-MADHUKAR

SR. REACT JS DEVELOPER

[madhu.ganga2018@gmail.com](mailto:madhu.ganga2018@gmail.com)

+1 984-377-1236

**PROFESSIONAL SUMMARY:**

* 10+ years of overall experience, with 8 years of experience as a senior React developer.
* Highly skilled React JS developer with 8 years of experience in building robust and user-friendly web applications. I am seeking a challenging role to utilize my expertise in React JS, along with proficiency in Docker, JIRA, Slack, GitHub, GitLab, and CI/CD, to contribute to the success of a dynamic and innovative development team.
* Strong expertise in web application development using modern JavaScript frameworks, libraries, and tools such as React.js, **Next.js**, **React-hook**, **Redux**, **Thunk**, **Redux-saga**,**, React-hook-form**, **storybook**, **Tailwind** **CSS**, **Material UI**, **Styled-Component**, **Typescript**, **GraphQL**, **Figma,** and **Invision.**
* Experienced in developing common components and documenting using **Storybook** to promote code reuse, enhance development efficiency, and ensure consistent user experiences across multiple projects.
* Skilled in front-end development with **HTML5**, **CSS3**, and **responsive design** principles
* I am proficient in writing unit tests using the **Jest** and **React Testing Libraries** to ensure the quality and stability of React components and applications.
* Experience in using build and deploy tools such as Jenkins and **Docker** for continuous integration.
* I am skilled in defining and using interfaces and types in **Typescript** to create explicit contracts between different parts of the codebase, ensuring consistency and reducing runtime errors.
* Proficient in utilizing TypeScript's advanced features, such as **generics**, decorators, and conditional types, to write more expressive and reusable code.
* Expertise in conducting thorough **code reviews**, providing constructive feedback, and ensuring adherence to coding standards, best practices, and scalability principles to maintain a high-quality codebase.
* Good experience in React JS for creating interactive UI’s using one-way data flow, virtual DOM, JSX, and React Native concepts.
* Experience in the Agile/Software **Development** Lifecycle (SDLC) right from information architecture, requirement gathering and analysis, documentation, wireframes, style guides, look-and-feel, and final layout to maintenance of proposed applications.
* Proficient in utilizing collaboration software tools such as **JIRA**, **Slack**, **Rally,** and **Microsoft Teams** to effectively communicate and collaborate with distributed teams, track project progress, manage tasks, and ensure smooth coordination.
* Continuously learning and staying up to date with the latest trends and best practices in modern frontend development, CI/CD methodologies drive innovation and deliver cutting-edge solutions.

**TECHNICAL SKILLS:**

* React Js, Js, CSS
* Redux, Thunk, Redux-Saga, Zustand
* Jest, Mocha, Chai, testing-library
* Typescript, Graphql, REST
* Webpack, Babel
* Material UI, Tailwind CSS
* Figma
* Gitlab, GitHub
* JIRA, Slack, and Agile Methodologies

## PROFESSIONAL EXPERIENCE:

**CLIENT: PEARSON, HYBRID OCT 2024 – PRESENT**

**Role: SR UI DEVELOPER**

**Description:**

**Centralized login: Customizable application developed using React based on Partner need.**

1. **Certiport.**
2. **Compass local**
3. **Certiport cloud Scheduler.**
4. **Certiport cloud Candidate.**

**AMXUI: ForgeRock.**

**Access Management &SSO:**

**Enables secure single sign‑on, multi‑factor & password less authentication.**

**Single Sign in: We been implemented Sign in with code authenticated the user using Email and SMS. Once it authenticated one of those asking the user MFA to add next level of authentication based on Partner (REVIBE, KIBO, HK LONGMAN).**

**Progressive Profile: Allow the user to add Mobile number after the registration.**

**We are allowing most of the countries. Once it gets verified get the success. Users enter 5 times wrong OTP will get error screen.**

**User can skip mobile screen won’t need it we can see only 7days after cannot see the mobile screen. Then it will navigate to user Profile Screen.**

