Vishnu Pasuleti

Vishnupasuleti1234@gmail.com, M: 7893184451,
https://vishnupasuleti.com/cv

Summary:

- Full Stack Software Engineer with over 10 years of experience in architecture design, product engineering, and application development.
- Demonstrated expertise in implementing a microservice design approach and managing secure, scalable n-Tier Web, Desktop, Multi-Threaded, Distributed, and Business Process Workflow-based systems.
- Proficient in building large-scale applications using Azure cloud and on-premise web server farms with Agile and Application Life-cycle Management with Git.
- Experienced in developing intelligent APIs across LOB applications and maintaining documentation with tools like Swashbuckle, AutoRest and Stoplight.
- Broad experience working with various architecture patterns including microservices, monolithic, and JAM Stack.
- Comprehensive knowledge and hands-on experience with a wide range of frontend libraries such as Angular JS, Angular, React, Next JS.
- Proven ability to adapt and learn new technologies, evidenced by a diverse skill set including Programming Languages (C#, JavaScript, Typescript), Backend Web Frameworks (ASP.NET core, ASP.NET MVC, Express JS), Cloud Technologies/Services (Azure), and Database Management (SQL Server, MongoDB, Cosmos DB)
- Proficient in multiple DevOps practices with experience in TeamCity, Octopus, GitLab, and Azure DevOps.
- Familiar with various Software Development methodologies, including Safe Agile, BDD, and TDD.

WORK EXPERIENCE

Full Stack developer

X-Team | Oct 2021 – Present

I am working on a project for a Compass Education client in Australia. The project involves developing a school management web application. My tasks include working on people management and enrolment application modules.

- Worked on backend, developing both GraphQL and REST APIs using C#. Implemented business logic and integrated data from MongoDB and SQL Server databases.
- Frontend development included designing UI components with React and Styled Components, along with maintaining legacy sections of applications using ExtJS.
- Significantly contributed to the creation of custom forms in React JS, similar to Google Forms. Heavily involved developing validation library for cohesive functionality between frontend and backend custom forms.
- Took part in meetings to brainstorm how we could make our product better.

SKILLS

Programming Language
C#, JavaScript, Typescript

Backend web frameworks
ASP.NET Core, ASP.NET MVC
WCF, Express JS

Database
SQL server, Mongo DB,
Cosmos DB

architecture patterns
Monolith, Microservice, JAMSTACK
micro front ends, Serverless
Event driven architecture

Frontend frameworks
React, Angular, 11ty, jQuery,
RXJS, MUI react, Angular material
Prime NG, HTML, CSS,
Nunjucks, Razor

Azure
Azure AD, Storage, Functions,
Event Grid, Event Hubs, App services.
Front door, APIM, Service bus,
Logic Apps, Durable functions
Key vault, App insights

CSS frameworks
Tailwind, Bootstrap,
Styled components

Mobile apps
React Native apps
Xamarin forms

Devops
Teamcity, octopus, gitlab
azure Devops, Docker

API design and development
OPEN API, Stoplight, Graphql

Authentication
Oauth, Open id connect,
Forms authentication,Identity server

Version control
Git, TFS

ORM
EF Core, ado.net, prisma

Testing Frameworks
Playwright, Selenium, NUnit

Software development
Safe Agile, BDD, TDD

Principal Software Engineer

Calyx(Formerly Parexel) | Jan 2021 – Oct 2021

Worked on an integration hub project that consolidated multiple integrations across various products into a single platform, thereby achieving significant standardization. Developed this solution using a micro frontend architecture and microservices.

- Acted as the team lead for a group of five developers. Responsibilities included sprint planning, conducting code reviews, and mentoring junior developers.
- Designed and implemented low-level architecture for several integrations using C#, .NET Core, and Azure Functions. Additionally, contributed to high-level design and architecture discussions.
- Developed UI components using Next JS and Typescript.
- Implemented CI/CD pipeline using teamcity and octopus deploy.
- Initiated and implemented a micro frontend architecture using React, which allowed for independent deployable components. Presented this initiative to technical leadership.

Senior Software Engineer

Parexel | Mar 2018– Jan 2021

Worked on diverse range of projects, including a Clinical Trial Management System, a Patient Engagement Platform, COVID survey systems and an Integration Hub. Developed these solutions using microservices architecture, JAM Stack, and monolithic architecture respectively.

