

Durgesh Rathod

Technical Lead | GenAI | Python | Distributed Systems | AI Agents | MCP | Kubernetes | AWS

Email: durgeshrathod.777@gmail.com

Contact number: +91 8208406596

Linkedin: <https://www.linkedin.com/in/durgesh-rathod-711a959b/>

Github: <https://github.com/DurgeshRathod>

Professional Summary

Technical Lead with over **9+ years** of hands-on experience in designing and developing **enterprise software, distributed systems, AI-powered applications, and large-scale data platforms**. Experienced in partnering with stakeholders, Product Managers, Business Analysts, and cross-functional teams to understand business requirements and deliver scalable technical solutions across telecom, HR, recruitment, and enterprise workflow domains.

Expertise spans the complete product lifecycle, including **system design, architecture, development, cloud infrastructure, testing, deployment, production support, and performance optimization**. Hands-on experience building AI applications using **OpenAI, AWS Bedrock, LangChain, CrewAI, and MCP**, along with designing high-performance telecom data pipelines processing **over 520 million data parameters every 15 minutes** using **Golang, Kafka, Kubernetes, and PostgreSQL**.

Demonstrates strong technical leadership with a proven track record of architecting scalable systems, mentoring engineering teams, delivering production-ready enterprise solutions, and driving projects from requirements gathering through production deployment.

Certifications

- Red Hat Certified Engineer
- Generative AI with Large Language Models

Open source contributions

- Published as an open-source Python package : <https://pypi.org/project/fast-intent-classifier/>
- npm package: <https://www.npmjs.com/package/ai-test-gen-angular>

Skills

- Technical Leadership, System Design, Architecture, Design Patterns, SOLID Design Principles
- Python, Golang, Node.js, Angular, JavaScript, HTML, CSS
- OpenAI, Claude, AWS Bedrock, Prompt Engineering, LangChain, CrewAI, AI Agents, Model Context Protocol (MCP)
- Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Transformers, FastAPI, REST APIs
- NumPy, Pandas, Scikit-learn, Statistical Analytics, Predictive Modeling, NLP
- Kafka, Distributed Systems, Microservices, Configuration-Driven Architecture, Mono-Repos
- PostgreSQL, Oracle, SQL Server, MongoDB, Elasticsearch, Vector Databases
- Kubernetes, Docker, YAML, Shell Scripting
- AWS Lambda, Amazon ECS, AWS Fargate, S3, API Gateway, SES, SQS, SNS, SageMaker, CloudWatch
- CI/CD (AWS CodePipeline, Jenkins), Git
- System Design, Database Design, Design Patterns, SOLID Design Principles

- Performance Optimization, Production Troubleshooting, Release Management
- NewRelic, Sonar, Snyk, Kibana
- CKEditor, Strapi, Puppeteer, Twilio, RChilli
- Good communication and problem-solving skills

Work Experience

SourceFuse Technologies India Pvt Ltd	Technical Lead	Feb 2021 - Present
Flairlabs	Software Engineer	Dec 2018 - Jan 2021
Merce Technologies	Application Engineer	Jun 2017 - Dec - 2018

Technical Lead

Mar 2022 - Present

- Discussion and brainstorming with Stakeholders and the BA team for GenAI features and enterprise platform solutions.
- Designed and developed large-scale Configuration Management (CM) and Performance Management (PM) telecom data pipelines.
- Hands-on experience with Golang, Kafka, Kubernetes, and PostgreSQL for building distributed backend systems.
- Designed and developed a configuration-driven data pipeline framework that reads source connectors, processing logic, and destination configurations dynamically.
- Processed over **200 million CM parameters** and **320 million PM parameters** every 15 minutes for a telecom client.
- Hands-on with AWS Bedrock, OpenAI, LangChain, CrewAI, and Model Context Protocol (MCP).
- Experience with developing ML services using FastAPI.
- Hands-on experience with SageMaker and Bedrock Knowledge Bases.
- Hands-on experience with CI/CD using AWS CodePipeline.
- Designed and developed the architecture of the system at both application and infrastructure levels.
- Used NewRelic to monitor and debug the entire stack for performance bottlenecks and optimize system performance.
- Used Snyk integrations to identify security vulnerabilities and upgrade dependencies.
- Led whiteboarding sessions with software engineers for designing modular applications and scalable system architecture.
- Plan, estimate, develop, code review, and ensure the implementation is done using engineering best practices.
- Release Management.
- Actively contributed to sprint planning, production releases, troubleshooting, root cause analysis, and removing technical blockers.
- Worked on database design, system design, workflow optimization, and performance tuning to build scalable solutions.
- Conduct regular code reviews, pair programming, and design sessions to ensure code quality, reusability, and maintainability.
- Technical planning with the team to finalize the right implementation approach using SOLID principles and design patterns.
- Integrated unit tests into the Jenkins pipeline to execute automated tests on every pull request.
- Mentored software engineers and ensured engineering best practices, design patterns, and coding standards were consistently followed.

