Sai Kiran N

Sr Full Stack .NET Developer

Email: nagamsaikiran7@gmail.com | Phone: +1 (919) 371 8202

LinkedIn - https://www.linkedin.com/in/saikirannagam

Professional Summary:

- With over 10+ years of experience in various phases of the Software Development Life Cycle (SDLC), I have actively contributed to the analysis, design, development, and maintenance of web applications. My expertise spans across technologies such as **ASP.NET, C#, .NET Core, MVC, JavaScript, Angular and SQL Server**.
- Proficient in Implementing and applying design patterns such as **MVC** (Model - View – Controller), **MVP** (Model - View – Presenter), **MVVM** (Model - View - View Model) and **Singleton**.
- Proficient in implementing applications using **ASP.NET MVC (5/5.2.4), ASP.NET Core MVC, ASP.NET Core Web API**.
- Extensive experience in Developing, Consuming and Debugging the **SOAP** and **REST Web services** using **WCF** and Web API.
- Skilled in utilizing **Entity Framework** to simplify database interaction, accelerating application development through model-driven data access.
- Defining and understanding how to structure endpoints, handle **HTTP** methods **(GET, POST, PUT, DELETE),** and define data formats **(JSON, XML)** is essential.
- Extensive experience in developing robust and scalable applications using **C# and WPF**, delivering rich and responsive user interfaces.
- Vast expertise in developing Web based applications using technologies, **Angular, ReactJs, Cascading Style Sheets (CSS), JavaScript, jQuery, Bootstrap, ASP.NET Master Pages, XML, JSON, and AJAX.**
- Proven ability to optimize Angular applications for performance, including techniques such as lazy loading, **ahead-of-time (AOT) compilation**, and efficient change detection strategies to ensure smooth and responsive user experiences.
- Extensive knowledge on **Azure web application**, App services, **Azure Storage, Azure cosmos DB and Azure DevOps server**.
- Experience with **Azure cloud** and in-depth knowledge on Azure Function Apps, Logic Apps, Application Insights, Azure B2C, Azure Blob application, Azure Data Factory, Azure Key Vault, and containers like **Kubernetes, Docker,** etc.
- Proficient in implementing secure authentication and authorization mechanisms in .Net Applications including **OAuth2, SSL/TLS and JSON Web Tokens (JWT).**
- Hands-on experience in **Angular features** such as Directives, Pipes, Components, Modules, Services, Routing, and Dependency Injection.
- Expertise in writing and optimizing Database design, **SQL Queries** and Stored Procedures, creating functions, views, triggers and indexes, SQL Query Performance Tuning using **MS-SQL Server**.
- Designed and built reliable, highly scalable, agile, and flexible systems using an expressive query language using **MongoDB**.
- Experienced in writing and executing unit testing using **MS Test, NUnit, Xunit**, and expertise in **Test Driven Development.**
- Developed, optimized, and maintained complex **T-SQL** queries, stored procedures, views, and functions in **SQL** Server environments, ensuring efficient data retrieval and manipulation.
- Proficient in managing **multi-model databases (Cosmos DB)** and **relational databases (PostgreSQL)** for seamless data retrieval, partitioning, and archiving, meeting specific industry compliance standards for data storage and access.
- Implemented robust backup and recovery strategies, ensuring data integrity and availability in **SQL Server** and **DB2** environments.
- Experience in creating and **RESTful Web Services** or **APIs** and supporting tools (**Postman, Swagger, JMeter,** hands on experience in developing and deploying Web API, Webservices, providing secure SOA.
- In-depth knowledge in **Visual Studio** and writing **C# code** to migrate **.NET** applications and services from on-premises environment to **Azure**.
- Implemented **AJAX** to load data asynchronously from the server, enhancing the responsiveness and performance of web applications without requiring full page reloads.
- Experienced in designing and implementing solutions using **Cosmos DB** and **PostgreSQL** to support high-performance, globally distributed, and compliant data models for the healthcare and financial sectors.
- Using modern **JavaScript frameworks** and libraries such as **Angular, React, and Vue.js** to develop dynamic single-page applications (SPAs) with efficient state management and component-based architecture.
- Hands-on experience in **Azure Cloud services** such as PaaS, SaaS, and IaaS. Ex: Webapps, WEB Jobs, Functions and Service Fabric

TECHNICAL SKILLSET:

