Prathmesh Jain

Full Stack AI Engineer | Agentic AI · Multi-Agent Systems · RAG | FastAPI · React

Mumbai, IN | +91 8657376595 | prathmeshjain42@gmail.com | LinkedIn | GitHub | LeetCode | Portfolio

PROFESSIONAL SUMMARY

Full Stack AI Engineer with production experience building AI systems for BFSI workflows at Tata Consultancy Services. Specializes in agent orchestration, multi-agent workflows, retrieval-augmented generation (RAG), and scalable full-stack applications using FastAPI, React, and modern LLM frameworks. Delivered platforms that reduced compliance investigations from 1–2 days to under 5 minutes and payment message repair to under 30 seconds.

TECHNICAL SKILLS

AI / LLM	LangChain, LangGraph, CrewAI, RAG, RAG Evaluation, Vector Search, Tool Calling, Multi-Agent Systems, Agent Orchestration, Model Context Protocol (MCP), Structured Outputs, Azure OpenAI, GPT-4o, GPT-5.1, Gemini API, Mistral OCR, Langfuse (Observability)
Languages	Python, TypeScript, JavaScript, Java
Backend	FastAPI, Node.js, Express.js, Flask, Fastify, REST APIs, Microservices, Asynchronous Processing, Server-Sent Events (SSE)
Frontend	React.js, Next.js, React Native, Redux Toolkit, Zustand, Tailwind CSS
Databases	MongoDB, PostgreSQL, Qdrant, Redis
Cloud & DevOps	Docker, AWS (EC2), Oracle Cloud, Cloudflare, Dokploy, Git
Tools & APIs	Playwright, Postman, FastMCP

PROFESSIONAL EXPERIENCE

Tata Consultancy Services (TCS)

Oct 2024 – Present

Full Stack AI Engineer (Research & Innovation, CTO Team, BFSI Americas)

Mumbai, IN

AI Compliance Screening Platform | FastAPI, React, LangChain, Playwright, MCP, AWS

- Built an end-to-end AI compliance screening platform automating KYC onboarding, sanctions screening, embargo verification, customer presence validation, association checks, and adverse media analysis, reducing investigation time from 1–2 days to under 5 minutes.
- Engineered stateful agent workflows combining Google Search, Playwright-based web crawling, structured outputs, deterministic validation, fuzzy matching, and LLM reasoning to improve screening accuracy while minimizing hallucinations and false positives.
- Developed reusable FastMCP servers exposing search, crawling, and document-processing capabilities as standardized tools across multiple AI workflows, alongside asynchronous processing pipelines with Server-Sent Events (SSE) for real-time workflow streaming.

GenAI Payment Repair System | FastAPI, LangChain, CrewAI, React

- Developed a multi-agent payment repair platform where repair and validation agents collaboratively analyzed, repaired, and validated MT/MX payment messages through iterative tool execution, recursive repair loops, and schema-driven validation.
- Reduced average repair time to under 30 seconds, decreasing maker-reviewer feedback cycles by approximately 30%; solution is expected to automate nearly 70% of repair cases through straight-through processing (STP).
- Implemented workflow reliability using retries, fallbacks, persisted workflow state, human review checkpoints, rollback handling, and Langfuse observability for tracing, latency analysis, evaluation, and token monitoring.

Commercial Lending AI / Credit Memo Generation | CrewAI, LangChain, FastAPI, React, Azure OpenAI, PostgreSQL

- Architected a multi-agent system with specialized agents for document parsing, extraction, financial spread analysis, and credit memo population to automate credit memo generation for large US financial institutions; validated across GPT-4o, GPT-5.1, and Gemini 2.5 Flash.
- Processed 300+ financial reports averaging 100+ pages each using parallel chunking, retrieval, structured extraction, and section-wise synthesis to populate standardized credit memo templates, with MCP tools for live web scraping and Mistral OCR.
- Built a custom RAG evaluation pipeline achieving 97% faithfulness, 95% hallucination resistance, and 95% answer relevance; scores published live to Langfuse with per-session cost, latency tracking, and production tracing.
- Measured in a prominent US bank's environment: reduced credit research time by 50% and operational cost by 40%.

Cross-Project Engineering

- Built concurrent processing pipelines (asynchronous programming, multithreading) and React applications with authentication, RBAC, and document uploads; containerized with Docker and deployed on AWS EC2.
- Collaborated with business stakeholders, solution architects, and domain experts from requirement analysis through deployment and production support.