- Served as a backend developer, where I developed REST APIs using C# and .NET Core, and capitalized on Azure cloud services. Gained experience working with SQL Server and Cosmos DB databases.
- Actively worked on enhancing application performance by implementing caching strategies, refactoring code, and optimizing application architecture.
- Created UI components utilizing Angular and TypeScript, reinforcing frontend development skills.

Software Lead Engineer

Technovert | Aug 2014 – Feb 2018

Worked as contractor for Microsoft for migrating their sales windows app to web app and involved in other projects. Worked on HRMS product and contributed to various module like attendance, leaves etc

- Worked as a full stack developer, where I developed REST APIs using C# and ASP.NET MVC, and capitalized on Azure cloud services. Gained experience working with SQL Server and Cosmos DB databases.
- Worked on frontend using Angular JS and Angular 2+ for developing UI.
- Worked on mobile app development using Xamarin.
- Lead a team of 3-4 developers and responsible for sprint planning, code reviews and client interaction.

Associate Technology L1

Sapient global markets (Now Publicis Sapient) | Jul 2012 – Jul 2014

Worked as a developer for various projects.

- Worked on desktop application development using WPF and C# and also worked on web application development using ASP.NET MVC and C#.

CERTIFICATION

AZ-900: Microsoft Azure Fundamentals

EDUCATION

Bachelor of Technology in Electronic and Communication Engineering (ECE)

VNIT | 2008 - 2012

KEY PROJECTS

Compass Education:

This platform operates as a Software as a Service (SaaS) solution designed for comprehensive school management. It serves a wide range of users, including staff, students, and parents, with several key modules for people management, enrolment, timetable scheduling, and attendance tracking. The platform centralizes essential administrative tasks, thus improving communication and overall efficiency within the school environment.

Tech stack:

ASP.NET Core, ASP.NET, C#, JavaScript, TypeScript, React, ExtJS, Material-UI, Webpack Module Federation
Mono Repo, Microservice, Microfrontend, MongoDB, SQL Server, Entity Framework Core, Elastic Stack
Hot Chocolate GraphQL, Apollo GraphQL Client, Redis, Couchbase, Styled Components

Technical details:

The system is structured around an ASP.NET monolithic architecture where most core module APIs are housed and exposed as RESTful (Open API) endpoints. This same architecture also delivers frontend HTML. Authentication is accomplished through SAML or forms authentication, contingent on school login protocols.

Data management is conducted through MongoDB and SQL Server databases. In addition, there are smaller, specialized microservices such as a search API that utilizes Elasticsearch. These microservices communicate with the monolithic backend via HTTP REST APIs.

On the frontend, the legacy UI is built with ExtJS while the new UI incorporates React JS in a micro frontend architecture. Here, individual module static files are sourced from an S3 bucket, which are produced from a frontend React mono repo using webpack module federation.

Role & contributions:

As a full-stack developer on this project, my contributions spanned both frontend and backend development.

- Developed a dynamic form generation system using React. This allowed for the creation of a variety of question and answers forms akin to Google Forms. These forms could be created by administrators and then presented on the enrolment form for parents to complete.
- Crafted a validation library compatible with both frontend and backend. The validation rules were stored in MongoDB, and converted into a "yup" schema for frontend use. In the backend, we created multiple validators using defined validators and expressions, ensuring compatibility with the FluentValidation library of C#.
- Developed a rate limiter to curb account creation, thus improving system security.
- Engineered a recommendation system for schools. This engine could take multiple parameters into account to suggest suitable schools to users.
- Uplifted multiple screens from the old ExtJS system to React Material-UI components, thereby modernizing the user interface.
- Created multiple UI components that enabled ExtJS and React to work in tandem. For instance, I was able to display an ExtJS window in a React context and vice versa.

Oneworld:

This logistic app serves as a comprehensive solution for managing multiple shipments using both air and road transportation modes. It offers users the ability to track their shipments in real-time, providing transparency and visibility throughout the logistics process. The app streamlines various logistics operations, including shipment scheduling, route optimization, documentation management, and delivery tracking. By utilizing this app, businesses can enhance their supply chain management, improve efficiency, and provide an enhanced customer experience.

