

Madhava Reddy Vemireddy

+91 8886380746

Vemireddyomr@gmail.com

in Madhava Reddy Vemireddy

maddytech.rf.gd

No 16, 18th Cross 6th Main, BTm 2nd Stage, Bangalore, 560076

Summary

- Full Stack .NET Developer with **3.7+ years** of experience in designing and delivering scalable web applications and REST APIs.
- Proficient in front-end technologies: **ReactJS**, **Redux**, and **TypeScript**.
- Skilled in back-end development with **ASP.NET Core**, **C#**, and **Entity Framework**.
- Experienced in cloud technologies, including **AWS (RDS, S3)** and familiarity with **Azure** services.
- Hands-on expertise in **CI/CD pipelines** and performance monitoring tools like **Splunk** and **Grafana**.
- Committed to building **high-quality, maintainable software solutions** in agile development environments.

Education

- **B.Tech in Computer Science**, *RISE Krishna Sai Prakasam Group Institutions* (June 2017 - Aug 2021) — GPA: 7.73/10.0
- **Intermediate**, *Narayana Jr College, Kadapa* (2015 – 2017) — GPA: 93.3/100
- **SSC**, *Vijaya High School, Porumamilla* (2015) — GPA: 87/100

Experience

Amiti Software Technologies Pvt. Ltd., Software Developer

Bengaluru

June 2021 to Present

- Built reusable UI components using **ReactJS** and **TypeScript** for scalable and consistent application design.
- Managed state efficiently with **Redux**, ensuring smooth data flow across modules.
- Designed and integrated **REST APIs** using **ASP.NET Core**, handling business logic and service layers.
- Developed responsive UIs using **CSS Grid**, **Flexbox** for better user experience.
- Ensured code quality through **Jest** unit testing and regular **code reviews**.

SKILLS

Front-End : ReactJS, Redux, TypeScript, JavaScript, HTML5, CSS3, Bootstrap, Jest, Styled-components

Back-End: ASP.NET Core, C#, RESTful APIs, Entity Framework, LINQ, Middleware, JWT Authentication

Databases: PostgreSQL, Oracle, MySQL

CLOUD PLATFORMS: AWS (RDS, S3)

CI/CD TOOLS: Azure Pipelines, Jenkins

MONITORING ANALYTICS TOOLS: Splunk, Grafana, Honeycomb, Cribl

VERSION CONTROL GitHub, Azure Boards, Jira

SOFTWARE IDEs: Visual Studio, VS Code, Postman, Swagger

OTHER TOOLS: OpenShift Container Platform (OCP), Azure Wiki's

Projects

Vehicle Auction Platform – Search and Purchase Optimization

Overview: Developed and optimized a vehicle auction platform for automotive financial companies to improve search, bidding, and purchasing workflows.

Key Features & Contributions:

- **Search 2.0:**
 - **Objective:** Enabled advanced vehicle search with customizable filters and saved search functionality.
 - **Backend:** Developed REST APIs in .NET Core for dynamic filtering and user preferences.
 - **Frontend:** Created reusable ReactJS components (**Filter & Saved Search Modal**) for seamless user interactions.
- **Bid Buy Migration:**
 - **Objective:** Improved the vehicle bidding and purchasing workflow, supporting various options like direct purchases, best offers, and autobids.
 - **Backend:** Created robust REST APIs to manage vehicle details, bidding, purchases, and add-ons like warranties and transport options.
 - **Frontend:** Developed dynamic ReactJS components (Vehicle Summary, Bid Summary, Transport Options) and integrated add-on services.

Technologies Used:

- **Frontend:** ReactJS, styled-components for styling, Redux for state management, CSS Grid for layout
- **Backend:** .NET Core, Entity Framework for ORM, Split.io for feature flagging.
- **Database:** PostgreSQL using AWS RDS

HCC Loan Payoff Platform

Overview: Developed a loan payoff platform for Hyundai Capital Canada, enabling users to view loan details, calculate fees, and submit requests.

Key Features & Contributions:

- **Backend Development:**
 - **Objective:** Built and maintained REST APIs in .NET Core to manage loan details, payment fees, and payoff requests.
 - **Core APIs:** LoanDetails (GET), LoanLineItem (GET), LoanPayOff (POST).
 - **Outcome:** Enabled real-time loan processing and fee calculations, optimizing backend efficiency.
- **Frontend Development:**
 - Created a responsive ReactJS interface for loan details, payment options, and submissions.
 - **Features:** VIN-based loan retrieval, dynamic loan breakdown, and seamless transaction flows.

Technologies Used:

- **Frontend:** ReactJS, styled-components for styling, Redux for state management, CSS Grid for layout
- **Backend:** .NET Core, Entity Framework for ORM
- **Database:** PostgreSQL using AWS RDS

Independent Projects

React Micro-Frontend (MFE) Portfolio

Project URL: maddy-shell.s3-website.ap-south-1.amazonaws.com [↗](#)

Project Overview: Developed a comprehensive React-based Micro-Frontend (MFE) portfolio, featuring personal information, skills, education, experience, and contact details. This project was structured using three key React repositories: `maddy-widgets`, `maddy-login`, and `maddy-shell`, which were deployed on AWS S3 using GitHub Actions. The backend services, including account management and email functionality, were handled by a separate **.NET API repository**.

Key Features & Contributions:

- **Main Features:**

- Designed and developed a variety of reusable UI components, including portfolio cards, contact forms, widgets, and navigation elements (header and footer).
- Implemented user login, signup, and password recovery operations using the `maddy-login` repository, integrated with backend **account management APIs**.
- Integrated email OTP verification for account management, leveraging an SMTP server to send real-time notifications.

- **Micro-Frontend Architecture:**

- Employed a modular architecture with the following repositories:
 - * `maddy-widgets`: Contains reusable UI components.
 - * `maddy-login`: Handles user authentication.
 - * `maddy-shell`: The main repository that ties everything together and loads the components dynamically using **Webpack with lazy loading**.

- **Deployment & CI/CD:**

- Deployed the frontend to AWS S3 for scalable hosting.
- Utilized **GitHub Actions** for automated deployment and smooth CI/CD workflows.

- **Backend API Integration:**

- The backend, including account management and email services, is powered by a **.NET API repository** which provides essential services like login, signup, and email notifications.
- Access the **API documentation** for the backend services on [Account Management Email Service API Swagger](#) [↗](#).

- **Contact Form Email Integration:**

- Designed the contact form widget to collect user queries and send confirmation emails via the integrated SMTP email service, delivering personalized "Thank You" messages to users.

Technologies Used:

- **Frontend:** ReactJS, Webpack (with lazy loading), GitHub Actions
- **Backend:** .NET Core (APIs for account management, email services using SMTP, JWT Authentication)
- **Deployment:** AWS S3, GitHub Actions


IRASAH (React Front-End App)

Project Overview: Built a static training institute application with **React**, designed with dynamic pages and responsive features to enhance user experience.

Key Features & Contributions:

- Developed a user-friendly, mobile-responsive interface for a training institute's website.
- Integrated **Email.js** to handle contact form submissions.
- Deployed the application publicly for users to access.

Technologies Used:

- **Frontend:** React, Email.js
- **Deployment:** Publicly hosted on irasah.com .