**Sai Krishna Mohan Kolla  **

**Sr. Full Stack Developer**

**<krishnamohank1501@gmail.com> [LinkedIn](https://www.linkedin.com/in/sai-krishna-mohan-k/) [Portfolio](https://full-stack-skm.vercel.app/) +1 251-721-7855**

**Professional Summary:**

* Senior Full Stack Developer with 10+ years of experience building scalable, secure, and high-performance applications across insurance, fintech, e-commerce, and retail domains.
* Specialized in **MERN stack** (MongoDB, Express.js, React.js, Node.js), with deep expertise in JavaScript, TypeScript, and API-centric architecture.
* Designed and implemented full-featured **claim processing portals, robo-advisors, e-commerce platforms, and budgeting systems** used by millions of users.
* Built dynamic, responsive UIs using **React.js**, **Material UI**, and **Tailwind CSS**, ensuring accessibility, SEO, and optimal performance on desktop and mobile.
* Created and maintained **RESTful APIs and microservices** using Express.js, secured with JWT and OAuth2, supporting role-based access control and multi-tenant logic.
* Engineered **complex backend workflows** including flavor-based inventory, recurring budget scheduling, claims routing, and automated investment calculations.
* Integrated third-party services like **AWS S3/SES**, **Cloudinary**, **Stripe**, **Alpha Vantage**, and **OAuth providers**, streamlining features with external APIs.
* Expert in **DevOps pipelines** using GitHub Actions, Docker, and AWS, automating CI/CD for both staging and production environments across EC2, Vercel, and Render.
* Applied schema design, aggregation, indexing, and optimization techniques across databases like **MongoDB**, **PostgreSQL**, **MySQL**, and **Firebase**.
* Wrote **unit, integration, and E2E tests** using Jest, Supertest, and Cypress, achieving 85–90% test coverage across multiple large-scale projects.
* Delivered data-driven UI components using tools like **Chart.js** and implemented dynamic forms and validations using **Formik**, **Yup**, and **React Hook Form**.
* Managed file handling, uploads, and storage through **Multer**, **signed URLs**, and **cron jobs** for cleanup, archival, and scheduled triggers.
* Used Swagger/Postman/Insomnia to maintain API documentation and testing standards, enabling smooth collaboration with QA and frontend teams.
* Collaborated with cross-functional teams using Agile (Scrum), managing stories in **JIRA**, documentation in **Confluence**, and wireframes in **Figma/Lucidchart**.
* Mentored junior developers, reviewed pull requests, and led sprint demos and design reviews, contributing to architectural decisions and delivery velocity.
* Known for **end-to-end ownership** of features—from backend logic and DB schema to frontend integration and deployment with strong focus on clean code, scalability, and long-term maintainability.

**Technical Skills:**

#### **Languages & Programming:** JavaScript (ES6+), TypeScript, HTML5, CSS3, Python (Scripting), SQL

#### **Frontend:** React.js, Next.js, Redux Toolkit, React Router, Material UI, Tailwind CSS, Bootstrap, Formik & Yup. React Hook Form, Chart.js, EJS (for older templates), jQuery (legacy systems).

#### **Backend: N**ode.js, Express.js, REST APIs, GraphQL (if applicable in future roles), JWT & OAuth 2.0, RBAC (Role-Based Access Control), API Rate Limiting & Middleware, File Uploads (Multer), Email (Nodemailer, AWS SES).

#### **Databases:** MongoDB + Mongoose, PostgreSQL, MySQL, Firebase, Redis (consider mentioning if used).

#### **DevOps & Cloud:** AWS (EC2, S3, SES, IAM, Lambda, RDS, CloudFront, CloudWatch), Docker & PM2, Vercel, Render, DigitalOcean, GitHub Actions & GitHub Workflows, Nginx (proxy setup), Terraform (basic).

#### **Testing:** Jest, Supertest, Cypress, Mocha & Chai, Postman, Swagger, Insomnia (for manual testing).

#### **CI/CD & Monitoring:** GitHub Actions, GitHub Webhooks, Sentry, AWS CloudWatch, JMeter (load testing).

#### **Collaboration & Project Management:** JIRA, Confluence, Notion, Lucidchart, Figma, Draw.io, Agile (Scrum)

**Work Experience:**

**USAA - Colorado Springs, Colorado June 2024 - Present**

**Senior Full Stack Engineer | Financial - Insurance**

**Project Overview:**

The project aimed to modernize USAA’s claims processing system by building a secure, scalable web portal that supports policyholder onboarding, multi-step claim submission, document uploads, and real-time status tracking. The application had to support millions of active users, integrate with legacy Spring Boot services, and comply with strict insurance data governance standards.