**Responsibilities:**

* Create layouts that adapt seamlessly to various screen sizes (desktop, tablet, mobile).
* Translate wireframes, mock-ups, and prototypes into high-quality code.
* Mastery of core technologies to structure, style, and add interactivity to applications.
* Use libraries React for building component-based applications.
* Minimize page load times by optimizing assets, lazy-loading, and reducing DOM manipulation.
* Write efficient and reusable code to avoid redundancy.
* Ensure the application works smoothly across browsers (Chrome, Firefox, Safari, etc.) and devices.
* Support multiple languages, including handling RTL (Right-to-Left) text for languages like Arabic, Hebrew and Urdu.
* Integrate APIs and back-end logic into the front-end interface.
* Work in Agile environments, attending stand-ups, and participating in sprint planning.
* Debug and resolve UI issues during development and post-production.
* Use browser tools for testing layouts and tracking down performance bottlenecks.
* Manage code repositories, create branches, and resolve merge conflicts.
* Work with tools like Jenkins, and GitHub for automated testing and deployment.
* Maintain thorough documentation for your codebase, and component libraries to ensure future scalability and ease of onboarding.
* Understand and align UI with ForgeRock AM authentication trees (journeys).
* Customize screen flows and UI behavior based on step responses.
* Customized ForgeRock AM (Access Management) screens such as:
* Login
* Registration
* Forgot Password
* Multi-factor Authentication (MFA)
* **Sign in code, Progressive Profile.**
* Develop reusable and modular React components using functional or class-based approaches
* Manage component state and side effects using **React hooks** (use State, use Effect, etc.)
* Implement **client-side routing** with **React Router**
* Handle conditional rendering, form inputs, and dynamic data-driven UIs
* Write clean, efficient JavaScript (ES6+) to manage DOM events, form validation, business logic, and API interactions
* Handle asynchronous operations using async/await, Promises, or **Axios/Fetch**
* Manipulate arrays and objects with modern JS features like map, filter, reduce, destructuring, etc.
* Connect frontend components to backend services via **RESTful APIs**
* Handle authentication/authorization (e.g., login flows, protected routes)
* Manage API error handling, response parsing, and edge cases
* Use **React Context API** to manage global application state
* Write unit tests using **Jest** and component tests using **React Testing Library**
* Debug UI and network issues using browser dev tools
* Conduct cross-browser and device testing for responsive behavior.
* Develop modular, reusable React components using functional or class-based approaches.
* Implement user interfaces based on design specifications (e.g., from Figma, Adobe XD)
* Fetch and handle data from RESTful using **Axios**, fetch, or React Query
* Manage asynchronous requests, loading states, and error boundaries
* Securely handle user authentication and token-based authorization
* Use **Git** for version control and tools like **Jira Cloud**, **Trello**, or **Azure DevOps** for task tracking

**CLIENT: COX AUTO INC. Remote AUG-2021 – OCT 2024**

**Role: Senior React. Js Developer**

**Responsibilities:**

**Project Description:** Cox Automotive Mobility specializes in comprehensive fleet services and operations. They provide a SaaS solution for the customers to keep fleets moving safely and sustainably for the next generation. Maximize fleet uptime with the premier partner in maintenance service. Manage scheduled and emergency maintenance.

The project is to develop an interface for customers to manage vehicle information, schedule preventative maintenance, define an automated service schedule, generate reports and analytics, etc.