Tech stack:

React Native, Angular, Bootstrap, RabbitMQ, C#, ASP.NET Core, SQL Server, TypeScript, JavaScript
Seq (for logging), Microservices, Identity Server

Technical details:

The frontend application is developed in Angular 8, with two modules catering to different user needs: one for customer-facing functionality and another for backend office operations, with support for multiple roles. The application is hosted using the ASP.NET Core framework, where the static file middleware is exposed. The Angular app is loaded from this middleware, allowing seamless access to the user interface. Authentication and authorization are handled through Identity Server, ensuring secure access to the application.

The backend of the application exposes APIs using ASP.NET Core 5. The data persistence layer is managed using EF Core Code First, serving as the object-relational mapping (ORM) for SQL Server. The application architecture follows the CQRS (Command Query Responsibility Segregation) pattern, utilizing the MediatR library for command and query handling. Additionally, a background service is implemented as a Windows service using TopShelf. This service performs various tasks such as payment status checks, email sending, and invoice processing. Communication between components is facilitated using RabbitMQ messaging. The backend also includes two microservices: the Print Service for generating printable HTML using Razor templates and the Tracking Service for retrieving historical track information of airway bills from previous transactions.

The mobile application is developed specifically for customers, allowing them to track their shipments and make online payments conveniently from their mobile devices.

The entire application is deployed on a Windows Server machine, created through Azure Virtual Machines. This deployment approach provides flexibility and scalability while leveraging the capabilities of the Azure cloud platform.

Role & contributions:

As the architect and developer for the entire project, I undertook various challenging tasks, including:

- Creating the mobile app using React Native and implementing notification handling to provide timely updates to users.
- Developing the backend system using the API-first approach, ensuring robust and efficient communication between the frontend and backend components.
- Integrating a payment gateway to facilitate secure and seamless payment transactions within the application.
- Implementing the functionality to generate and allow users to download invoice documents, streamlining the billing process.
- Developing multiple Angular components to enhance the user interface and provide a smooth user experience.
- Implementing patterns like infinite scroll to efficiently handle large datasets and optimize data loading and rendering.

Integration Hub:

The platform functions as a Software as a Service (SaaS) solution, providing an integrated suite of services. It offers a comprehensive set of features including Job Monitoring, Bulk Uploads, Audit Logs, and Event Subscriptions. These capabilities make the platform a versatile tool for managing a variety of tasks, streamlining operations, and enhancing visibility into system events, all in one place.

Tech stack:

Azure Functions, Azure AD, Azure Event Grid Topics, Azure Event Hubs, Azure Event Domains,Azure Service Bus, Azure Storage, Cosmos DB, Azure SQL Server, ASP.NET Core, Azure Front Door,Azure API Management, Azure App Services, C#, Next.js, React, Bootstrap, TypeScript,Scaled Agile Framework (SAFe), Clean Architecture, Entity Framework Core, API Design First,Stoplight, Microservices and Microfrontends, JAMstack, Azure Key Vault, TeamCity, Octopus,Azure Active Directory, Azure Identity Management

Technical details:

The system is designed as a series of microservices, each corresponding to a different integration. These microservices communicate asynchronously using Azure Event Grid topics. The user interface for all integrations is developed using Next.js.

APIs have been written in a combination of Azure Functions and ASP.NET Core, showcasing a diversified approach to backend development. These are all deployed on the Azure cloud platform across multiple regions, with Azure Front Door serving as the entry point.

Code organization follows the Clean Architecture pattern, promoting separation of concerns and making the system easier to understand and maintain. Builds are managed via TeamCity, and deployments to Azure are handled through Octopus, streamlining the continuous integration and deployment pipeline.

Role & contributions:

As a Principal Engineer on this project, my responsibilities were wide-ranging and integral to the project's success.

- Owning the low-level design and contributing significantly to the high-level design of the system. I developed low-level designs for several integrations, such as notifications, event subscriptions, and audit design.

- Setting up the foundational code components and corresponding services on Azure, thereby initializing the system infrastructure.
- Conducting research and contributing to various proofs of concept (POCs), aiding in decision-making for high-level design.
- Implementing a range of features using Next.js and React for UI development, and ASP.NET Core, Cosmos DB, and SQL Server for backend development.
- Assuming a leadership role within the team, providing guidance, and mentoring junior members to foster their professional growth.
- Intensively working on Azure Event Grid subscriptions and Azure Functions, demonstrating expertise in these areas.

