Manish Botta

Software Engineering / AI Engineering / Forward-Deployed Engineering / Forward-Deployed AI

San Francisco, CA | manish.botta@hotmail.com
botta.in | github.com/Manishrdy | linkedin.com/in/manish-reddyb

Software engineer with experience building backend platforms, production AI components and enterprise integrations. Combines hands-on implementation with customer discovery, technical tradeoff discussions and production delivery. Experience spans a solo-built B2B SaaS product, retrieval and agent workflows for Microsoft field sales, and conversational-AI product development and customer delivery at Yellow.ai.

TradeLiv

Founding Software Engineer (product and engineering)

Apr 2026 - Present

First hire and sole engineer; product discovery through full-stack delivery.

- Delivered a B2B SaaS platform from zero to production in four months, spanning 40+ REST endpoints and designer, administrator and client workflows, by owning product discovery, architecture and full-stack implementation as the first hire and sole engineer.
- Reduced new e-commerce integration time from two days to under 15 minutes by replacing per-site scrapers with a reusable AI-assisted product-extraction pipeline combining browser rendering and structured product parsing.
- Enabled end-to-end procurement across product curation, client approval, checkout and brand-specific purchase orders by implementing the React/TypeScript frontend, Express APIs and PostgreSQL/Prisma data model.
- Enabled clients to review designer shortlists and receive live updates without repeated page refreshes by implementing token-based client portals and scoped server-sent event streams.
- Enabled payment collection and order progression through Stripe Checkout and webhook-driven processing by connecting approved selections to consolidated orders and brand-level fulfillment records.
- Reduced repeated extraction work within a ten-minute cache window by caching normalized product results and controlling concurrent browser/AI requests.

Earlier retrieval prototype - not current production

Extended an earlier retrieval approach into a TradeLiv prototype, preserving document-section context through heading-based parent/child chunks, by adapting the technique first implemented at Microsoft to TradeLiv's retrieval experiments.

Technologies: TypeScript, React, Vite, Node.js, Express, PostgreSQL, Prisma, Stripe, Puppeteer, AI-assisted extraction, SSE

Microsoft

Software Engineer (Contract)

Aug 2025 - Apr 2026

Owned retrieval, orchestration and reliability components within a team-built field-sales agent. Overall agent: 75,000 daily active users. Owned component/request category: 1.5 million requests over six months.

- Delivered retrieval and orchestration components processing 1.5 million requests over six months within a field-sales agent serving 75,000 daily active users by implementing and extending workflows across CRM, document and SQL data sources on Azure.
- Improved retrieval-context relevance by approximately 30-40% by replacing fixed-size chunks with heading-based parent/child chunks that preserved context across long contract sections.
- Enabled grounded responses to questions about tables and diagrams in long contracts by indexing descriptive metadata in PostgreSQL alongside vector retrieval and connecting retrieved evidence to response citations.
- Supported new sales-agent use cases while retaining context across tool calls by implementing LangGraph nodes, edges, shared state and routing, and extending the existing orchestration graph.
- Bounded agent planning and tool-execution failure paths with timeouts, capped retries, exponential backoff and fallbacks by implementing explicit controls and dead-letter handling for stalled or unsuccessful runs.
- Added policy checks at both request and response boundaries by implementing input validation, prompt-injection checks, scope controls and output guardrails around agent execution.
- Enabled production investigation of divergent agent plans and retrieval behavior through traced execution paths by instrumenting workflows with LangSmith and evaluating contextual relevance and retrieval quality.
- Delivered an RLHF-trained model into production by implementing reward-model training and the reinforcement-learning stage using human feedback.
- Contributed to a final satisfaction score of approximately 75%, assessed against the first three months of operation, by improving retrieval grounding and agent reliability within the team system.

Technologies: Python, Azure, LangGraph, LangChain, LangSmith, PostgreSQL, SQL, Pinecone, BGE/Qwen embeddings, RAG, reranking, RLHF, CRM integrations

Yellow.ai

Software Engineer

Sep 2021 - Jul 2023

Backend/services and enterprise delivery for the first nine months, followed by product engineering. Worked with 25+ enterprise customers on a platform encompassing approximately 50 integrations.

- Delivered enterprise chatbot solutions for 25+ customers by translating customer requirements into platform features, implementing integrations and coordinating launches with product and customer-success teams.
- Enabled director-level stakeholders to choose implementation approaches against cost, delivery time and resource constraints by conducting discovery discussions and presenting architecture options in accessible diagrams and explanations.
- Improved intent-classification accuracy by 35% relative to the baseline and reduced unrecognized inputs or fallback rates by 60% in owned components by improving intent/entity handling and adapting open-source models with LoRA.
- Improved CSAT by 35% relative to the baseline by strengthening query routing across FAQs, document knowledge and automated workflows, with nearest-match suggestions and live-agent handoff when required.
- Enabled goal-oriented conversations through both product implementation and enterprise rollout by building and delivering LLM-driven Dynamic Chat capabilities.
- Enabled website- and document-based chatbot setup through both product implementation and enterprise rollout by building and delivering Zero Setup ingestion/retrieval capabilities.
- Supported a platform processing approximately 50-100 million daily product events across customers by refining Kafka partitioning and workload distribution for high-traffic customer segments.
- Reduced API response time from 820 ms to 490 ms by optimizing MySQL, PostgreSQL and MongoDB queries and caching hot reads in Redis.
- Added validation and fallback controls across incoming queries, generated responses and agent handoff by implementing input checks, output checks and escalation handling around LLM-driven workflows.

Technologies: Node.js, Python, Kafka, MySQL, PostgreSQL, MongoDB, Redis, BERT, spaCy, LoRA, intent classification, entity recognition, conversational AI, CRM integrations

Cognizant

Program Trainee Analyst, Full Stack Developer

Mar 2021 - Aug 2021

Internal banking proof of concept in a team delivery environment; hands-on service development, testing, deployment and monitoring.

- Reduced response time from 1.2 seconds to 700 ms in an internal banking proof of concept by helping decompose a Java monolith into four Spring Boot microservices.
- Achieved 85% code coverage in the proof of concept by writing unit tests and carrying out regression, white-box and black-box testing.
- Enabled independent deployment and operational monitoring of four services by deploying them on AWS and configuring CloudWatch monitoring and automated alerts.

Technologies: Java, Spring Boot, Spring MVC, H2, MySQL, AWS EC2, S3, Elastic Beanstalk, CloudWatch

Selected projects

Coldstart - github.com/Manishrdy/coldstart

A complete local workflow from public ATS data to a ranked review surface, with explicit spending limits, retry behavior, and an audit trail.

Role Collector - github.com/Manishrdy/role-collector

An end-to-end discovery, extraction, storage, and review workflow across six ATS connectors, with configurable Langfuse tracing.

SimCricketX - github.com/Manishrdy/SimCricketX

Served 13,000 active users and resolved a 120-delivery match in under five seconds.

Async Audio Pipeline - github.com/Manishrdy/async-audio-pipeline

Reduced turnaround for a one-hour meeting from 45-75 minutes to 7-11 minutes.

Education

M.S. Computer Science - California State University, San Bernardino | Aug 2023 - May 2025

B.Tech Computer Science - Amrita Vishwa Vidyapeetham, Coimbatore, India | Jul 2017 - May 2021