Mohit Singh Sengar

Senior Full Stack Developer | Solution Architect

in/mohitsengarr/ mohitsengarr

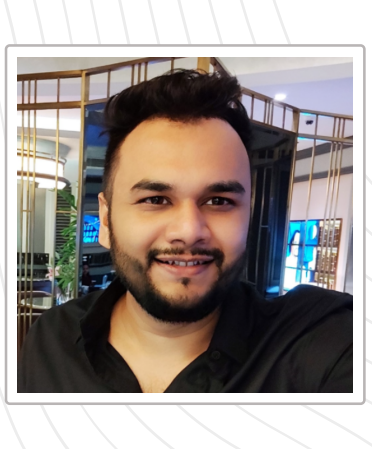
mohitsengarr

mohitsingh.shiv@gmail.com

+971504059627

Currently in Dubai

India



SUMMARY

Full stack expert having more than 10 years of experience, I took part in making great applications built on top of the .NET platform, Javascript platforms, and Blockchain-related technologies.

Programming Languages and Work Domain:
Node.js, .Net core (C#), Blockchain, NFT, Asp.net Web API/MVC, Blazor, Vue.Js, ReactJs, Angular, MEAN stack, MERN stack, MEVN stack, Next.js, Nuxt.js, Nest.js, Azure, AWS, Twilio, WPF, WCF, XAML, Entity Framework, Sequelize, Autofaq, Linq, Socket Programming, Socket.IO, WebRTC, Agora.io, Azure Functions, Python, Azure App Services, Azure API Management or Gateway, Azure Web Jobs, Azure Notification Hub, Azure Application Insights, Azure Cognitive Services, Serverless.com, AWS, Lambda Functions, AWS CloudFront, Firebase, Bubble.io, Flutter, AWS App Sync, GraphQL, Hasura, GCP, Heroku, PowerBI. Sitecore, Ninja Trader.

Blockchain Technologies:
Ethereum, NFT, Near blockchain, Remix IDE, Web3, RPC, Solidity, RUST, EOS, ADA, Defi Apps, Decentralized lending protocol, Decentralized Applications, Hyperledger, Fabric, Iroha, Sawtooth, Cello, Uniswap V1, V2, V3, V3 optimism, DApps, Tezos, CosmWasm, Cosmos SDK, Metaverse.

Identity Management
Azure AD B2B, B2C, Auth0, Okta, AWS Cognito

Bot Framework:
Microsoft bot framework, Api.ai, Luis.ai, Motion.ai, Botkit, Chatbot, ChatGPT

Programming Methodologies and Patterns :
SOLID, Microservices, MVC, MVVM, Singleton design pattern, Factory pattern, Command Pattern, ObserverPattern,
Object-Oriented Concepts(OOPs), Microservices, DDD(Domain Driven Design), TDD(Test Driven Design)

Web Technologies and Protocols :
ASP.NET,SOAP,REST,XMPP,JSON,OAUTH,Basic AUTH,HTML,XSLT,XML,FTP, HTTP,SMTP,VBScript,RESTful Webservices, Socket.

Database Management Systems :
MySql,Sql Server, Aws DynamoDB, MongoDB, RedisDB, Azure CosmosDB, Cassandra, PostgreSql, Neo4j.

Web Service Testing Consoles :
Postman, REST Console, Swagger.

Server :
Docker, Kubernetes, IIS 6.0,7.0,8.0, Nginx, Express, Azure AKS, AWS containers

Project Management Tools :
Trello, Redmine, Atlassian JIRA, ClickUp, Clubhouse, Kiss flow

Framework :
.Net Framework 2.0,4.0,4.5. 4.6, .Net Standard, .Net Core 2.0, 2.1, 3.0, 5

Version Control Systems :
GIT,KDiff ,Bitbucket,Gitlab,Tortoise SVN,Github.

Continuous Integration and Deployment :
CircleCI, Jenkins, App Harbor, Team Foundation Server with Git, Visual Studio Online.

CRM, CMS, and others :
Strapi, Contentful, nopCommerce, Shopify, Scribe, Sitefinity, Apigee, Microsoft Intune, Microsoft CRM Dynamics 365, Salesforce, Hubspot, Zoho CRM, ArcGIS, Umbraco, Sharepoint

WORK EXPERIENCE

Technology Lead
Apr 2021 - Present -

Invatu

Dubai

System Architecture and Design: Collaborate with domain experts to understand the specific needs, constraints, and regulatory requirements of E-commerce, Financial, and Healthcare sectors.
Design and architect robust, scalable, and modular software systems that can be adapted to different domains.
Create architectural documentation, design patterns, and guidelines for consistent system development across domains.