Covid RWE APPS:

This is a web-based application designed to facilitate clinical trials. The application provides survey questions to subjects and physicians. Their responses are then collected and analyzed to generate valuable insights for clinical trials. This data can be used to assess treatment effectiveness, monitor side effects, and evaluate patient experiences, among other uses. The application thus serves as a crucial tool in conducting and optimizing clinical trials.

Tech stack:

C#, Azure Functions, Azure Durable Functions, Azure Service Bus, Azure B2C, Azure Storage, Azure CDN, 11ty Nunjucks, jQuery, Bootstrap, JAMstack, Azure Automation

Technical details:

This project employs the JAMstack architecture, with the user interface developed using 11ty, Nunjucks, and jQuery templates. During the build process, these tools generate static HTML and JavaScript files, which are then uploaded to Azure Storage and presented as a static website.

On top of this, Azure CDN profiles are created to enhance content delivery. The APIs are developed using Azure Functions, with all survey data stored in Azure Blob Storage. An event subscription to an Azure Function triggers data push into Azure Data Lake for storage and analysis.

Data analytics are conducted using Power BI, which operates on the data gathered in the Azure Data Lake, transforming the raw survey data into actionable insights for clinical trials. The overall architecture of the system ensures a fast, reliable, and scalable solution for conducting and analyzing clinical trials.

Role & contributions:

As a Senior Full Stack Software Engineer on this project, my contributions were instrumental in various areas:

- I wrote code to implement multiple features, actively contributing to the development of the overall system.
- I had a significant role in the notification projects where Azure Durable Functions were used. This involved creating complex, stateful patterns in the serverless environment, providing reliable and efficient notification services.
- I implemented localization for different languages. This involved writing all context in YAML files. During the build process, we generated multiple static files for different languages, enabling the system to cater to a global audience by providing a user-friendly, localized experience.
- I implemented several design patterns including Orchestration and Choreography.

E-Clinical platform:

This is a Software as a Service (SaaS) production environment that encompasses multiple modules, designed to manage all operations for an e-clinical platform. It provides a comprehensive toolset to streamline and handle various aspects of clinical operations, from patient management to data analysis, making it an integral part of the healthcare technology ecosystem.

Tech stack:

Azure App Service, Azure API Management, Redis, Azure AD, Azure Functions, Azure SQL Server, Cosmos DB, Azure Storage, Azure Virtual Machines, ASP.NET Core, Angular 4, TypeScript, JavaScript, Azure Application,Insights, Scaled Agile Framework (SAFe), Behavior-Driven Development (BDD), SpecFlow

Technical details:

Each module within the system is a separate microservice, partitioned using Bounded Context. The microservices communicate with each other asynchronously using Azure Service Bus, with some microservices interacting via

REST APIs. Azure services are utilized to host these microservices with autoscaling capabilities, ensuring scalability and resilience.

For data storage, Azure Cosmos DB and SQL Server are used as backend databases. Azure Logic Apps facilitate various workflows, streamlining the processes within the system. Worker processes are deployed for different cron jobs and notification tasks, ensuring timely execution of scheduled operations.

The user interface is developed using Angular 4+, providing a dynamic and user-friendly frontend. The application is deployed on Azure App Service, benefiting from its scalable and secure hosting environment.

Role & contributions:

As a Full Stack Developer, contributions spanned various aspects of the product's lifecycle, from feature development to stabilization. Key responsibilities and accomplishments included:

- Worked on both frontend and backend development, utilizing Angular 4 for frontend development and ASP.NET Core for backend logic.
- Leveraged Azure services for application development and hosting, harnessing its scalability and robustness.
- Enhanced application performance through various techniques, such as caching and minimizing API calls via data redundancy using the Bounded Context concept.
- Implemented resilience strategies such as retry and circuit breaker patterns, contributing to improved system reliability and fault tolerance.
- Contributed to test automation by utilizing Selenium and SpecFlow, ensuring robust and reliable product delivery.

Amendment App:

This is a sales application that streamlines the contract creation process by consolidating multiple document amendments to generate a final, unified contract document. By effectively organizing and managing amendments, the application simplifies the contract generation process, saving time and reducing the potential for errors

Tech stack:

C#, ASP.NET MVC, Angular 2, Universal Windows Platform (UWP) App, XAML, TypeScript, JavaScript, Azure Service Bus

Technical details:

The backend of this system utilizes ASP.NET MVC to serve APIs, deployed to Azure App Service for secure and scalable hosting. For asynchronous operations, Windows services are utilized. The frontend comprises a Windows application, which was later converted into a web application implemented using Angular 2+. This provides a unified, responsive user interface across multiple platforms.