Senior Software Engineer

Feb 2021 - Feb 2022

- Designed and developed the architecture of the system at both application and infrastructure levels.
- Deployment was done using Amazon ECS Fargate containers.
- Led whiteboarding sessions with software engineers for designing modular applications and scalable system architecture.
- Hands-on experience with Amazon ECS, ECR, EC2, and S3.
- Hands-on experience with CI/CD using AWS CodePipeline.
- Plan, estimate, develop, code review, and ensure the implementation is done correctly.
- Created Technical Design Documents (TDD) for backend services.
- Developed the complete backend system along with the required AWS infrastructure.
- Worked closely with cross-functional teams during feature planning and implementation.
- Mentored junior developers and supported them during development and code reviews.

Application Engineer / Software Engineer

Jun 2017 - Jan 2021

- Developed backend features and microservices.
- Experience with AWS Lambda, Amazon S3, and AWS deployment pipelines.
- Worked extensively on Microsoft SQL Server.
- Wrote SQL queries and stored procedures for heavy database operations and report generation.
- Worked closely with the AWS team for setting up cloud services and project infrastructure.
- Understood business requirements, prepared Technical Design Documents (TDD), and implemented the required features.
- Participated in feature development, testing, deployment, and production support.

Projects

1. Telecom Configuration Management & Performance Management Data Pipeline

Technologies Used: Golang, Kafka, Kubernetes, PostgreSQL, Oracle, Azure Data Explorer (ADX), Temporal, Azure

Description: Designed and developed a large-scale, configuration-driven telecom data pipeline platform for processing Configuration Management (CM) and Performance Management (PM) data. The platform dynamically reads configurations for source connectors, processing logic, transformation rules, and destination connectors, enabling new telecom pipelines to be onboarded without any application code changes.

The Configuration Management pipeline processes over **200 million telecom configuration parameters every 15 minutes**, while the Performance Management pipeline processes over **320 million telecom performance parameters every 15 minutes**. The processed data is stored in analytical platforms and serves as the foundation for downstream analytics, reporting, and Machine Learning / AI model training..

Responsibilities:

- Designed a configuration-driven data pipeline framework that enabled onboarding new telecom data sources and business workflows without requiring application code changes..
- Developed high-performance backend services in Golang for processing large-scale telecom data.

- Designed and implemented dynamic source connector, processing, and destination frameworks driven entirely through configuration.
- Processed over 200 million Configuration Management (CM) parameters every 15-minute ingestion cycle.
- Processed over 320 million Performance Management (PM) parameters every 15-minute ingestion cycle.
- Built scalable, distributed data processing services using Kafka and Kubernetes.
- Designed optimized processing logic to efficiently handle hundreds of millions of telecom parameters with high throughput and low latency.
- Developed reusable pipeline components to support multiple telecom data sources and destination systems.
- Designed PostgreSQL-based metadata and configuration management for dynamic pipeline execution.
- Performed performance tuning and memory optimization to improve throughput and scalability.
- Collaborated with architects and cross-functional teams to design and deliver enterprise-scale distributed systems.
- Contributed to system design, implementation, troubleshooting, deployment, and production support.

2. Gen AI - MCP & Crew AI-Based Chatbot for Project Management Tools:

Technologies Used: Python, FastAPI, MCP (Model Context Protocol), Crew AI, Langchain, Azure OpenAI, Postgres, Docker

Description: Developed an AI-powered chatbot integrated into a project management environment using Model Context Protocol (MCP) and Crew AI architecture. The chatbot enables natural language interaction with project data — including tasks, boards, deadlines, and statuses — through an LLM-coordinated agentic system. It supports project updates, status queries, and intelligent task suggestions. Crew AI was used to orchestrate multiple specialized agents (e.g., How To Guides, Project planning, Updates), enabling semi-autonomous multi-agent workflows and collaborative task management.

Responsibilities:

- Implemented the MCP protocol to modularize tools (task manager, board manager, status handler) and expose them to the Crew AI agent layer.
- Created and configured Crew AI agents with role-based behavior to collectively handle task queries, assignments, blockers, and project overviews.
- Enabled real-time chat interface for users to interact with the project management system conversationally and receive actionable updates.
- Dockerized and deployed the service, ensuring modular, scalable, and agent-driven chatbot behavior.