**Key Results Areas:**

* Developed scalable and responsive web applications using React, Redux, Thunk and React Router, Next Js, NodeJS, Js, jQuery.
* Developers use ESLint to ensure that their code adheres to the established coding standards and best practices.
* Collaborating with front-end and back-end teams to ensure the GraphQL API meets client needs.
* Working with ECMA Script 6 features for the build of JSX and ES2015 (ES6), we used Babel and Webpack.
* ES6 features like arrow functions, template literals, and destructuring enhance code clarity and reduce boilerplate.
* Integrating templates with build tools and task runners such as Gulp or Webpack to streamline the development process.
* Built reusable components and maintained a consistent UI/UX across applications using React.js, while drawing on experience with Ember.js to handle complex routing and state management.
* Utilized knowledge of Ember.js to inform best practices in React.js development, enhancing application structure and state management.
* Managing updates and revisions of Markdown documents using version control systems like Git.
* Developed dynamic, responsive web applications using both React.js and Ember.js, allowing for a deeper understanding of component-based architecture and MVC patterns.
* Ensuring the GraphQL schema, which outlines the data structure and how clients can query it.
* Creating interactive and responsive web applications using JavaScript, HTML, and CSS, React 18.
* Building user interfaces and improving the user experience through client-side scripting.
* ES6 is widely supported by modern browsers and Node.js environments, ensuring broader compatibility for your React applications.
* Use **React JS18**, **classes,** and **functional** components (including hooks) in JSX, which combines HTML and JavaScript.
* Ensure the application adheres to security best practices, particularly around user data, checkout processes, and payment handling.
* Use APIs to retrieve and display real-time data, like product availability, pricing, and promotions.
* Build and maintain interactive, responsive, and visually appealing user interfaces for product browsing, checkout processes, and user accounts.
* Implement error handling within AWS Lambda functions to gracefully handle failures and provide meaningful error messages.
* Ensure the retail site or application can handle high traffic, especially during peak times like holidays and promotions.
* Specify the entry point for your function where AWS Lambda can start execution.
* Manage user sessions and implement secure access to resources using AWS.
* Implemented state management using Redux, ensuring efficient data flow and predictable application behavior.
* Set up and configure the build process to compile LESS files into regular CSS.
* Use LESS variables to store and reuse values such as colors, font sizes, and spacing.
* Utilizing Cypress’s capabilities to mock and stub network requests, ensuring tests are not dependent on external services.
* I worked closely with backend developers to integrate frontend components with RESTful and GraphQL APIs, ensuring smooth data flow and optimal functionality.
* Participated in code reviews, providing constructive feedback, and ensuring adherence to coding standards.
* Collaborated with the QA team to identify and resolve bugs and performance issues through comprehensive testing.
* Utilized JIRA for project management, tracking tasks, and ensuring timely completion of deliverables.
* Actively participated in Agile/Scrum methodologies, contributing to sprint planning, daily stand-ups, and retrospectives.
* Develop and manage the shopping cart experience, ensuring data is stored correctly across sessions (e.g., items remain in the cart if the user navigates away and returns).
* Designed and implemented reusable components and libraries; documented those using Storybook, improving code reusability and reducing development time.
* Implemented custom hooks to encapsulate reusable logic and promote code reusability across different parts of the application.
* Act as a bridge between various stakeholders, such as product, design, and engineering, to ensure cohesive project execution.
* Ensuring that micro frontends are responsive, accessible, and performant.
