

K SRINIVASAN

TECHNICAL SKILLS

- Programming Languages : C#, Solidity
- Frameworks : .NET Core, ASP .NET, MVC, React JS, NodeJS, GraphQL, Angular, Linq & Entity Framework
- Web Services : Restful APIs, Web Services
- Microservices & Messaging : Kafka, MQTT, Rabbit mq, SQS, SNS
- Cloud Services : AWS (CloudFormation, Lambda Functions, DynamoDB, S3, RDS, EC2)
- DevOps Tools : Azure DevOps, Git
- Database : SQL, MySQL, NoSQL (MongoDB), DynamoDB

PROFESSIONAL EXPERIENCE

- Experienced .NET Developer with over 5+ years of expertise in C#, .NET Core, and web services.
- Proficient in designing and developing scalable applications using microservices architecture with Kafka and MQTT for seamless communication.
- Integrated AWS services for cloud-based solutions and implemented DevOps practices for automated build and deployment.
- Strong understanding of design principles, solid coding practices, and excellent troubleshooting skills.
- Collaborative team player with effective communication abilities.

Work Experience:

Dotnet developer, Lithi Info Tech:

- Designed and developed scalable and high-performance components using C#, ASP .NET and .Net Core.
- Integrated AWS services, including CloudFormation, ReactJS, NodeJS, DynamoDB, and S3, to build cloud-based solutions.
- Implemented DevOps practices for automatic build and deployment using tools like Azure DevOps and Git. Maintained solid coding practices, adhering to design principles, and regularly contributed to code reviews.
- Created and maintained test cases to ensure application reliability and performance.

Microservices, Devops and Dotnet developer, JLVM Tech:

- Implement and maintain DevOps processes, including continuous integration (CI) and continuous delivery (CD) pipelines, developed data-driven applications using LINQ and Entity
- Collaborate with development and operations teams to automate deployment processes and ensure smooth software releases and maintained microservices with different brokers in .net.
- Utilize React for front-end development, creating user-friendly and responsive interfaces.
- Ensure scalability and high availability of web applications in a cloud environment.
- Use version control systems (Git) to manage source code effectively.

Project Reports:

Project 1: Real-time Chat Application

Description:

Built a real-time chat application using React for the front-end, C# for the back-end, and MongoDB to store chat messages and user data. Used Kafka for real-time messaging and AWS S3 to store multimedia files.

Key Features:

- Real-time messaging with Kafka integration for event-driven communication.
- Secure user authentication and personalized user profiles.
- Efficient storage and delivery of product images and media via AWS S3.
- Used Entity Framework and LINQ for optimized database interactions

Project2: E-commerce Website

Description:

Description:

Developed a robust e-commerce website using React for the front-end and C# for the back-end. Integrated **Kafka** for order processing and messaging between microservices.

Key Features:

- User-friendly product catalogue with detailed descriptions and images.
- Order management and tracking with asynchronous communication using Kafka.
- implemented LINQ queries for data manipulation and efficient querying.
- Recommendations and reviews system for products.

Project 3: Health Records Management System

Project Description:

Develop a Health Records Management System that leverages modern technologies to efficiently manage patient health records and healthcare information. This system will use React for the front-end and C# for the back-end, integrating both a NoSQL database (e.g., MongoDB) and a relational database (e.g., MySQL) for seamless data storage and retrieval.

Key Features and Components:

- **Electronic Health Records (EHR):** Create a secure and user-friendly EHR system for healthcare providers to store and access patient health records, medical history, diagnoses, and treatment plans.
- **Patient Portal:** Develop a patient portal where individuals can view their health records, schedule appointments, and communicate with healthcare professionals securely.
- **Appointment Scheduling:** Implement an appointment scheduling system that enables patients to book appointments online and receive reminders, Optimized database performance using Entity Framework and LINQ.

Project 4: Hybrid Data Management System

Description:

Developed a hybrid data management system in ASP .NET that seamlessly integrates DynamoDB, a NoSQL database, and MySQL, a relational database, using C# for the backend. This system is designed to provide flexibility

in handling both structured and unstructured data, making it suitable for various applications.

Key Features and Components:

- **Database Integration:** Created a unified API layer in C# that interacts with both DynamoDB and MySQL seamlessly. The system determines the data storage location dynamically based on data models and user-defined configurations.
 - **Microservices Architecture:** Implemented Kafka as an event streaming platform for real-time data processing and communication between microservices.
 - **Caching and Performance Optimization:** Utilized Redis as a caching layer to improve data retrieval performance and reduce database load.
 - **Message Broker:** Integrated MQTT as a lightweight messaging protocol to handle real-time communication for IoT data and asynchronous messaging between services.
-

Project 5: Audit Logix

Description:

Audit Logix is a computer-assisted coding (CAC) software developed for Risk Adjustment Auditing in ASP .NET, leveraging the domain expertise of risk adjustment auditing experts.

Key Features and Components:

- **Automated Analysis:** Automatically analyzes clinical information and provides HCC suggestions to auditors, enhancing productivity and efficiency.
 - **Performance and Reliability:** Built with ASP .NET and C# to ensure robust performance and reliability.
 - **Event-Driven Architecture:** Implemented Kafka for event streaming and asynchronous communication between various auditing modules, enabling real-time processing of clinical data.
 - **Distributed Caching:** Utilized Redis to cache frequently accessed data, reducing latency and improving the performance of the auditing system.
 - **Message Queue Integration:** Integrated RabbitMQ as a message broker to manage task queues for background processing and communication between microservices.
-

Project 6: Healthcare Information Management System

Description:

Developed a Healthcare Information Management System using a combination of modern technologies. Utilized Angular and React for the front-end and C# for the back-end, while integrating both a NoSQL database (MongoDB) and a relational database (MySQL) to efficiently manage healthcare data.

Key Features and Components:

- **Electronic Health Records (EHR):** Implemented a secure and user-friendly interface for managing electronic health records, allowing healthcare providers to access and update patient information easily.
- **Appointment Scheduling:** Created a scheduling system for patients to book appointments with healthcare professionals, with reminders and notifications for upcoming appointments.
- **Patient Portal:** Developed a patient portal where individuals can view their health records, schedule appointments, and securely communicate with healthcare professionals.
- **Event Streaming and Real-Time Data Processing:** Leveraged Kafka to enable real-time data processing and event-driven communication between different modules, ensuring scalability and fault tolerance.
- **In-Memory Data Store:** Implemented Redis for session management and caching to optimize performance and minimize database access latency.
- **IoT and Real-Time Communication:** Integrated MQTT to support real-time communication for IoT devices like wearable health monitors, enabling data synchronization between patient devices and the system.
- **Message Broker:** Used RabbitMQ to handle asynchronous messaging and communication between microservices, ensuring reliability and scalability of the healthcare platform.

Declaration:

I hereby declare all the statements provided are true to my knowledge and I bear responsibility for the correctness of above particulars.

SRINIVASAN K