3. Digital Contract Lifecycle Management Platform

Technologies Used: Angular, Node.js (LoopBack), PostgreSQL, CKEditor, Kubernetes

Description: Developed an enterprise Digital Contract Lifecycle Management (CLM) platform that streamlines the complete contract lifecycle, including contract authoring, review, approval, amendments, clause management, document editing, and template-based contract generation. The platform enables legal and business teams to efficiently create, review, manage, and maintain contracts while ensuring consistency through reusable clause and document templates.

Responsibilities:

- Led discussions with Product Managers, Business Analysts, and Technical Architects to understand business requirements and finalize the technical solution.

- Designed the overall application architecture and guided the team on the implementation approach.
- Planned, estimated, developed, performed code reviews, and ensured high-quality implementation.
- Led whiteboarding sessions with software engineers to design modular, scalable, and maintainable solutions.
- Developed backend services using **Node.js (LoopBack)** and **PostgreSQL**.
- Developed Angular-based user interfaces for contract creation, review, and management.
- Integrated **CKEditor** to provide rich document editing capabilities within the application.
- Designed and implemented contract review, approval, and amendment workflows throughout the contract lifecycle.
- Implemented reusable clause libraries and template-based contract generation to improve consistency and reduce manual effort.
- Designed database schemas and REST APIs for contract, template, clause, and document management.
- Worked closely with stakeholders to convert business requirements into scalable technical solutions.
- Conducted regular code reviews and mentored software engineers to ensure coding standards, design principles, and best practices were followed.
- Worked on system design, database design, workflow optimization, and application performance improvements.
- Managed release planning, production deployments, and post-release support.
- Participated in troubleshooting production issues, root cause analysis, and application maintenance.
- Contributed throughout the complete software development lifecycle, including design, development, testing, deployment, and production support.

4. Gen AI - AI Agent-Based HR Analytics Engine and Chatbot:

Technologies Used: Python, Bedrock, Langchain, fast API, Postgres, Langchain Agents

Description: Designed and developed a multi-tenant Generative AI-powered HR Analytics Engine and Conversational Assistant to help HR teams make data-driven decisions through natural language interactions. The platform enables multiple organizations to securely manage their HR data while maintaining complete tenant isolation. It leverages vector embeddings, Large Language Models (LLMs), LangChain Agents, and Machine Learning models to provide descriptive analytics (workforce dashboards and KPIs), diagnostic analytics (root-cause analysis of attrition, performance trends, etc.), and predictive analytics (employee attrition, time-to-start, and time-to-fill forecasting).

The integrated AI assistant enables HR users to retrieve reports, dashboards, and workforce insights through conversational queries, eliminating the need for manual report generation and significantly reducing the time required to access business insights.

Responsibilities:

- Led discussions with Product Managers, Business Analysts, and stakeholders to understand business requirements and design scalable AI solutions.
- Designed the architecture of a multi-tenant HR Analytics platform with secure tenant isolation and scalable data processing.

- Integrated LangChain Agents to build an intelligent multi-turn conversational assistant for HR reporting and decision support.
- Designed data models and Machine Learning models for predictive workforce analytics.
- Developed a Text-to-SQL engine enabling business users to query HR data using natural language.
- Developed backend services and REST APIs using Python and FastAPI.
- Worked on vector embeddings, Retrieval-Augmented Generation (RAG), and LLM integration using AWS Bedrock.
- Designed scalable architecture to support dynamic querying, report generation, and analytics workloads.
- Planned, estimated, developed, performed code reviews, and ensured high-quality implementation.
- Dockerized and deployed the solution.
- Participated in release management, production support, troubleshooting, and continuous platform improvements.
- Reduced HR reporting turnaround time from **2–3 days to under 30 seconds** for the majority of reporting requests through conversational analytics.
- Developed predictive models achieving approximately **75% forecasting accuracy** for employee attrition, time-to-start, and time-to-fill forecasting, enabling proactive workforce planning.
- Designed and optimized the platform to support **2,000 concurrent active chatbot users** while maintaining secure multi-tenant isolation and responsive conversational experiences.

5. Recruitment Platform with AI-Powered Candidate Matching (meytier.com)

Technologies Used: Python, Node.js, Angular, MongoDB, Elasticsearch, AWS, Python, FastAPI, AWS Bedrock, NLP, RAG, Twilio, RChilli

Description: Developed an enterprise recruitment platform consisting of a **B2B portal** for employers to manage job postings, recruitment workflows, candidate pipelines, and third-party ATS integrations, and a **B2C portal** for candidates to search, apply, and manage job opportunities. The platform leveraged **Elasticsearch** for high-performance search, **RChilli** for resume parsing, and **Twilio** for candidate communication. The platform was further enhanced with an AI-powered Job and Candidate Matching Engine using NLP, vector embeddings, and similarity scoring to improve recruitment efficiency.