**Frontend Development:**

* Built a dynamic single-page application using **React.js**, integrated with JWT-based session control and multi-role views.
* Designed responsive UI layouts using **Tailwind CSS** and **Material UI**, following accessibility best practices (ARIA roles, keyboard navigation).
* Developed modular form components with **Formik** and **Yup**, including validation for document attachments, claim reasons, and timelines.
* Implemented a claim progress tracker UI with dynamic status updates based on backend events.
* Managed file previews (PDF, image, DOCX) using file reader APIs and conditionally rendered components.
* Built role-based dashboards showing personalized claim summaries for policyholders, adjusters, and admins.
* Used **React Router** for authenticated navigation and lazy-loaded pages for better performance.

**Backend Development:**

* Developed RESTful APIs with **Node.js** and **Express.js** for claim intake, document linking, user session management, and notifications.
* Integrated with legacy **Java/Spring Boot** microservices via secure API gateways to fetch policy metadata and claim history.
* Created MongoDB schemas for storing claim lifecycle data, file references, user activity logs, and audit trails with timestamps.
* Applied **Joi** and custom middleware for backend input validation, rate limiting, and secure error formatting.
* Developed logic for role-based workflow triggers, including status transitions (Submitted → In Review → Approved/Rejected).
* Added support for temporary claim drafts, allowing users to resume incomplete submissions.
* Reduced claim processing time by 30% through backend API optimization and intuitive frontend workflows.

#### **DevOps & Deployment:**

* Created GitHub Actions workflows for linting, testing, and continuous deployment to **AWS EC2**.
* Set up deployment environments using .env files and **AWS IAM roles** to control access to sensitive services like SES and S3.
* Provisioned and secured **AWS S3** buckets for claim attachments with signed URL generation and file type validation.
* Worked with DevOps to containerize the back-end using **Docker** for development parity.
* Implemented environment-specific logging and monitoring with **CloudWatch** and integrated log streams with Kibana.
* Managed cron jobs for stale draft purging and claim escalation reminders using Node-cron.

#### **Testing & QA:**

* Developed unit and integration test suites for all backend routes using **Jest**, achieving >85% code coverage.
* Used **Supertest** to simulate end-to-end API scenarios including multi-role login, claim routing, and failure cases.
* Coordinated with QA team to build automated regression test scripts with **Postman** and shared environments.
* Logged and triaged bugs using **JIRA**, attending daily triage meetings to prioritize test failures.
* Validated file upload flows against edge cases like unsupported formats, large file sizes, and duplicate documents.
* Used **Swagger** to keep API documentation in sync with codebase, enabling QA and frontend to align quickly.
* Performed smoke testing post-deployment and verified via CloudWatch log monitoring.

#### **Collaboration & Agile:**

* Led sprint ceremonies including backlog grooming, story point estimation, and retrospective discussions.
* Created technical design documents and system flow diagrams using **Lucidchart** and **Confluence**.
* Reviewed PRs from junior developers, focusing on error handling, API consistency, and code readability.
* Interfaced with USAA security team to ensure encryption-at-rest and encryption-in-transit for all sensitive data.
* Coordinated with UI/UX designers to validate Figma prototypes and ensure pixel-perfect implementation.
* Conducted feature demos for product owners and internal stakeholders bi-weekly.

**Tech Stack:**  
React.js, Node.js, Express.js, MongoDB, Spring Boot (integration), JWT, OAuth2, Material UI, Tailwind CSS, AWS EC2, S3, SES, IAM, Joi, Formik, Yup, Docker, GitHub Actions, Swagger, CloudWatch, Postman, Jest, Supertest, Lucidchart, JIRA, Confluence, Agile (Scrum)

**SoFi - San Francisco, California August 2023 - May 2024**

**Senior Full Stack Developer |Fintech / Wealth Management**

#### **Project Overview:**

Worked on SoFi’s Automated Investing (Robo-Advisor) platform to build internal modules and dashboards that empowered users to track spending, align budgets with investment goals, and manage recurring financial activities. The platform supported real-time portfolio summaries, goal-based planning, CSV imports, and chart-driven insights, aimed at improving user financial health and long-term investing behavior.