Programming Languages	C#, Python, Typescript, C++, C
Technologies	ASP .NET, Blazor, LINQ, MVC, .NET Core, Web Services, Entity Framework, WPF, WCF, Web API, ADO.NET
Web Technologies	JavaScript, JSON, XML, ReactJS, Angular, NodeJS, HTML, CSS
Services	Web Services, Web API – Restful API, Microservices
Testing	Postman, MS UNIT Test, Mock Test, N-unit Testing
Cloud Technologies	Microsoft Azure, AWS
Design Pattern	MVC, MVVM, CQRS, Repository Pattern, Domain Driven Design Pattern, Singleton, Dependency Injection
Development Tools	Visual Studio .Net, SQL Server Reporting Services, SQL Server Management studio
Version Control Tools	GIT, SVN, TFS
Databases	SQL Server, DynamoDB, PostgreSQL

Certifications:

- Microsoft Certified: Azure Developer Associate
- AWS Certified: Data Engineer Associate
- HackerRank Certified SQL (Basic and Advanced)
- Cisco Networking Academy Certified Python Essentials
- Microsoft Certified Foundational C#

PROFESSIONAL EXPERIENCE:

Fiduciary Exchange (FIDx) Aug 2024 - Current
Remote

Role: Sr. Full Stack .Net Developer

Responsibilities:

- Developed and maintained modern web applications using **Angular, Blazor**, and **.NET 8**, delivering seamless and responsive user experience.
- Built and consumed **RESTful APIs in ASP.NET Core**, ensuring scalable, secure, and performant backend services for client applications.
- Collaborated in **Scrum methodolog**y, participating in sprint planning, daily standups, and retrospectives to deliver incremental value to stakeholders.
- Utilized **Entity Framework Core** for efficient data access, implementing **LINQ queries**, and handling migrations for **MSSQL databases**.
- Integrated **RESTful APIs** with **Angular** applications, implementing secure authentication and efficient data binding, ensuring real-time updates and improved user interactions.
- Integrated **Azure Storage Blob** for scalable and cost-effective storage solutions for static files, such as documents and media.
- Designed and developed multiple **API endpoints** using **.NET Core**, ensuring optimized performance and security, and seamlessly integrated them into the **UI** using **Angular**, enabling efficient data retrieval and user interactions.
- Performed **API testing** and validation using **Postman**, ensuring compliance with business requirements and maintaining API documentation.
- Optimized database schemas and wrote complex **T-SQL queries**, views, and stored procedures to handle data operations and analytics efficiently.
- Conducted code reviews and implemented **TypeScript** best practices in the **Angular codebase** for improved maintainability and reduced technical debt.
- Utilized **Mockaroo** to generate and mock realistic test data during **API development**, ensuring efficient testing and validation of multiple API endpoints before integrating them into the UI.

- Developed comprehensive **API** documentation using **Swagger** for interactive testing and clear API contracts, while leveraging **Postman collections** to document, validate, and automate testing of RESTful endpoints for seamless integration and debugging.
- Developed **Razor Pages** using Blazor to create interactive and dynamic web applications, leveraging Blazor Server and Blazor Web Assembly for enhanced performance and maintainability.
- Leveraged **RxJS** to handle asynchronous data streams in the Angular application, improving real-time interactivity and user experience.
- Utilized **Azure Monitor** and Application Insights for performance monitoring and diagnosing issues in real time, ensuring a reliable production environment.
- Designed and implemented **CI/CD pipelines** using **Azure DevOps**, automating build, test, and deployment processes to ensure rapid and reliable delivery of updates.
- Designed and implemented reusable **Blazor** components to improve development efficiency and maintain a consistent look and feel across the application.
- Enhanced user interaction and application performance by optimizing Angular change detection and employing best practices for state management.
- Delivered interactive data visualizations and dashboards using **Blazor and Angular**, meeting the dynamic reporting needs of stakeholders.

Environment: Angular , .NET 8, Blazor, C#, TypeScript, MSSQL, Entity Framework Core, Postman, Azure Storage Blob, Azure DevOps, CI/CD Pipelines, Scrum/Agile Methodology, Visual Studio 2022, Git.

Blue Cross and Blue Sheild (BCBS) May 2023 – Aug 2024
North Carolina

Role: Sr. Full Stack .Net Developer

Responsibilities:

- Employed **Agile methodology** to ensure iterative and flexible development, adapting to evolving requirements seamlessly.
- Leveraged **ASP.NET Core Web API** to develop RESTful web services for seamless integration with the front-end client.
- Used **Azure Blob Storage** for storing static assets, such as images, documents, and media files, serving them to the front-end Angular application and optimizing content delivery and scalability.
- Developed **RESTful APIs** to facilitate communication between the backend services and the **Angular-based front end**.
- Hosted the frontend Angular application and backend **ASP.NET Core Web API** on Azure App Service, providing a scalable and managed platform for web application hosting.
- Upgraded BCBS healthcare analytics platform to **.NET 8**, leveraging performance enhancements and minimal APIs to streamline backend API design, improving request processing times for medical data queries.
- Utilized **Entity Framework Core** for efficient object-relational mapping and database access, streamlining data operations.
- Adopted **JSON** as the data interchange format for efficient communication between the client and server components.
- Orchestrated the deployment of the application on **Azure Cloud**, leveraging API Gateway services for efficient management and security of API endpoints.
- Implemented **OAuth2** for robust authentication and authorization mechanisms, ensuring secure access to banking functionalities.
- Developed the user interface using **Angular** to deliver a responsive and dynamic application experience.
- Employed **HTML, CSS, JavaScript, and Bootstrap** for frontend styling and user interaction enhancements.
- Developed **PostgreSQL** views and complex joins to facilitate real-time analytics on patient data, supporting BCBS's healthcare analytics platform with insights into treatment outcomes and patient engagement.
- Integrated Azure Monitor for monitoring the performance, availability, and health of the application deployed on Azure Cloud, allowing proactive detection and resolution of issues through metrics, logs, and alerts.
- **Cosmos DB** offers multi-region writes and reads, ensuring high availability and low latency for a healthcare analytics platform that may require global data accessibility.
- Developed robust backend services using **C# and ASP.NET Core Web API**, ensuring seamless integration with the Angular front-end for various healthcare applications.
- Designed data partitioning strategies within **Cosmos DB** to manage healthcare data effectively, optimizing query performance and cost efficiency in high-volume, multi-region environments.
- Developed custom **C# scripts for Raspberry Pi** to interface with medical equipment, ensuring seamless data collection and analysis.

- Utilized **WPF** to create dynamic and interactive user interfaces for internal tools, enhancing the productivity of the healthcare management team.
- Implemented complex data processing and analytics using **C# and SQL Server**, improving data-driven decision-making for healthcare services.
- Selected **SQL Server** for the **relational database management system**, facilitating efficient storage and retrieval of transactional data.
- Written Queries using **T-SQL** Statements, Views, User Defined Functions, Stored Procedures, Triggers in **SQL** Server 2019 for inserting/Updating/Deleting the data into the relational tables.
- Integrated **RabbitMQ** as the message queue for handling asynchronous communication and processing of banking transactions.
- Utilized Postman for API testing and **Swagger UI** for comprehensive API documentation.
- Leveraged **Azure Cloud** for hosting the application, managing infrastructure resources, and ensuring scalability and reliability.
- Implemented **PostgreSQL** data replication and backup strategies to maintain data integrity and support disaster recovery needs, ensuring secure and reliable access to sensitive healthcare data.
- Led successful database migration projects, transitioning from legacy systems to modern **SQL Server** architectures with minimal downtime and data loss.
- Integrated **Azure Application** Insights for application performance monitoring and diagnostics, tracking user interactions, exceptions, and dependencies to gain insights into application behavior and improve user experience.
- Employed **Visual Studio** and **Visual Studio Code** as the primary IDEs for coding, and **SQL Server Management Studio** for database management tasks.
- Utilized **Git** for version control, enabling collaborative development and code management.

Environment: Visual Studio, .Net Framework, C#, .Net Core, JavaScript, Azure, MVC, ASP.NET, Web API, Rest API, Microservices, AJAX, HTML, CSS, WCF, WPF, ADO.NET, Bootstrap, YAML, SOAPUI, SSRS, IIS, T-SQL, SQL Server, MongoDB, PostgreSQL, Cosmos DB, Nlog, TFS.

First Citizens Bank Jun 2021 – Apr 2023
Raleigh, NC

Role: Full Stack .Net Developer

Responsibilities:

- **Agile methodology** was adopted to facilitate iterative development, continuous feedback, and adaptability to changing requirements.
- Employed **MVC** architecture for the development of the banking platform, utilizing the **.NET Framework**.
- Implemented **SOAP** web services for communication between different components of the banking platform, ensuring interoperability and reliability.
- Utilized **ADO.NET** for database access, allowing for efficient data retrieval and manipulation.
- Implemented Cross-Origin Resource Sharing (CORS) policies in the **ASP.NET Core Web API** to restrict access to resources from unauthorized domains, enhancing security against cross-site scripting (XSS) attacks.
- Implemented **Cosmos DB** with advanced indexing and partitioning to support high-volume financial transactions, ensuring quick data access and secure storage for banking operations.
- **XML** was chosen as the data format for exchanging structured data between different modules of the banking platform.
- Implemented **API gateways** as part of the microservices architecture to expose **RESTful APIs** to external clients, providing a unified interface for accessing multiple microservices, and enforcing security, rate limiting, and authentication policies at the edge of the system.
- Employed **C# and .NET Framework** to develop **MVC architecture** for the banking platform, ensuring a clean separation of concerns and scalability.
- Designed a data archival strategy using **PostgreSQL** to maintain compliance with regulatory data retention policies while ensuring efficient storage management for historical banking records.
- Designed and developed **WPF** applications for internal banking tools, improving user interaction and operational efficiency.
- Utilized **Angular framework** for building dynamic and responsive frontend components, leveraging features such as components, services, and modules for modular and maintainable codebase.
- **HTML, CSS, JSP, JavaScript, jQuery, and Ajax** were used for developing the frontend, providing a responsive and interactive user interface.

- Configured multi-region write and read capabilities in **Cosmos DB** to support transaction reliability and failover across banking locations, enhancing the resilience of the financial platform.
- Integrated **Azure AD** for identity and access management in the banking platform, enabling single sign-on (SSO), user authentication, and role-based access control (RBAC) for securing access to resources and **APIs**.
- Developed **RESTful APIs** using **ASP.NET Core Web API** to expose backend functionalities to client applications, ensuring a scalable and decoupled architecture.
- Leveraged **Angular** HTTP Client module for making **HTTP** requests to **RESTful APIs**, handling asynchronous operations, and managing application state effectively.
- Implemented **JWT** (JSON Web Token) authentication in **RESTful services** to secure access to protected resources and authenticate users in a stateless manner.
- Optimized **PostgreSQL** schemas and stored procedures to process large transaction datasets efficiently, enhancing the speed and reliability of customer account management and reporting.
- Employed **SQL Server** as the relational database management system for storing product information, user data, and order details.
- Flexible Data Model in **Cosmos DB:** Supports multiple data models (key-value, document, graph, column-family) allowing flexibility in how banking data is stored and accessed.
- Utilized **XUnit** for testing **.NET Framework** components, ensuring code quality and reliability.
- Defined and deployed infrastructure as code using **Azure Resource Manager (ARM)** Templates, enabling automated provisioning, management, and versioning of **Azure resources** for the banking platform, ensuring consistency and repeatability across environments. Utilized **XUnit** for testing **.NET Framework components**, ensuring code quality and reliability.
- **SVN** was used for version control, allowing for collaborative development and code management.
- **Visual Studio** was used as the primary integrated development environment (IDE) for coding, debugging, and building the banking platform.
- Integrated **Log4Net** for logging and monitoring application events and errors, facilitating troubleshooting and debugging processes.
- Jira was chosen as the ticket/project management tool for managing project tasks, tracking work items, and facilitating team collaboration.
- Hands-on experience in **Azure Cloud Services PaaS**, Web Apps, Blob Storage Application Insights, and Logic App.

Environment: C#, .NET Core 3, Visual studio 2019, HTML, JavaScript, Entity framework, Entity framework Core, SQL Server, Angular 9, Typescript, Bootstrap, Restful Services, SOAP Services, jQuery, Agile, WEB API Core 3.0, XML, UML, IIS 8.0, Azure, TFS, GIT.

Inspire Brands Aug 2020 – Apr 2021
New Jersey, United Sates

Role: .Net Developer

Responsibilities:

- Implemented the **Waterfall methodology** for sequential project phases, including requirements gathering, design, implementation, testing, deployment, and maintenance.
- Utilized **.NET Web Forms** for building the backend of the website, providing a familiar and event-driven model for developing dynamic web applications.
- Implemented **SOAP web services** for communication between the website and external systems, ensuring interoperability and standardized messaging.
- Leveraged **ADO.NET** for database access, allowing for efficient data retrieval, manipulation, and storage.
- Utilized **XML** as the data format for exchanging structured data between different components of the website.
- **HTML** and **CSS** were used for structuring and styling the website's layout and design.
- Web Forms provided server-side controls and events for building interactive and dynamic web pages.
- **JavaScript, jQuery, and Ajax** were employed for client-side scripting and asynchronous communication with the server, enhancing the user experience.
- Developed backend services using **C# and .NET Web Forms**, ensuring efficient data processing and business logic implementation for the website.
- Implemented **WPF** applications to streamline internal processes and enhance user experience for the brand management team.

- Utilized **Raspberry Pi** devices for monitoring and controlling various brand assets, ensuring efficient and automated operations.
- Employed **PostgreSQL** as the **SQL database management system** for storing product information, user data, and order details, ensuring data integrity and reliability.
- Implemented and optimized **MongoDB** queries and indexes to improve data retrieval speed, reducing latency and enhancing the overall performance of the application.
- Successfully migrated and integrated legacy **SQL databases into MongoDB**, ensuring data integrity and enhancing the scalability and performance of the backend infrastructure.
- Implemented **XUnit** for **unit testing .NET Framework** components, ensuring code quality, reliability, and maintainability.
- Utilized **SVN** (Subversion) for version control, enabling collaborative development, code management, and version tracking.
- Used **JIRA** as the ticket/project management tool for tracking project tasks, managing work items, and facilitating team collaboration.

Environment: MS Visual Studio 2012/2013, .NET Framework 4.5, C#, MVC 5.0, Agile development, Entity Framework, Web API, SQL Server 2008, Web Services, SSRS, HTML, CSS, XML, JQuery, AngularJS, Angular 2, Bootstrap, TDD.

LVB Jun 2014 – Oct 2019
Hyderabad, India

Role: .Net Developer

Project 1: Online banking Platform

Responsibilities:

- Conducted detailed requirements analysis to understand the business needs and translate them into technical specifications.
- Designed the system architecture using **MVC** (Model-View-Controller) to ensure a clean separation of concerns and scalability.
- Developed backend services using **ASP.NET Core Web API** to handle business logic and data processing.
- Utilized **Entity Framework Core** for **ORM** (Object-Relational Mapping) to facilitate efficient database operations with **SQL Server**.
- Utilized **C# and ASP.NET Core** to develop backend services, handling business logic and data processing for the online banking platform.
- Implemented **RESTful APIs** to expose banking services to front-end applications and third-party clients.
- Created dynamic and responsive user interfaces using **Angular** 10 to enhance user experience.
- Employed **Angular** features such as components, services, and dependency injection for a modular and maintainable codebase.
- Developed dynamic and responsive user interfaces using **WPF**, improving the banking experience for end-users.
- Employed **Raspberry Pi** devices for secure and efficient IoT-based banking transactions and monitoring.
- Utilized **HTML, CSS, and Bootstrap** for styling and responsive design.
- Implemented **OAuth2** and **JWT** (JSON Web Token) for secure authentication and authorization.
- Deployed the application on **Azure App Service** for scalable and managed hosting.
- Utilized **Azure Blob Storage** for storing and serving static assets such as documents and images.
- Implemented data migrations using **Entity Framework Core** to handle schema changes.
- Utilized **Azure Monitor** and **Azure Application** Insights to track application performance, availability, and diagnose issues proactively.

Environment: C#, ASP.NET Core, Entity Framework Core, Angular 10, Azure App Service, Azure Blob Storage, SQL Server, HTML, CSS, Bootstrap, OAuth2, JWT, Postman, Swagger, Git, Log4Net, Azure Application Insights.

Project 2: Mobile Banking Application

Responsibilities:

- Collaborated with stakeholders to gather requirements and define the project scope and objectives.
- Created detailed technical specifications and project plans to ensure clarity and alignment with business goals.
- Implemented secure **RESTful APIs** to support mobile client interactions.

- Developed the mobile **frontend** using **Angular** and **Ionic framework** to ensure a consistent and responsive user experience across different devices.
- Leveraged **Angular** features such as Reactive Forms and **HTTP** Client for efficient form handling and **API** communication.
- Distributed Database with **Cosmos DB:** Ensures data is replicated across multiple regions, providing high availability and disaster recovery capabilities for mobile banking applications.
- Utilized **Entity Framework** for database operations and managed data access through repository patterns.
- Implemented complex queries and stored procedures in **SQL Server** for efficient data retrieval and manipulation.
- Deployed backend services on **Azure App Service**, ensuring scalability and reliability.
- Used **Azure Key Vault** for managing sensitive information such as connection strings and **API keys.**
- Integrated **Azure** Application Insights for monitoring application performance and user interactions.
- Utilized **Log4Net** for logging critical events and errors, facilitating quick identification and resolution of issues.

Environment: C#, ASP.NET Core, Entity Framework, Angular, Azure App Service, Azure Key Vault, SQL Server, Azure Active Directory B2C, Git, Log4Net, Azure Application Insights,