Responsibilities:

- Worked closely with Product Managers, Business Analysts, and senior engineers to understand business requirements and implement scalable solutions.
- Developed backend services using **Node.js** and **MongoDB**.
- Developed Angular-based user interfaces for both employer (B2B) and candidate (B2C) portals.
- Integrated **Elasticsearch** to implement high-performance job and candidate search functionality.
- Integrated **Twilio** for SMS notifications and candidate communication.
- Integrated **RChilli** for resume parsing and candidate profile extraction.
- Integrated with third-party **Applicant Tracking Systems (ATS)** to synchronize candidate and recruitment workflows.
- Developed REST APIs for job management, candidate profiles, employer workflows, and application management.
- Designed and optimized database models and search indexing for scalability and performance.
- Developed an AI-powered Job and Candidate Matching Engine using Python, NLP, vector embeddings, similarity scoring, and AWS Bedrock.

- Developed FastAPI services to expose AI-powered matching capabilities.
- Participated in application design discussions, code reviews, testing, deployment, and production support.
- Worked closely with cross-functional teams to deliver scalable and reliable enterprise software solutions.

6. AI Zoom bot: Meeting summarization and Action Items identification

Technologies Used: Bedrock, transformers,, Bedrock, NLP, Bedrock, RAG, fast API,

Description: The AI-based Zoom bot was developed using the Zoom sdk. It receives the real-time audio in chunks. The audio is converted to text using the speech-to-text Whisper model. The transcription was then used along with the GenAI capabilities of Bedrock to summarize the meeting and identify the action items.

Responsibilities:

- Design and develop the end-to-end application
- Write code in Python and create api using fast api
- Used NLP an Bedrock to match the job against a candidate's CV
- Dockerize and deploy the solution

7. AI-based Angular Unit test generator (Open Source npm package):

Technologies Used: OpenAI, JavaScript, npm

Npm package name: ai-test-gen-angular

Description: ai-test-gen-angular is a powerful tool that leverages OpenAI's advanced capabilities to automate the generation of unit tests for your Angular components and services. This innovative package intelligently analyzes your codebase and produces comprehensive test cases, saving you time and effort in the testing process. Seamlessly integrate it into your Angular projects and watch as it enhances your development workflow by ensuring robust and reliable code.

Responsibilities:

- Innovative package intelligently analyzes your codebase and produces comprehensive test cases, saving you time and effort in the testing process.
- Deployed package on npmjs.org

8. Text to SQL using Machine Learning and AI

Technologies Used: Python, pandas, nltk, transformers, datasets, OpenAI/Bedrock, fast API, FAISS vector store

Database: A stateless tool that connects to any database for functioning

Role: Machine Learning Engineer

Description: It is a Machine Learning Solution to convert user-friendly natural language prompts into intricate SQL queries. Employing advanced Natural Language Processing (NLP) techniques, the project dynamically processes user prompts, intelligently identifies relevant database tables, and extracts their corresponding schemas. Leveraging this enriched information, the system then channels it through a sophisticated machine-learning model meticulously trained for SQL query generation..

By seamlessly integrating NLP and Machine Learning, the solution enables the effortless translation of intricate human language into precise SQL commands, bridging the gap between non-technical users and databases.

Responsibilities:

- Design & Development of the application.
- Write code in Python. Integrate OpenAI with the application.
- Fine-tuning a custom ML model with a dataset of user prompts and complex queries to get better accuracy.

- Dockerize and deploy the solution.

9. Lost Child Detection using Face recognition and street CCTV (Computer Vision):

Technologies Used: Python, PyTorch, face recognition, Dlib,

Description: Developed a sophisticated computer vision solution aimed at locating lost individuals. Utilized Python and PyTorch frameworks alongside face recognition algorithms to create a robust system. The project involved extensive training in the face recognition model on images of missing persons. To enhance accuracy and reliability, a diverse dataset was constructed by applying various image augmentation techniques such as blurring, rotation, and cropping. This augmented dataset ensured the effectiveness of the solution in detecting lost individuals within CCTV footage.

Responsibilities:

- Designed the architecture of the application
- Guide the software engineers under me. Plan, estimate Develop, Code review, and get the flow implementation done right.

Education

BE (Computer Engineering), 2017