Lead Developer
Mar 2022 - Jun 2023

APCOA PARKING Holdings GmbH

Collaborate with product managers and stakeholders to understand requirements and translate them into robust system architecture and design.
Design scalable and reliable software solutions that cater to the complex needs of car park operations, including ticketing, payment, access control, occupancy monitoring, and reporting.

Full Stack Engineer
Apr 2021 - Mar

Erasmus MC

Collaborate with cross-functional teams including product managers, designers, and healthcare professionals to define system requirements and translate them into a robust, scalable, and user-friendly software architecture.
Design and architect the doctor-patient management system to accommodate appointment scheduling, medical records management, prescription generation, communication tools, and other relevant features.

Senior Software Engineer
Jun 2020 - May 2021

Social27 Inc.

Implement WebRTC technologies to facilitate high-quality audio and video conferencing with features like adaptive bitrate streaming, noise cancellation, and echo suppression.
Design and integrate real-time chat, Q&A sessions, polls, and interactive whiteboards to enhance engagement during seminars and virtual events.

Senior Back End Developer
May 2020 - Jan 2021

Authority Partners

Sitefinity CMS Customization: Develop custom modules, widgets, and templates within Sitefinity CMS to support document management functionalities tailored to HR workflows.
Integrate Sitefinity CMS with third-party systems and HR tools, such as payroll systems, applicant tracking systems, and employee portals.

Implement advanced document categorization, tagging, and metadata functionalities to enhance document search and retrieval capabilities.
Design and build version control mechanisms to track document revisions and changes, ensuring data accuracy and compliance.

Frontend Engineer
Nov 2019 - Apr 2020

Summit Wealth Systems

Lead the development of critical components of the wealth and asset management platform, using best practices in coding, testing, and documentation.
Implement features such as portfolio management, risk assessment, investment tracking, and performance analysis to enhance the platform's functionality.

Technology Stack:
Leverage a variety of technologies including but not limited to Java, Python, C#, SQL databases, REST APIs, and cloud services (such as Azure) to build a robust and scalable platform.

Senior Software Engineer
Jan 2018 - Nov 2019

Glasswall

Design, develop, and enhance software systems that specialize in the detection, analysis, and mitigation of file-based threats such as malware, viruses, zero-day exploits, and other malicious content.

Threat Analysis:
Collaborate with cybersecurity experts to analyze emerging file-based threats, understand their patterns, behaviors, and attack vectors, and incorporate this knowledge into the system's threat detection algorithms.

File Analysis and Sanitization:
Develop algorithms and mechanisms to deeply inspect different file formats, identifying any anomalies, malicious code, or hidden threats.
Implement content sanitization processes to neutralize potential threats without compromising legitimate data.

Integration:
Integrate the file-based threat detection and mitigation system with existing security infrastructure, leveraging APIs, protocols, and frameworks to ensure seamless interoperability.

Twilio Expert
Feb 2017 - Jan 2018

AgriSync

Implement Twilio APIs and services to enable SMS, voice, and messaging functionality within the ticket management system.
Integrate Twilio features like two-way communication, message templates, and automated responses to enhance user experience.

Develop the backend infrastructure using appropriate programming languages and frameworks to handle ticket creation, updates, and tracking.
Design databases and data models for storing ticket information, user data, and communication logs.

Associate Technology
Mar 2015 - Feb 2017

Publicis Sapient

Implement features and functionality for ticket purchasing, management, and validation in the Ferrari World ticket management system.
Develop features for Ladbrokes' online betting platform, including user account management, odds calculation, bet placement, and result processing.

Conduct thorough testing, including unit testing, integration testing, and end-to-end testing, to ensure the reliability and correctness of the software.
Collaborate with QA engineers to identify and resolve bugs and issues promptly.
Implement automated testing strategies to maintain code quality and stability.

Software Developer
Mar 2013 - Mar 2015

Nomor AB

Implement mechanisms for capturing diverse pest-related data, including species identification, population levels, and environmental factors.
Develop algorithms to analyze collected data, identify trends, and generate insights to inform pest management decisions.
Integrate data visualization tools to present findings in a clear and actionable manner.

Collaborate with external systems and APIs to enhance the system's capabilities, such as weather data integration or geographic information systems (GIS) integration for spatial analysis.
Ensure cross-platform compatibility to support both desktop and mobile devices for field data collection.