#### **Frontend Development:**

* Built an interactive React.js dashboard to help users visualize spending vs. investing patterns and asset performance over time.
* Used Redux Toolkit for managing global financial state across budgeting, goal progress, and asset summaries.
* Integrated **Chart.js** to render pie charts, historical investment trends, and net worth changes.
* Created responsive UI components with **Tailwind CSS**, optimized for mobile and tablet.
* Developed editable budget forms and goal planners using **Formik** and **Yup** for dynamic validations.
* Implemented conditional rendering for users based on KYC and investment onboarding status.
* Enabled drag-and-drop goal prioritization and inline edits for recurring expenses.

#### **Backend Development:**

* Built secure RESTful APIs with **Node.js** and **Express.js** for transaction logging, income analysis, and asset tracking.
* Designed **MongoDB schemas** for wallets, categorized expenses, savings goals, and historical performance logs.
* Fetched and normalized real-time market data using **Alpha Vantage API**, applying rate-limited fetch queues and caching layers.
* Ensured financial accuracy in dynamic portfolio calculations through strict input validation and audit logging.
* Implemented cron-based logic for recurring deposits, subscriptions, and budget carry-forwards.
* Built upload functionality for CSV files with custom column mapping, format checks, and bulk inserts.
* Secured endpoints with **JWT**-based auth, refresh token rotation, and scoped access for dashboard APIs.

#### **DevOps & Deployment:**

* Deployed frontend to **Vercel** and backend to **Render**, with environment-specific build configurations.
* Used **GitHub Actions** for CI/CD, including test runs, linting, and auto-deploy pipelines.
* Managed sensitive keys via GitHub secrets and .env encryption.
* Integrated **AWS S3** for secure storage of uploaded CSVs and user documents with signed URL logic.
* Collaborated on Docker-based dev containers to reduce onboarding time across dev machines.
* Conducted load testing with JMeter to identify API bottlenecks under simulated traffic.

#### **Testing & QA:**

* Developed unit and integration tests using **Jest** and **Supertest**, targeting 90%+ backend code coverage.
* Built Cypress test suites for investment dashboard workflows, including goal creation and data sync.
* Wrote custom test cases for malformed CSV data, missing fields, and auth edge cases.
* Used Postman for manual API testing and shared collections with QA team.
* Collaborated with QA and PMs for UAT testing in staging environments before every sprint release.

#### **Collaboration & Agile:**

* Worked closely with SoFi’s product and compliance teams to align budgeting tools with robo-advisor flows.
* Participated in sprint planning, demos, and retrospectives with cross-functional squads.
* Documented API contracts in Swagger and created flow diagrams using Lucidchart.
* Mentored two junior developers on Express.js routing patterns and Redux logic.
* Converted Figma mockups into pixel-perfect, accessible components.
* Maintained feature and bug tracking using **JIRA** and documentation in **Confluence**.

**Tech Stack:**  
React.js, Redux Toolkit, Node.js, Express.js, MongoDB, Mongoose, JWT, Alpha Vantage API, Chart.js, Tailwind CSS, Vercel, Render, AWS S3, Formik, Yup, CSV Parser, Cypress, Jest, Supertest, Postman, Docker, GitHub Actions, Swagger, Lucidchart, JIRA, Confluence, Agile (Scrum)

**Lowe’s - Charlotte, NC May 2019 - July 2023**

****Full Stack Developer | Retail****

#### **Project Overview:**

#### Lowe’s launched a tailored e-commerce portal focused on B2B customers like contractors and construction professionals. This initiative included advanced catalog search, inventory tracking, and multi-location cart support. I contributed end-to-end to build an internal product and customer management dashboard, including authentication, product deals engine, and flavor-based inventory logic.

#### **Frontend Development:**

* Developed reusable components for contractor dashboard, including order history, scheduled pickups, saved lists, and real-time inventory views using **React.js** and **Material UI**.
* Integrated **React Router** and **Protected Routes** for seamless navigation between user-level and admin-level dashboards.
* Implemented dynamic product filtering by store location, category, stock status, and promotional tags using Redux Toolkit and backend query hooks.
* Designed responsive product cards with deal badges, flavor selectors, and inventory indicators.
* Used **Formik** and **Yup** for contractor registration forms with license uploads and multi-step onboarding.
* Added global Snackbar system for status alerts, confirmations, and error feedback.
* Built flavor-aware quantity selectors and purchase limit enforcement logic in frontend.