* I worked on using React native components, forms, events, keys, navigation, and Redux concepts.
* Writing and maintaining Jest unit tests to ensure individual components and functions work as expected.
* Making sure all interactive elements are navigable and usable using keyboard input only.
* Integrating Jest tests into CI/CD pipelines to automate testing and ensure code quality.
* Integrate assets exported from Figma (images, icons) into the React application.
* Using Jest's debugging tools to identify and resolve issues in the application.
* Ensuring that focus indicators are visible and properly styled for keyboard users.
* Ensuring that the application is robust, with adequate error handling and recovery paths.
* Media queries were used for integrating dashboard components like the sidebar and adjusting the UI for various screen sizes.
* Incorporating responsive design using a media query in a React dashboard.
* Implement voice and video calling functionality using Twilio Programmable Voice and Video services.
* Use functional JavaScript to develop front-end applications.
* Optimizing Just tests to run efficiently and reduce execution time.
* Use **Node.js** to run Grunt tasks and build the project properly (compile, minify, etc.).
* I worked on GIT as a version control. Utilized NPM scripts for build and test environments in a **Node**.js application.
* Implement Redux saga to manage complex, multi-step workflows where actions are dispatched based on conditions or the outcome of previous actions.
* Integrate Redux-Saga with the existing Redux store to maintain a predictable state across the application
* Write Redux saga to perform asynchronous tasks like fetching data from remote APIs or services and dispatching the appropriate Redux actions.
* Decide between client-side rendering based on the requirements of the project and ensure efficient use of Next.js rendering methods.
* Used Ember.JS for its built-in data library for better routing capacity and handling of asynchronous calls
* Responsible for defining and managing GraphQL queries, mutations, and subscriptions.
* Developed UI components for email and link sharing of documents and files for a content management system using React Promises.
* Develop React applications following best practices, ensuring responsiveness, and integrating with serverless components.
* Create intuitive filtering and sorting options for products, helping users find items by category, price range, color, or other criteria.
* Use Redux-Saga to handle side effects such as API calls, timeouts, background processing, and other asynchronous operations.
* Collaborate with developers, QA, and IT operations to streamline development workflows.
* Implement DevOps best practices like automated testing, shift-left strategies, and continuous feedback.
* Maintain and govern **Git repositories**, enforce branching strategies like GitFlow.
* Manage pull requests, code reviews, and integration workflows.
* Use Azure Boards to manage work items, sprints, backlogs, and dashboards.
* Define custom workflows and process templates as per team/project needs.
* Test the application on different browsers regularly to identify and address any rendering or functionality issues specific to each browser (Safari, Chrome, and Edge).
* Linters can be integrated into build processes, IDEs, or code editors to provide real-time feedback to developers.
* In retail, react developers focus on reliability, security, and performance to create an intuitive and efficient shopping experience. Each feature aims to enhance usability, support conversions, and meet high standards in a competitive online marketplace.
* Ensure that all React components are fully responsive and perform well on various devices, particularly mobile, where a significant portion of retail traffic originates.
* Writing client-side JavaScript code for user interface components and interactions.
* Create comprehensive documentation for APIs using Postman.
* Design and implement user interfaces (UI) for interacting with Twilio services, such as messaging, voice calls, video calls, or other communication features.
* Consume APIs provided by AWS Lambda to fetch and send data.
* Handle API responses, errors, and loading states effectively.