KEY PROJECTS

BizAssist – AI Business Finance Assistant

Live Demo | GitHub

React, FastAPI, LangGraph, Qdrant, MongoDB

- Built a full-stack AI business assistant supporting document Q&A, uploaded file analysis, Google Sheets automation, and multi-step business workflows using LangGraph, React, FastAPI, MongoDB, and Qdrant.
- Designed a stateful multi-agent LangGraph workflow with chat, planner, executor, summarization, guardrails, interrupt-based clarification, and shared tool execution to handle both conversational and complex business tasks.
- Implemented a RAG pipeline using Qdrant for semantic retrieval over user documents (PDFs, text files, invoices, reports, contracts, image-based documents) with user-scoped filtering.
- Built a semantic tool discovery system indexing tool metadata in a persistent vector database, dynamically loading only relevant tools at runtime to reduce prompt context from ~3000 tokens to under 1000 tokens.
- Developed document ingestion pipelines with PDF extraction, multimodal OCR, and chunking, alongside real-time SSE streaming of token generation, tool activity, planner/executor status, and Google Sheets integration.

SmartKiranaBot – Telegram AI Operations Agent

Live Demo | GitHub

TypeScript, Node.js, Fastify, Vercel AI SDK, PostgreSQL, Drizzle

- Engineered an end-to-end, Telegram-first AI agent for retail operations, executing Zod-validated tool calls across inventory, draft billing, GST line calculations, customer khata, and analytics.
- Enforced backend guardrails with atomic non-negative stock updates in Drizzle/PostgreSQL and conditional service locks to prevent overselling and inventory race conditions.
- Implemented multi-turn bill editing, webhook update idempotency, and durable shop memory persisting cross-session settings across conversation resets.
- Built artifact generation tools to render and deliver GST invoice PDFs and analytical PowerPoint decks directly within the Telegram chat pipeline.

Transcribe – Lecture Transcription Workspace

Live Demo

React.js, FastAPI, Redis, PostgreSQL, Gemini API, Langfuse, Docker, Oracle Cloud, Cloudflare

- Built a niche AI audio transcription service for spiritual content, a genre where general-purpose tools fail, converting long-form discourses into structured, searchable, downloadable PDF transcripts; responsive React interfaces for authentication, uploads, history, and live processing status.
- Designed a 3-microservice architecture (Backend API, CPU Worker, IO Worker) communicating over Redis queues; the CPU Worker chunks audio into 20–25 minute segments (FFmpeg conversion, silence detection) and the IO Worker transcribes each chunk via Gemini API and assembles the final PDF, with orphan-task recovery and DB failover for Redis outages.
- Optimized transcription for Sanskrit shlokas and multilingual pravachans using Gemini, followed by AI-powered transcript aggregation into structured HTML and automated PDF generation for 50+ active users (25+ daily active); 200+ files transcribed, averaging 1.5 hours each at 90%+ accuracy.
- Reduced cost per transcription to $0.10–$0.15, 75% cheaper than leading market alternatives.
- Hosted on Oracle Cloud (Dockerized) with the frontend on Cloudflare Pages and Cloudflare DDoS protection on both tiers; a daily evaluation pipeline publishes accuracy metrics to a self-hosted Langfuse instance.

ExpenseGauge – Expense Tracking App

Live Demo | GitHub

React Native, Node.js, MongoDB

- Developed a cross-platform, offline-first expense management app with optimistic updates and background sync queues that reconcile locally stored transactions once connectivity is restored.
- Built secure authentication using short-lived JWT access tokens and rotating refresh tokens, admin dashboards, and analytics for monthly spending trends and category-wise breakdowns.

KEY ACHIEVEMENTS

Compliance Screening	Reduced investigation time from 1–2 days to under 5 minutes with the AI compliance screening platform
Payment Repair Speed	Average MT/MX payment repair time under 30 seconds with multi-agent repair and validation
Review Efficiency	~30% fewer maker-reviewer feedback cycles; expected to automate nearly 70% of repair cases via STP
Credit Research Impact	Reduced credit research time by 50% and operational cost by 40% in a prominent US bank's environment
Document Scale	Processed 300+ financial reports averaging 100+ pages each for credit memo generation
RAG Quality	97% faithfulness, 95% hallucination resistance, and 95% answer relevance on a custom evaluation pipeline
Prompt Optimization	Cut prompt context from ~3000 to under 1000 tokens using semantic tool discovery in BizAssist
Transcription Scale	50+ active users; 200+ files (avg. 1.5 hours each) transcribed at 90%+ accuracy
Cost Efficiency	$0.10–$0.15 per transcription, 75% cheaper than leading market alternatives

EDUCATION

Vidyalankar Institute of Technology, Mumbai, IN **May 2024**

Bachelors in Computer Engineering | GPA: 9.2 / 10