EDUCATION

B.Tech - Computer Science

Azad Institute of Engineering and Technology

PROJECTS

Expandopedia.com

Helping companies in global employment and compliance

Expandopedia.com is an exciting software development project that combines the MERN (MongoDB, Express.js, React, Node.js) stack with Sitefinity CMS to create a powerful and dynamic online platform for international business expansion resources. This project overview provides an insight into Expandopedia.com, outlining its objectives, key features, and the technologies utilized to develop an information-rich and user-friendly website.

Project Objectives
The primary objectives of the Expandopedia.com project are as follows:
a. Develop a comprehensive online platform that offers valuable resources and insights for businesses planning to expand internationally.
b. Implement a user-friendly and visually appealing interface to provide seamless navigation and enhance user engagement.
c. Utilize Sitefinity CMS to enable content authors and administrators to easily create, manage, and publish new content.
d. Integrate the MERN stack to handle dynamic features and interactivity, such as user authentication, real-time search, and personalized recommendations.
e. Ensure scalability and high performance to accommodate a growing user base and increasing content resources.
f. Apply responsive design principles to optimize the website's accessibility on various devices.
Key Features
Expandopedia.com will offer a range of features to facilitate international business expansion and empower users with valuable resources. Some of the key features include:
a. Content Management: Sitefinity CMS integration to allow content authors to create, edit, and publish articles, guides, and resources.
b. User Profiles and Authentication: Secure user registration and authentication to provide personalized content recommendations and user-specific resources.
c. Resource Library: A comprehensive library of articles, tools, templates, and country-specific guides for business expansion.
d. Search and Filtering: Advanced search functionality with real-time suggestions and filtering options to help users find relevant information.
e. Multilingual Support: Support for multiple languages to cater to a diverse global audience.
f. News and Updates: Regularly updated news and insights on global business trends and expansion opportunities.
g. Interactive Forums: User discussion forums to foster a community of like-minded businesses and experts.
h. Responsive Design: A mobile-responsive layout for seamless access on various devices, including desktops, tablets, and smartphones.

Agrisync.com

Support management platform

The primary objectives of the agrisync.com project are as follows:
a. Develop a comprehensive platform that facilitates effective communication and collaboration among farmers, agricultural experts, and stakeholders.
b. Provide real-time data visualization and analytics for agricultural activities, such as crop management, livestock tracking, and weather monitoring.
c. Implement a scalable and secure system that accommodates agricultural data from various sources and ensures data privacy.
d. Utilize GraphQL to optimize data retrieval and enable efficient communication between the front-end and back-end components.
e. Create an intuitive user interface with interactive features for seamless navigation and ease of use.
f. Enable integration with third-party agricultural services and APIs to enhance functionality and expand the platform's capabilities.

Technology Stack

Node.js: A server-side JavaScript runtime for building scalable applications.
Express.js: A minimalist web application framework for Node.js, simplifying routing and middleware creation.
MongoDB: A NoSQL database for flexible and efficient data storage.
Mongoose: An Object Data Modeling (ODM) library for MongoDB, providing schema-based data modeling.
GraphQL: A query language for APIs that enables efficient data retrieval and offers flexibility to clients.
b. Front-End:
React: A modern JavaScript library for building interactive user interfaces.
Redux: For managing application state and ensuring predictable data flow.
HTML and CSS: For structuring and styling the user interface.
c. Deployment and Hosting:
Docker: For containerization and ease of deployment.
AWS

Social27.com

Virtual Event Platform for Conferences

The primary objectives of the Social27.com project are as follows:
a. Develop a feature-rich and user-friendly social networking platform to connect individuals and communities.
b. Provide users with a personalized and interactive environment to share content, connect with friends, and join communities of interest.
c. Enable seamless communication through instant messaging, video calls, and other interactive features.
d. Implement advanced privacy and security measures to protect user data and maintain user trust.
e. Ensure high performance, scalability, and availability to accommodate a growing user base.
f. Incorporate gamification elements to enhance user engagement and promote user-generated content.

Technology Stack

a. Back-End:
.NET Framework or .NET Core: A powerful framework for building robust and scalable applications.
C#: A versatile and modern programming language for creating high-performance applications.
ASP.NET MVC: A web application framework for building model-view-controller (MVC) pattern-based applications.
Entity Framework: An object-relational mapping (ORM) tool for data access and management.
Microsoft SQL Server: A relational database management system for efficient data storage.
b. Front-End:
HTML, CSS, and JavaScript: For structuring and styling the user interface and implementing interactive elements.
Razor Views: For server-side rendering and generating dynamic HTML in ASP.NET MVC.
c. Deployment and Hosting:
AWS:

Vourity.com

Coupon management Platform

The primary objectives of the Vourity project are as follows:
a. Develop a feature-rich and user-friendly platform that connects users with local events and activities.
b. Enable event organizers to create, manage, and promote events seamlessly.
c. Provide users with personalized recommendations and a streamlined event discovery process.
d. Implement real-time updates and notifications to keep users informed about upcoming events.
e. Ensure high performance and scalability to accommodate a growing user base and increasing event listings.
f. Incorporate robust security measures to safeguard user data and privacy.

