Vamsi Krishna

USA | +1 (646) 361-7267 | vamsikrish1200@gmail.com | LinkedIn | github | vamsii.com

Professional Summary

- 10+ years of experience delivering enterprise web applications across airlines, finance, healthcare, and government domains.
- Specialized in React.js, Next.js (SSR/SSG), and modern frontend architectures, driving modular, scalable, and high-performance UIs.
- Implemented Micro Frontend architecture and component-driven design systems, reducing delivery time for new modules.
- Integrated Node.js + Express.js backends with RESTful APIs and GraphQL, improving interoperability and reducing API response times.
- Designed resilient, secure API ecosystems with Azure API Management and AWS API Gateway, improving reliability.
- Engineered secure authentication flows using JWT, OAuth2.0, Firebase Auth, and Auth0, ensuring compliance with enterprise security standards.
- Optimized PostgreSQL/MySQL and MongoDB databases with indexing, stored procedures, and connection pooling, reducing query latency by up to 40%.
- Applied Redux, Recoil, Zustand, and Context API for state management, improving consistency and debugging efficiency.
- Delivered responsive, mobile-first UIs with TypeScript, TailwindCSS, Material UI, and Bootstrap, enhancing accessibility
- Automated quality assurance with Jest, React Testing Library, Cypress, Playwright, and Cucumber (BDD), reducing regression defects by 25–30%.
- Validated and documented APIs with Postman, Swagger UI, and GraphQL Playground, streamlining partner integrations and QA.
- Deployed and managed solutions on Azure and AWS Cloud with Docker and Kubernetes, achieving high availability and zero-downtime rollouts.
- Implemented CI/CD pipelines using Azure DevOps, GitHub Actions, GitLab CI/CD, and Jenkins, cutting release errors by up to 40%.
- Hosted and scaled frontend builds on Vercel and Netlify, enabling globally distributed, zero-downtime deployments.
- Proven track record in Agile Scrum delivery, mentoring teams, and cross-functional collaboration, consistently improving delivery speed and stakeholder satisfaction.

Technical Skills

Languages: JavaScript (ES6+), TypeScript, HTML5, CSS3, SQL, C#
Frontend Frameworks & Libraries: React.js (class & functional components, lifecycle methods, hooks), Next.js (SSR/SSG), Vite Redux, Recoil, Context API, Tailwind CSS, Material UI, Bootstrap, Responsive Design (Flexbox, Grid, Mobile-first), jQuery, Ajax
Backend / Runtime: Node.js, Express.js, RESTful APIs, GraphQL APIs, Microservices Architecture, Java/Spring Boot (integration experience)
Databases: PostgreSQL, MySQL, Microsoft SQL Server, MongoDB, Firebase, T-SQL (procedures, views, triggers, indexing)
Messaging & Event Streaming: Apache Kafka, RabbitMQ
Cloud & API Management: AWS (EC2, S3, RDS, Lambda, API Gateway, CloudFront), Azure (App Service, AAD, APIM, Storage, Functions), Vercel, Netlify
Authentication & Security: JWT, OAuth2.0, Firebase Auth, Auth0, Azure Active Directory (AAD), Role-based Authorization, SSL/TLS, Security Headers
Testing Frameworks: Jest, React Testing Library, Cypress,
CI/CD & DevOps Tools: Azure DevOps Pipelines, GitHub Actions, GitLab CI/CD, Jenkins, Docker, Kubernetes
Version Control & Collaboration: Git, GitHub, GitLab, Bitbucket, JIRA, Confluence, Agile/Scrum, Kanban
API & Development Tools: Postman, Swagger UI, GraphQL

Professional Experience

Delta Air Lines **Nov 2021 - Present**
Senior Full-Stack Developer

- Delivered projects under Agile Scrum methodology with sprint planning, daily stand-ups, and retrospectives, which improved collaboration and accelerated delivery timelines.
- Adopted Micro Frontend architecture and implemented a component-driven design system in React, which enabled modular releases and improved reusability across teams.
- Developed applications using React.js and Next.js with SSR/SSG, which improved page performance and optimized SEO for customer-facing modules.
- Integrated with backend services using Node.js and Express.js, which streamlined server-side rendering and improved API response handling.
- Built and consumed RESTful APIs with contract-first design using Swagger/OpenAPI, which accelerated partner integrations and reduced onboarding time.

- Managed global and local state with Redux Toolkit and React Context API, which improved consistency and reduced state-related defects.
- Standardized data exchange using JSON, which ensured seamless integration across distributed services and reduced serialization issues.
- Secured APIs through Azure API Management, which enabled throttling, monitoring, and improved security compliance.
- Implemented authentication using JWT, OAuth2.0, and Azure AD, which strengthened access control and reduced unauthorized access attempts.
- Built responsive UIs with HTML5, CSS3, ES6+, and TypeScript, styled using TailwindCSS and Material UI, which improved cross-device compatibility and enhanced user satisfaction.
- Optimized PostgreSQL queries, indexes, and stored procedures, which reduced execution times for high-volume workloads by 40% and improved system throughput.
- Increased quality by developing unit tests with Jest and React Testing Library, automating end-to-end tests with Cypress, and applying BDD with Cucumber, which reduced production defects and improved regression coverage.
- Validated APIs with Postman and documented endpoints using Swagger UI, which streamlined testing and improved collaboration with external teams.
- Configured Azure DevOps pipelines with automated builds, static code analysis, and blue/green deployments, which minimized downtime and improved release confidence.
- Delivered applications on Azure Cloud with Docker containerization, which standardized environments and improved deployment reliability.
- Deployed frontend builds on Vercel, which enabled zero-downtime rollouts and improved global performance delivery.
- Applied blue/green deployment strategies, which reduced release risks and ensured high availability during production rollouts.

.

Broadridge **May 2019 - Oct 2021**

Full-Stack Devloper

- Delivered enterprise solutions under Agile Scrum methodology, coordinating sprints, backlog refinement, and demos, which improved transparency and reduced delivery risks.
- Applied Micro Frontend architecture with React-based modules, which enabled independent deployments and accelerated feature delivery by 20%.
- Developed applications using React.js with Next.js for SSR, which enhanced SEO and reduced time-to-first-byte for client-facing portals.
- Integrated Node.js and Express.js services for API orchestration, which streamlined server-side processing and improved response times.
- Designed and consumed RESTful APIs with Swagger-driven contract management, which reduced integration defects and improved partner adoption rates.
- Managed global and module-level state using Redux Toolkit and Recoil, which simplified state handling and reduced boilerplate code.
- Standardized all communication with JSON, which improved debugging and interoperability across distributed services.
- Implemented AWS API Gateway for secure, monitored, and throttled API access, which reduced latency and improved reliability.
- Enforced JWT and OAuth2.0 authentication, which improved endpoint security and ensured compliance with financial industry standards.
- Built responsive, accessible UIs using HTML5, CSS3, JavaScript (ES6+), TypeScript, TailwindCSS, and Bootstrap, which improved mobile usability and increased engagement metrics.
- Optimized PostgreSQL queries, indexes, and connection pooling, which increased throughput and reduced query latency by 30%.
- Enhanced reliability by creating unit tests with Jest and React Testing Library, end-to-end tests with Cypress, and BDD flows with Cucumber, which reduced regression defects by 25%.
- Validated APIs using Postman and documented endpoints in Swagger UI, which accelerated onboarding for developers and external teams.
- Automated build and deployment pipelines using Jenkins, which reduced manual errors by 40% and improved release cadence.
- Delivered solutions on AWS Cloud leveraging Docker-based deployments, which improved scalability and increased uptime by 25%.
- Hosted frontend builds on Vercel, which enabled globally distributed, zero-downtime deployments with CDN-level performance.
- Implemented blue/green deployment strategies, which reduced downtime during releases and increased stakeholder confidence in production rollouts.

State of Ohio **Jan 2017 - Apr 2019**

Senior FrontEnd Developer

- Delivered solutions under Agile Scrum methodology with sprint planning, retrospectives, and backlog grooming, which improved delivery predictability and cross-team collaboration.
- Applied component-driven architecture with React and reusable UI libraries, which standardized design patterns and reduced new feature delivery time by 30%.

- Developed applications using React.js with early adoption of SSR techniques, which improved load times and enhanced SEO for public-facing portals.
- Integrated Node.js + Express.js middleware services, which streamlined API orchestration and reduced integration overhead.
- Built and consumed RESTful APIs with Swagger/OpenAPI documentation, which reduced onboarding time and improved API adoption across partner systems.
- Managed state with Redux and React Context API, which ensured predictable state handling and improved debugging efficiency.
- Standardized data exchange with JSON, which improved cross-system communication and reduced serialization issues.
- Secured APIs using Azure API Management, which enforced rate limiting, monitoring, and consistent access controls.
- Implemented JWT and OAuth2.0 authentication with enterprise role-based access, which improved application security and ensured compliance with state policies.
- Built responsive, accessible UIs with HTML5, CSS3, JavaScript (ES6+), TypeScript, and Bootstrap/Material UI, which improved mobile compatibility and increased accessibility compliance.
- Authored optimized queries and stored procedures in PostgreSQL, which reduced query execution time by 30% and improved database efficiency.
- Strengthened reliability with unit tests in Jest and Enzyme, and implemented end-to-end tests with Cypress, which improved regression coverage and reduced production defects.
- Validated APIs using Postman and published contract documentation with Swagger UI, which accelerated QA cycles and reduced integration errors.
- Configured Azure DevOps pipelines with artifact versioning and automated deployments, which reduced release errors and improved delivery speed.
- Delivered and managed workloads on Azure Cloud App Services with container-based deployments, which improved scalability and reduced downtime.
- Published frontend builds to Netlify, which improved deployment speed and simplified hosting for public-facing portals.
- Applied blue/**green release strategies**, which reduced downtime and ensured smoother production transitions.