**CLIENT: STATEFARM,** **Richardson, TX MAR-2020 – AUG-2021**

**ROLE: SENIOR REACT DEVELOPER**

**Responsibilities:**

**Project Description: AQP** (Auto Quote Purchase) is an automatic quotation management platform from State Farm, enabling its users to explore, prepare, submit, negotiate, and purchase different insurance products, like auto insurance, home insurance, medical insurance, etc.

Front-end developed in React JS, some user-facing pages are written in plain JSP, and backend apis are developed with a Spring-based framework.

**Key Result Areas:**

* Developed React.js components and implemented them into the web applications, improving user interactivity and responsiveness.
* Implemented state management using Redux, ensuring efficient data flow and application performance.
* Utilized React Router for creating dynamic and seamless navigation within single-page applications.
* Employed modern JavaScript frameworks and libraries, such as Axios and Lodash, to streamline development processes.
* Uses Visual Studio Code to create and manage React components, leveraging IntelliSense and snippets for efficiency.
* Use Next.js next/image component to serve optimized images, reducing loading times and improving the overall performance of the application.
* Used Ember.JS for its built-in data library for better routing capacity and handling of asynchronous calls
* Use HTTPS to ensure secure communication between React and Lambda.
* Implement security best practices, such as input validation and encryption.
* Utilizes extensions like Visual Studio Code Styled Components for writing CSS-in-JS or other styling methodologies.
* Leverage TypeScript’s type system to catch bugs during testing before they become runtime errors.
* Utilizes the Debugger for Chrome extension for debugging React applications directly within VS Code.
* Conducted unit testing and integration testing of React.js components, ensuring high-quality and bug-free code.
* Integrate third-party libraries with TypeScript, ensuring proper type definitions through DefinitelyTyped (@types).
* Debug and troubleshoot issues in sagas by using Redux-Saga’s built-in tools and logging capabilities.
* Utilized version control systems, such as Git, to manage source code and streamline collaboration within the team.
* Use **Azure API Management** to secure, monitor, and scale APIs consumed by React apps. Apply rate limiting, quotas, and logging for API access.
* Encourage an environment where teams can freely share information and resources, helping to solve cross-functional challenges.
* Help team members solve complex technical problems by offering guidance, best practices, and sharing expertise.
* Configure and customize **Vite.js** for fast development and optimized production builds.
* Handle **ES module bundling**, **code splitting**, and **tree-shaking**.
* Implement advanced features like **pre-bundling**, **lazy loading**, and **HMR** (Hot Module Replacement).
* Set up new React/Vue/Svelte projects with Vite.
* Integrate Vite with TypeScript, Sass, TailwindCSS, and other modern frontend tooling.
* Define environment variables and manage .env configurations securely.
* Leverage or build **custom Vite plugins** to extend functionality.
* Integrate third-party plugins (e.g., vite-plugin-pwa, vite-plugin-compress, etc.) for specific features.
* Leverage **Azure Cache for Redis** to cache API responses and optimize the performance of data heavy React applications, reducing database load.
* Maintained JSP pages, ensuring adherence to industry standards and an optimal user experience.
* Actively participated in continuous learning, staying updated with the latest React.js trends and best practices.
* Encourage continuous learning and professional development through training sessions, workshops, and code reviews.
* Used Ember.js for a robust data layer and interactive web design.
* Leveraging S3 Transfer Acceleration for faster content upload and download by users globally.
* Developed web pages to comply with Web Content Accessibility Guidelines (WCAG) and the ability to apply W3C web standards, including NVDA tools.
* Consume data from backend APIs (implemented in Rails) using asynchronous requests.
* Writing, testing, and deploying code to AWS.
* Customizes Material UI themes to match the project's branding, using theming options and CSS-in-JS solutions.
* Use Gatsby source plugins to connect to content management systems (CMS) like WordPress, Contentful, Sanity, or Strapi for dynamic content.
* Leverage Gatsby’s built-in features to optimize CSS, JavaScript, and other static assets, ensuring that the website runs efficiently across different devices and browsers.
* These components use GraphQL queries to fetch data from the server. They are typically placed in container components or directly in functional components using hooks.
* ReactJs application with GraphQL in this way, you can maintain a clear separation of concerns, enhance code reusability, and effectively manage roles and responsibilities across different layers of your application.
* Implement **routing**, **linking**, and page creation using Gatsby APIs.
* Develops reusable and responsive UI components using Material UI and React.
* Integrating applications with AWS services.
* Ensure smooth communication between Ember.js and React components. You may need to pass data or trigger actions between them using props, events, or a state management solution like Redux.
* Express.js is used to create the backend server and define API routes, React is used to build the user interface components and manage the frontend state.
* Node.js is used to run the server-side code and handle HTTP requests.
* Collaborate with developers, testers, and other stakeholders to define API requirements.
* Integrate Postman with version control systems like Git for team collaboration.
* Provide clear documentation and guidelines for developers working with both Ember.js and React to streamline development and collaboration.
* Collaborating with other teams on application architecture.
* Build UI components for checkout, payment forms, and any necessary user interactions using React components.
* Redux saga it listens for dispatched actions and performs corresponding side effects before dispatching new actions.
* Side effect logic complements Redux's reducer-based state updates.
* Use advanced TypeScript features like union types, mapped types, and intersection types to write more expressive and safe code.
* It coordinates complex async logic (e.g., call sequence, parallel processing, cancellation) in a clean and testable way.
* Optimize navigation using Next.js built-in Link component for client-side transitions and smooth navigation across the app.
* Ensure sagas integrate well with action creators and reducers.
* Redux Saga handle complex async flows (e.g., API calls, timeouts, retries).
* Implement robust error handling within Redux saga, ensuring that any failures in asynchronous tasks are captured and managed properly.
* Redux saga Catch errors during side effects and dispatch failure actions.
* Use Next.js’s file-based routing system to organize pages and set up dynamic routing easily without the need for third-party routing libraries.
* Manage state and user input using the React state or context API.
* Validate user input before submitting payment details.
* Verify the payment request details received from the client.
* Communicate with the payment gateway's server-side API to process payments securely.
* Handle any necessary authentication and authorization with the payment gateway.
* Handle client-side interactions with the payment gateway's API.
* Send payment request details securely to the server for processing.
* Implement client-side routing, state management, and other frontend functionalities using React and Next.js features.

