MANOJ KUMAR GADIPALLI

Dallas, TX | (940)-9775738 | gadipallimanojkumar@gmail.com
https://www.linkedin.com/in/manojkumargadipalli/ | https://manojkumargadipalli-portfolio.netlify.app/

EDUCATION

University of North Texas — Denton, TX Jan 2024 – Dec 2025
Master of Science, Information Systems and Technologies
Relevant Courses: Web-Based Systems Development, Database Development, Big Data, Information Systems Development

CMR Institute of Technology — Hyderabad, IN Aug 2018 – Aug 2022
Bachelor of Technology
Relevant Courses: Web Technologies, Database Management System, Java Programming.

TECHNICAL SKILLS

Programming Languages & Frameworks: JavaScript, Python, React.js, Node.js, Express.js
Frontend Engineering: HTML5, CSS3, Tailwind CSS, Redux, React Hooks, Responsive UI, Component Architecture
Backend Engineering: REST APIs, JWT Authentication, Role-Based Authorization, Error Handling, Input Validation
Databases: MongoDB, MySQL (Schema Design, Indexing, Query Optimization)
Testing & QA: Postman, Selenium (TestNG), API Testing, UI Testing
Cloud & Dev Tools: AWS (S3, EC2, Glue), Git, GitHub, CI/CD Fundamentals

PROFESSIONAL EXPERIENCE

Junior Full Stack Engineer — CloudYard | Hyderabad, IN May 2022 – Sep 2023

- Designed and built a scalable role-based file-sharing platform using React, Node.js, Express, and MongoDB, enabling secure document access, secure file exchange workflows, and collaboration for 10–20 internal users across teams.
- Reduced unauthorized access incidents by 11% by implementing JWT-based authentication, role-based access control (RBAC), token validation, and secure session handling, guided by QA defect trends and internal security reviews.
- Refactored frontend architecture into modular, reusable React components and standardized UI state management, reducing redundant code by 20%, improving maintainability, and accelerating future feature development.
- Delivered 2–4 production-ready features per Agile sprint by collaborating closely with frontend, backend, and QA teams, resolving cross-service integration issues, managing dependencies, and ensuring timely, high-quality releases aligned with business requirements.
- Prepared MongoDB query performance through indexing, schema tuning, query restructuring, and performance profiling, improving API stability, lowering response times, and reducing latency-related QA defects by 6%.
- Enhanced backend robustness by improving validation, error handling, and edge case coverage across authentication and file-sharing modules.
- Strengthened backend scalability by optimizing request handling, reducing redundant API calls, lowering server overhead, and improving performance under concurrent usage.

PROJECT EXPERIENCE

Rule-Based Board Game Engine (Ashtachamma) Sep 2022 – Nov 2022

- Implemented a rule-based Ashtachamma game engine in JavaScript (ES6) for 4 players and 16 tokens, implementing turn sequencing, cowrie-based movement, safe-cell logic, capture rules, and win/finish conditions using centralized state management.
- Developed path-driven token traversal (PATH1–PATH4) with Set/Object-based state tracking and DOM event handling, ensuring accurate move validation, turn consistency, and rule enforcement across 10+ scenario-driven gameplay test validations.

User Authentication & E-commerce Management System Apr 2023 – Jun 2023

- Built a secure role-based authentication and authorization system using Node.js, Express.js, JWT, and password hashing, enforcing protected routes and controlled admin/user access across sessions.
- Developed RESTful APIs with Express.js to support user management and e-commerce operations (products, cart, orders), improving backend reliability and reducing UI duplication by 7% through reusable component architecture.

Data Engineering Project: Database Design & Optimization Nov 2024 – Dec 2024

- Modeled and normalized relational database schemas up to BCNF across 4 interrelated tables, enforcing primary/foreign key constraints to improve data consistency and integrity.
- Optimized MySQL query performance by analyzing slow query logs, applying index optimization, refactoring join-heavy SQL queries, and improving execution plans, achieving an estimated 10–15% reduction in query runtime and more consistent database response times.