Calpine **Sep 2014 - Dec 2016**
FrontEnd Developer

- Delivered enterprise applications under Agile Scrum methodology with sprint planning and daily stand-ups, which improved collaboration and reduced cycle time for feature delivery.
- Adopted component-driven architecture with React (class components + lifecycle methods), which modularized UI design and improved maintainability across the team.
- Developed applications using React.js (v15) with lifecycle methods which ensured predictable rendering and compatibility with legacy modules.
- Integrated backend services using Node.js + Express.js as middleware with data access through Java-based APIs (Spring + Hibernate), which reduced integration complexity and improved maintainability.
- Built and consumed RESTful APIs with contract-first design, using Swagger for documentation and Spring controllers for backend endpoints, which improved onboarding and integration reliability.
- Managed global state using early Redux implementation, which reduced inconsistent data handling and improved debugging.
- Standardized data interchange using JSON, which improved interoperability across Java, Node.js, and React components.
- Secured API traffic with AWS API Gateway, which enabled request throttling, monitoring, and improved application resilience.
- Implemented authentication with JWT and OAuth2.0, which improved security compliance and protected sensitive business data.
- Built responsive frontends using HTML5, CSS3, ES6, TypeScript (early adoption), and Bootstrap, which improved cross-device compatibility and reduced UI defect rates.
- Designed optimized PostgreSQL schemas, indexes, and stored procedures, which reduced query execution time by 25% and improved system efficiency.
- Integrated Apache Kafka for event-driven workflows between React applications and backend services, which reduced latency and improved reliability of asynchronous processes.
- Strengthened quality with unit tests using Mocha and Chai, and component tests with Enzyme, which improved regression reliability and reduced production bugs.
- Validated API behavior using Postman and documented endpoints with Swagger, which accelerated QA validation and reduced integration errors.
- Configured Jenkins CI/CD pipelines with static code analysis and automated deployments, which reduced manual release errors by 35% and improved delivery speed.
- Deployed and managed workloads on AWS (EC2, S3, RDS) with Docker containerization, which improved scalability and ensured high availability.
- Published React frontend builds with AWS S3 static hosting + CloudFront, which reduced latency and improved application load times for global users.

Biocon **Oct 2013 - Jun 2014**
FrontEnd Developer

- Built ASP.NET MVC 4 features with Entity Framework Code-First and fluent mappings, improving scalability, maintainability, and accelerating feature development.
- Developed a responsive React UI to replace legacy views, reducing page load times by 30% and boosting user satisfaction.
- Exposed REST APIs and maintained WCF services for partner integrations, ensuring seamless interoperability and reliable data exchange.
- Authored optimized T-SQL queries, stored procedures, and database objects, improving data retrieval performance by 25% and reducing query latency.
- Implemented JWT-based authentication integrated with Single Sign-On (SSO), strengthening endpoint security and simplifying user access management.
- Wrote NUnit test suites and integrated them into CI builds, reducing manual testing efforts and improving overall release quality.
- Published Swagger documentation with working examples, accelerating API adoption for partner and internal teams.
- Tuned IIS site configuration and enabled compression for static assets, improving performance and cutting page load times by 20%.
- Enhanced UI interactivity with Ajax and deferred data loading, reducing server load and improving application responsiveness.
- Applied robust input validation and anti-forgery tokens, mitigating OWASP security risks and hardening web endpoints.
- Participated in requirements workshops and authored detailed technical design documents, improving delivery clarity and reducing rework.
- Collaborated with QA to design test plans and end-to-end scenarios, increasing coverage and reducing production defects.
- Refactored data access logic with resiliency and retry patterns, improving stability and reducing downtime under transient failures.
- Authored deployment scripts and release checklists, reducing deployment errors and ensuring predictable release cycles.
- Partnered with DBAs on index tuning and statistics updates, which improved query efficiency and increased database reliability.
- Enforced role-based authorization with granular access controls, strengthening compliance and data protection policies.
- Set up Git/SVN repositories and CI jobs, improving collaboration, enabling version control discipline, and streamlining continuous integration.

Education

Mahaveer Institute of Science and Technology, Hyderabad **Mar 2011 - May 2013**

B.Tech, Information Technology