#### **Backend Development:**

* Developed RESTful APIs in **Node.js** and **Express.js** for catalog management, stock validation, pricing logic, user management, and order tracking.
* Designed **MongoDB schemas** to handle both standard and flavor-based SKUs with embedded variants, dynamic price fields, and discount metadata.
* Implemented secure login flows using **JWT**, with role-based access (user, admin, guest) and OAuth integration for single sign-on.
* Created admin APIs to manage bulk product uploads, toggle product visibility, and monitor low-stock alerts.
* Integrated **AWS S3** and **Cloudinary** for high-resolution product image uploads and real-time optimization.
* Added discount rule engine that supports fixed, percentage, and conditional price drops with backend triggers.
* Created order pipelines for cart-to-checkout with status tracking and estimated delivery slots.

#### **DevOps & Deployment:**

* Deployed backend to **Render** and frontend to **Vercel**, separating staging and production environments with preview builds.
* Integrated **GitHub Actions** for CI/CD workflows with automated linting, tests, and environment syncing.
* Managed infrastructure secrets and deployment configs using .env, GitHub Secrets, and Render dashboard.
* Utilized **AWS CloudFront** for CDN distribution of product media.
* Automated backup snapshots for MongoDB clusters and used indexing strategies for faster lookup.

#### **Testing & QA:**

* Wrote Jest test suites and API integration tests with Supertest for product logic, deals, and cart flows.
* Built **Cypress E2E tests** for contractor dashboard journeys including inventory lookup, add-to-cart, and checkout.
* Created Postman collections for regression testing on staging environments.
* Validated CSV upload logic for product data entry with various edge-case scenarios.
* Collaborated with QA to write test plans based on Jira stories and acceptance criteria.
* Logged and tracked bugs, performance metrics, and flaky flows using **Sentry** and **Chrome DevTools**.

#### **Collaboration & Agile:**

* Partnered with Lowe’s product owners, backend architects, and UI/UX team to map user flows and implement logic-heavy features.
* Conducted weekly demos, UAT sessions, and participated in backlog grooming.
* Created feature diagrams and workflow specs in **Lucidchart** and **Figma**.
* Onboarded new developers with code walkthroughs and architecture deep dives.
* Maintained tech documentation and API contracts in Swagger and Confluence.

**Tech Stack:**  
React.js, Redux Toolkit, Node.js, Express.js, MongoDB, Mongoose, JWT, OAuth2, AWS S3, Cloudinary, Material-UI, Tailwind CSS, Formik, Yup, Axios, Postman, Jest, Cypress, Supertest, Vercel, Render, GitHub Actions, Sentry, Swagger, Lucidchart, Figma, Agile (Scrum)

**The Hartford - Chicago, Illinois December 2016 - April 2019**

**Full Stack Developer | Insurance / Claims Management**

#### **Project Overview:**

Developed a scalable digital platform to streamline the intake, review, and approval of insurance claims. The system supported end-users (policyholders), internal reviewers, and admins, with built-in workflows for document management, claim escalation, and notifications. The application improved claims processing speed and compliance reporting across departments.

#### **Frontend Development:**

* Built modular interfaces using **React.js** with role-based dashboards for users, reviewers, and claim officers.
* Developed intake forms for FNOL (First Notice of Loss) submissions with support for multiple file attachments and conditionally rendered fields.
* Used **Redux Toolkit** for state management across claims list, filters, and audit logs.
* Integrated dynamic filtering and claim search with API-driven pagination and sorting.
* Designed approval workflows with color-coded status indicators and real-time field validation via **Formik** and **Yup**.
* Ensured full mobile responsiveness and accessibility compliance (WCAG) using **Material UI** and custom ARIA attributes.
* Added alert/notification systems for pending approvals, missing documents, and escalations.

#### **Backend Development:**

* Built robust REST APIs in **Node.js** and **Express.js** for managing claim submissions, document attachments, reviewer actions, and audit trails.
* Implemented **RBAC** logic via JWT tokens with permission-based middleware for fine-grained access control.
* Structured **MongoDB** collections with relationships between policies, claims, and documents using **Mongoose**.
* Developed pipelines for claim lifecycle states with rollback support.
* Integrated **AWS S3** for storing uploaded documents like invoices, reports, and ID proofs with signed URL access.
* Enabled reviewer comments, status updates, and escalation notes with timestamped entries and change history.
* Added pre-validation and sanitization layers using **Joi** to ensure clean and secure data intake.

#### **DevOps & Deployment:**

* Managed application deployment using **DigitalOcean** droplets with Nginx reverse proxy setup.
* Used **PM2** for process management and monitoring of Node.js services.
* Wrote shell scripts for automated backup of claims data and asset snapshots.
* Configured CI/CD pipelines with **GitHub Webhooks** for deployment and logs syncing.
* Managed environment variables securely using .env and restricted file-level access via IAM roles.
* Ensured high availability through scheduled service restarts and Docker containerization for isolated testing.

#### **Testing & QA:**

* Created unit tests for claim lifecycle logic using **Jest** and **Mocha**, with mocking libraries.
* Automated regression tests on file upload and approval workflows using **Cypress**.
* Used **Postman** and **Swagger** to manually test and document API behaviors.
* Logged test results and edge cases using JIRA integration.
* Worked with QA team to simulate high-volume claim submissions for load testing.
* Set up alerts for backend errors and frontend exceptions using **Sentry** and **AWS CloudWatch**.

#### **Collaboration & Agile:**

* Participated in daily standups, retrospectives, and sprint demos with business analysts and product owners.
* Collaborated with legal and compliance teams to ensure claim workflows met industry standards.
* Wrote onboarding documentation for developers and maintained architectural diagrams in **Draw.io**.
* Delivered feature specs and coordinated releases through **JIRA** and **Confluence**.

**Tech Stack:**  
React.js, Redux Toolkit, Node.js, Express.js, MongoDB, Mongoose, JWT, AWS S3, Joi, Multer, Swagger, Cypress, Jest, Mocha, PM2, DigitalOcean, Docker, GitHub, GitHub Webhooks, Postman, Sentry, Draw.io, JIRA, Agile (Scrum)

**GAP Inc - New York, NY August 2015 - November 2016**

**Junior Full Stack Developer | Retail / E-commerce**

#### **Project Overview:**

Contributed to the development of an internal budgeting and expense tracking tool used by GAP Inc.'s regional retail managers. The system helped monitor in-store operational costs, promotional budgets, travel expenses, and quarterly spend analysis. Focused on building data pipelines, permission-based dashboards, and secure upload/reporting workflows.

#### **Frontend Development:**

* Built responsive web interfaces using **EJS templating** and **Bootstrap**, supporting real-time expense entry and approval tracking.
* Created role-specific dashboards (store manager, finance lead) with permission-based component rendering.
* Implemented filterable tables for transaction history using **DataTables** and pagination logic.
* Added dynamic modals for uploading receipts, editing entries, and attaching comments.
* Ensured UI consistency across devices with mobile-friendly layouts and print-friendly styles.
* Used **jQuery** and **vanilla JS** for frontend logic like field validations, date pickers, and status toggles.
* Helped implement dynamic budget sliders and visual feedback for remaining limits.

#### **Backend Development:**

* Developed RESTful APIs using **Node.js** and **Express.js** for submitting expenses, uploading receipts, and exporting quarterly reports.
* Integrated **MongoDB** with Mongoose schemas for expense categories, store metadata, users, and file references.
* Implemented role-based access with **JWT** and built middleware to restrict sensitive endpoints (e.g., approvals, exports).
* Built cron jobs to send end-of-month reminders and auto-generate financial summaries.
* Developed CSV export endpoints with custom column mapping and time-based filters.
* Optimized queries for store-region filtering, and added compound indexing for reporting performance.
* Used **Dotenv** for environment configs and secure connection strings.

#### **DevOps & Deployment:**

* Deployed the application on **Heroku**, managing staging and production environments.
* Used **PM2** for managing background services and graceful restarts.
* Managed Git-based deployments via Heroku pipelines with rollback support.
* Implemented .env-based config management for multi-region deployment.
* Conducted memory profiling using Heroku metrics to optimize usage.
* Created README and setup scripts for internal dev onboarding.

#### **Testing & QA:**

* Tested API endpoints using **Postman** and created test collections for expense flows.
* Wrote unit tests for critical routes using **Mocha** and **Chai**.
* Performed cross-browser and mobile compatibility testing for dashboard UI.
* Partnered with QA to test role-based access, especially for finance-level approvals.
* Documented edge cases around budget limits and file upload size handling.
* Logged defects in **JIRA** and resolved critical issues ahead of quarterly reporting launches.

#### **Collaboration & Agile:**

* Worked closely with backend team and finance SMEs to align APIs with reporting logic.
* Participated in daily standups and retrospectives using Agile (Scrum) processes.
* Attended sprint reviews and demoed completed features bi-weekly.
* Maintained changelog and API usage documentation for cross-team collaboration.
* Used **GitHub** for source control and **JIRA** for tracking tasks and blockers.

**Tech Stack:** Node.js, Express.js, MongoDB, Mongoose, EJS, Bootstrap, jQuery, JWT, Dotenv, PM2, Mocha, Chai, Postman, GitHub, Heroku, JIRA, Agile (Scrum)