Role & contributions:

As a Lead Engineer on this project, my responsibilities were broad and multi-faceted. They involved liaising directly with Microsoft product managers, implementing key features, and guiding a team of developers. Key contributions include:

- Regularly meeting with product managers from Microsoft to gather requirements and update progress. This ensured the alignment of development activities with business objectives and stakeholder expectations.
- Creating a migration plan for transitioning from a Windows application to a web application, thereby enabling contract creation from mobile or desktop devices. This significantly expanded the accessibility and user base of the application.
- Leading the implementation of numerous features, utilizing Angular JS for frontend development and C#, ASP.NET MVC, SQL Server, and Azure Storage for backend tasks. This combination of technologies allowed us to build robust, scalable, and user-friendly features.
- Acting as a team lead, conducting code reviews, and distributing tasks among team members. This ensured high-quality code, efficient work distribution, and the continual development and learning of the team.

KEKA HR Platform:

This is a Software as a Service (SaaS) platform designed for Human Resource Management System (HRMS). It includes numerous modules like attendance tracking, leave management, policy administration, and more, providing a comprehensive solution for efficient and effective human resource management.

Tech stack:

C#, ASP.NET MVC, AngularJS, Bootstrap, Azure App Service, Azure SQL Server, Redis, Azure Storage

Technical details:

The platform is built on ASP.NET MVC, deployed to Azure App Service for reliable hosting and scalability. Azure Database is used for managing data, ensuring high performance and security. The frontend is developed with AngularJS and Bootstrap, providing a responsive and user-friendly interface.

In addition to the web application, a mobile application is developed using Xamarin Forms, allowing users to access HRMS services on the go. Both the web and mobile applications employ the Backend for Frontend (BFF) pattern, ensuring each client gets what it needs from the backend services in an efficient manner.

The mobile application is built with an 'offline-first' approach. This is achieved by maintaining a local SQLite database on the user's device which synchronizes with the server when a reliable network connection is available. This ensures uninterrupted access to essential HRMS services, even in areas of poor connectivity.

Role & contributions:

As a Full Stack Developer, my role involved contributing to multiple modules and projects. Key responsibilities included:

- Development of a Xamarin Forms mobile app, providing users with access to essential features and functionalities on their mobile devices.
- Utilizing WPF (Windows Presentation Foundation) to develop a client-facing app. This app allowed users to upload attendance logs and synchronize data from their attendance devices.
- Collaborating with the team to deliver high-quality software solutions, ensuring efficient and reliable performance across different platforms.
- Participating in the full software development lifecycle, including requirements gathering, design, implementation, testing, and deployment.
- Working with various technologies such as C#, ASP.NET MVC, Xamarin Forms, WPF, and integrating them seamlessly to deliver comprehensive solutions.
- Conducting code reviews, providing feedback, and assisting team members to ensure code quality and adherence to best practices.
- Continuously exploring new technologies and staying up-to-date with industry trends to bring innovation and improvements to the development process.

Global analytics and Risk management:

Enterprise data management system which stores critical business information from trading positions, P & L and risk information.

Tech stack:

WPF, ASP.NET MVC, Telerik Controls, WCF, SQL Server

Technical details:

The web application is developed using ASP.NET MVC with Telerik controls and is hosted on an on-premises server. In addition to the web application, there is also a WPF (Windows Presentation Foundation) application that utilizes WCF (Windows Communication Foundation) to serve APIs.

Role & contributions:

As a full stack developer, my role encompassed various responsibilities in both frontend and backend development. Here is a high-level description of my contributions:

- Involved in the complete software development lifecycle, from requirements gathering to deployment.
- Developed user interfaces using technologies such as HTML, CSS, and JavaScript.
- Implemented business logic and data processing on the backend using languages like C# and frameworks such as ASP.NET.
- Worked with databases to design and implement data models and perform database operations.
- Collaborated with the team to ensure seamless integration and communication between the frontend and backend components.
- Utilized frameworks and libraries, such as Angular, React, or Vue.js, to enhance the frontend development process.
- Conducted testing and debugging to ensure the functionality and quality of the application.
- Employed version control systems and followed best practices for code management and collaboration.
- Kept up with industry trends and emerging technologies to continuously improve development skills and stay current with the latest tools and techniques.