Technology Stack

a. Back-End:
Node.js: A powerful server-side JavaScript runtime for building scalable applications.
Express.js: A minimalist web application framework for Node.js, simplifying routing and middleware creation.
MongoDB: A NoSQL database for flexible and efficient data storage.
Mongoose: An Object Data Modeling (ODM) library for MongoDB, providing schema-based data modeling.
Cloud Hosting: Azure

Ferrari World - Theme park

Tickets booking platform

ferrariworldabudhabi.com will showcase a range of features to engage users and promote Ferrari World Abu Dhabi's offerings. Some of the key features include:
a. Home Page: A visually captivating homepage with interactive elements to showcase key attractions and upcoming events.
b. Rides and Attractions: Detailed descriptions, images, and videos of each ride and attraction available at the theme park.
c. Events and Promotions: Real-time updates on upcoming events, promotions, and special offers.
d. Ticket Booking: Integration with external services for a smooth and secure ticket booking process.
e. Interactive Map: An interactive map of the theme park, enabling visitors to plan their itinerary efficiently.
f. Virtual Tours: Immersive virtual tours of select attractions to give users a taste of the experience.
g. Contact and Support: Contact information and support channels for inquiries and assistance.
h. Responsive Design: A mobile-responsive layout for seamless access on various devices.

Technology Stack

a. Back-End:
.NET Core: A modern and cross-platform framework for building scalable web applications.
C#: A powerful and versatile programming language for developing robust back-end logic.
ASP.NET Core: A web application framework for building high-performance web APIs.
Microsoft Azure: A cloud computing platform for hosting the application and managing resources.
Azure SQL Database: A fully-managed, cloud-based relational database for data storage.
b. Front-End:
React: A popular JavaScript library for building interactive user interfaces.
HTML and CSS: For structuring and styling the user interface.
Redux: For state management, ensuring a predictable and scalable application.
c. Deployment and Hosting:
Azure App Service: For deploying and hosting the web application.
Azure Content Delivery Network (CDN): For delivering content efficiently to users across the globe.

Canteenhub

Project Objectives
The primary objectives of the Canteenhub project are as follows:
a. Develop a user-friendly and efficient canteen management system.
b. Provide a platform for canteen administrators to manage menu items, orders, and inventory.
c. Enable canteen visitors to view the menu, place orders, and make payments online.
d. Implement real-time order tracking and notifications for users and administrators.
e. Ensure a seamless user experience through a responsive and intuitive web interface.
f. Apply modern security practices to protect user data and prevent unauthorized access.

Technology Stack
a. Front-End:
React: A modern JavaScript library for building interactive user interfaces.
Redux: For state management, ensuring a predictable and scalable application.
HTML and CSS: For structuring and styling the user interface.
b. Back-End:
Node.js: A server-side runtime environment for building scalable applications.
Express.js: A fast and minimalistic web framework for Node.js.
MongoDB: A NoSQL database for efficient data storage and retrieval.
c. Deployment and Hosting:
Docker: For containerization and ease of deployment.
AWS

APCOA PARKING Holdings GmbH

Project Scope:
Performance Optimization: Analyze the current system to identify performance bottlenecks and implement optimizations to ensure the platform handles increased load efficiently. This may involve optimizing database queries, implementing caching mechanisms, and utilizing asynchronous programming techniques.

Scalability: Enhance the system's architecture to support horizontal scalability. Integrate technologies like load balancers and containerization (e.g., Docker) to ensure the system can handle a growing number of users and parking lots.

New Features Implementation:

Reporting and Analytics: Develop comprehensive reporting and analytics tools for Apcoa administrators. This includes generating insights about parking occupancy, revenue trends, and customer behavior, aiding in data-driven decision-making.

Security Enhancements: Implement robust security measures, including data encryption, user authentication, and authorization mechanisms, to safeguard sensitive user information and prevent potential security breaches.

Technology Stack:

Node.js: As the primary backend language for server-side development.
Express.js: As the web application framework.
MongoDB: As the NoSQL database for storing parking-related data.
Redis: For caching and improving performance.
React.js/Angular/Vue.js: As the frontend framework for building dynamic user interfaces.
Docker: For containerization and deployment.
AWS/Azure/GCP: For cloud hosting and infrastructure management.
Git: As the version control system for code management.
Project Deliverables: