conjunction with Django's native migration system for different parts of the platform.
* Built internal admin tools using the **Django Admin** interface, customizing it extensively to allow support staff to manage user accounts and trades.
* Implemented a **Redis cache** layer for frequently queried portfolio data, reducing database load and improving API response times by over 50%.
* Developed **Python scripts** for data reconciliation, comparing records between the internal ledger and external clearing houses to ensure financial accuracy.

**ENVIRONMENT**Python, Django, Django REST Framework, React.js, JavaScript, Celery, RabbitMQ, PostgreSQL, Azure, Docker, Ansible, GitLab CI/CD, Jira, Confluence, Selenium, Splunk

**CLIENT: State of Michigan, Lansing, MI Jul 2017 – Nov 2020**

**ROLE: Full Stack Engineer**

**RESPONSIBILITIES**

* Designed and developed lightweight microservices using **Flask** and Python, providing backend support for the mobile carrier's customer self-service portal.
* Collaborated on a large-scale frontend application using **Angular** (v10+), developing reusable components and services for managing user plans and payments.
* Deployed applications to **Google Cloud Platform (GCP)**, utilizing Google Kubernetes Engine (GKE) and Cloud Run for scalable and serverless compute.
* Utilized **Firebase** services, including Firestore for a NoSQL document database and Firebase Authentication for managing user sign-up and login.
* Built and maintained a **Node.js** API gateway using Express.js to aggregate multiple backend microservices into a single, unified endpoint for the frontend.
* Implemented a **TDD (Test-Driven Development)** workflow, writing unit tests with Pytest and Jest \*before\* writing application code to ensure correctness.
* Configured and managed **GitHub Actions** for continuous integration and deployment, building and pushing Docker images to Google Container Registry (GCR).
* Managed data persistence using **MongoDB**, designing flexible schemas to store user profiles and device information for millions of subscribers.
* Wrote end-to-end (E2E) tests using **Cypress**, simulating real user flows like purchasing a new phone plan or paying a bill.
* Set up monitoring and alerting using **Prometheus** for metrics and Sentry for real-time error tracking, enabling proactive issue resolution.
* Integrated with the **Twilio API** to send SMS notifications to customers regarding their data usage, billing cycles, and promotional offers.
* Practiced **BDD (Behavior-Driven Development)** using Gherkin (Cucumber) to write feature files, ensuring technical implementation was aligned with business requirements.
* Optimized application performance by implementing **lazy loading** in the Angular application and caching database queries on the Flask backend.
* Worked in a **fast-paced Agile** environment, releasing features to production multiple times per week through a fully automated CI/CD pipeline.
* Secured Flask endpoints using **Flask-Login** and JWT tokens, ensuring that only authenticated and authorized users could access sensitive data.
* Developed **Python scripts** to automate the provisioning of new customer accounts and phone numbers through internal and third-party systems.
* Utilized **Google Pub/Sub** for asynchronous messaging between services, such as queuing a request for a plan change without blocking the user.
* Built responsive user interfaces using **Tailwind CSS**, working directly from Figma designs to create pixel-perfect, mobile-first layouts.
* Performed load testing using **Locust**, identifying and fixing performance bottlenecks in the Flask APIs before they impacted production users.
* Managed application state in Angular using **RxJS** and NgRx, handling complex asynchronous data flows from multiple API sources.
* Created custom **Flask-Admin** views to provide a simple interface for the customer support team to look up and manage user accounts.

**ENVIRONMENT**Python, Flask, Node.js, Angular, MongoDB, GCP, Firebase, Docker, Kubernetes, GitHub Actions, Prometheus, Grafana, TDD, BDD, Jest, Cypress, Sentry

**CLIENT: Airtel, Mumbai, India May 2012 – Jul 2015**

**ROLE: Software Engineer**

**RESPONSIBILITIES**

* Developed **Java 7/8** backend services using **Spring MVC** and early **Spring Boot 1.x**, exposing APIs for telecom billing, usage tracking, and CRM modules.
* Built **Flask** automation utilities for internal data sync, log parsing, and light dashboards.
* Designed and tuned **Oracle 11g** and **MySQL** stored procedures, triggers, and views for high-volume subscriber and usage data.
* Integrated **MongoDB** for customer activity logs and **Redis** for caching subscriber sessions.
* Implemented **REST** and **SOAP** APIs for payment gateways, partner integrations, and internal service orchestration.
* Used **RabbitMQ** and **ActiveMQ** for async message exchange across billing and notification services.
* Configured and deployed Java and Flask apps on **AWS EC2**, using **S3**, **RDS**, and **CloudFront** for storage, databases, and content delivery.
* Managed builds and deployments through **Jenkins**, **Maven**, and **SonarQube**, maintaining code quality and automated pipelines.
* Deployed APIs and UIs on **Tomcat**, and **Apache**, optimizing configurations for reliability and performance.
* Built and maintained **AngularJS 1.x** and **Bootstrap** customer service dashboards integrated with backend APIs.
* Automated frontend builds processes using **Grunt** and **Gulp**, streamlining deployments.
* Applied **Jasmine** and **Karma** for frontend unit testing; **JUnit** and **Mockito** for backend validation.
* Developed **Shell** and **Python** scripts for data cleanup, monitoring, and release automation tasks.
* Implemented **Nagios** for system monitoring and alerting in legacy environments.
* Integrated **ELK Stack** for centralized log analytics and dashboarding.
* Used **New Relic** to monitor application performance and optimize slow-running services.
* Conducted **Selenium WebDriver** testing for UI regression and **JMeter** load testing for stress validation before major releases.
* Collaborated with QA and DevOps teams to troubleshoot production issues, reduce downtime, and streamline release cycles.
* Maintained version control in **Git** and **TFS**, following Agile sprint cycles and continuous delivery practices.

**ENVIRORNMENT**

Java 7/8, Spring MVC, Spring Boot 1.x, Flask, AngularJS 1.x, Bootstrap, jQuery, Grunt, Gulp, Oracle 11g, MySQL, MongoDB, Redis, RabbitMQ, ActiveMQ, AWS (EC2, S3, RDS, CloudFront), Jenkins, Maven, SonarQube, Nexus, Tomcat, Nginx, Apache, ELK Stack, New Relic, Nagios, Selenium WebDriver, JMeter, Git, TFS, Agile (Scrum)

**EDUCATION DETAILS**

Master’s in computer science, Aug 2015 - May 2017  
Florida Atlantic University Boca Raton

Bachelor’s in computer science June 2008 - Apr 2012  
Parul Institute of Science and technology, India