**CLIENT: FIDELITY INVESTMENTS, Durham, NC JUN-2019 – MAR-2020**

**ROLE: SENIOR UI DEVELOPER**

**Responsibilities:**

**Project Description:** Working on ReactJS-based web applications providing user interfaces for different functionalities like the PI-Login Experience, 2FA, Security Center, and NUR related to the user security of Fid.com. Additionally, we are working on the inactivity timeout and DBS (Delegated Brokerage Service) enhancements, DAE, dashboards that are part of Fid.com, and moving applications on-premises to AWS.

**Key Result Areas:**

* Communicating with stakeholders to understand their needs and requirements.
* Incorporated React Hooks with a gradual adoption and migration strategy.
* Migrated major class-based containers to Redux connected functional component containers, allowing for compatibility with newer functional component libraries, while introducing react-redux hooks into newer containers to maintain React Redux global state management.
* Set up and manage CI/CD pipelines using AWS tools.
* Use tools like **Lighthouse**, **WebPageTest**, and **Gatsby Cloud reports**.
* Write clean, maintainable, and efficient ES6 code.
* Write clean, scalable, and maintainable code using TypeScript for frontend React or backend (Node.js) applications.
* Develop fast, SEO-friendly websites using **GatsbyJS** and **React**.
* Migrate existing JavaScript projects to TypeScript, ensuring proper type safety across the codebase.
* Ensure that the code follows best practices for maintainability, modularity, and scalability in TypeScript.
* Implement dynamic data fetching using GraphQL to source data from various APIs, CMSs, or local files (Markdown, JSON) and integrate it into static pages.
* Build page components that Gatsby can pre-render and ensure that each page is optimized for static rendering and is SEO-friendly.
* Use **Gatsby** to create static or hybrid sites with optimal performance.
* Manage **Gatsby config**, plugins, and theming.
* Manage the canvas state and ensure efficient rendering and updating of graphics.
* Develop user interfaces using modern web technologies.
* Stay updated with the latest ES6 features and best practices.
* Utilize ES6 features to build dynamic and interactive user interfaces.
* Worked on GIT as a version control. Utilized NPM scripts for build and test environments in a Node.js application.
* Use Redux-Saga to optimize performance by managing how side effects are handled, ensuring that actions and asynchronous processes are non-blocking and handled efficiently.
* Using Cypress to test individual React components in isolation to verify their functionality.
* Integrate Lambda functions with microservices to offload specific tasks or computations, handle background processing, or respond to events generated by other services.
* Making asynchronous HTTP requests to microservices or AWS Lambda endpoints to fetch data required for rendering the UI components.
* React, microservices, and AWS Lambda, development teams can build scalable, resilient, and maintainable applications that meet the requirements of modern web development.
* Designed new, modular, Ul components to be used/documented in our interactive pattern library using Pattern Lab & React Storybook
* Developed responsive client-side application using JavaScript, React, Node. js, and Bootstrap, Material Ui.
* Writing and maintaining end-to-end tests using Cypress to ensure React components work as expected.
* Use CSS Grid or Flexbox for layouts when creating responsive react dashboards.
* Understand each module and gathered and managed the project content by using Content Management System (CMS).
* Integrating Cypress tests into the CI/CD pipeline to ensure tests are run automatically on code changes.
* Develop and maintain the user interface components using React and Next.js.

**CLIENT: FORD, Dearborn, MI DEC-2018 -MAY 2019**

**ROLE: UI DEVELOPER**

**Responsibilities:**

**Project Description**: SYNC is an integrated in-vehicle communications and entertainment system embedded in Ford and Lincoln vehicles. SYNC handles all of the commands sent from an App Link application and delivers the intended message inside the vehicle. All UI including commands, display requests, vehicle data authorization, and responses are managed by the SYNC system on behalf of the mobile application and communicated over App Link.

**Key Result Areas:**

* Developed high-performance web applications using JavaScript, HTML5, CSS, **React Js**, and **Redux**.
* Developed the application logos and images by using **Adobe Photoshop CS3** and Illustrator.
* Built a high-performance, reusable code library for UI and backend components.
* Involved in dealing with CSS and SCSS animations, Backgrounds, Layouts, Positioning, Text, Border, and Padding.
* Ensure sensitive data such as user information is encrypted and securely transmitted between React apps and Azure services (e.g., using HTTPS, OAuth, and token-based authentication).
* Utilize the HTML5 Canvas API to draw graphics and create animations within React components.
* Involved in reworking on single-page applications using **React, Redux, React-Router, Bootstrap**.
* Extensively used **Git**for version controlling and regularly pushed the code to **GitHub.**
* Working with ECMA Script 6 features. For build of JSX and ES2015(ES6) used Babel, webpack.
* Experienced in working with NodeJS and NPM modules.
* Implement frontend logic and behavior using ES6 modules, classes, and functions.
* Utilize new ES6 features such as arrow functions, template literals, destructuring, and spread/rest operators.
* Debug and troubleshoot ES6 code to identify and fix issues.
* Work with frontend frameworks/libraries like React while taking advantage of ES6 capabilities.
* Implemented **React container** and **presentational components** (as Stateless and Functional components when applicable).
* Architect and maintain **highly customized Vite configurations** for scalable, high-performance frontend applications.
* Optimize build performance using advanced techniques (e.g., **pre-bundling**, **manual chunking**, **dynamic imports**, **rollup options**).
* Fine-tune **production build strategies** to reduce bundle size, improve load times, and support multiple environments.
* My primary responsibility is to build user interfaces that are functional, visually appealing, and user-friendly. I need to work with other developers, designers, and stakeholders to ensure that the interfaces meet the needs of the users.
* React/Redux is widely used in building interactive and scalable user interfaces for web applications.
* Implemented routing for all the components developed. created and injected reducers in store and worked on Third party Library.
* Develop web applications using Angular framework.
* Implement responsive and dynamic user interfaces.
* Integrate with backend services and APIs using HTTP services.
* Handle data retrieval and manipulation.
* Next.js should be skilled in React, JavaScript, CSS, and HTML, with experience in building and optimizing web applications using Next.js.
* The Front-end developer will be responsible for designing and implementing front-end components that are optimized for performance, accessibility, and usability.
* In Agile Development Team is responsible for designing, developing, testing, and delivering the product. They are self-organizing and cross-functional, meaning they work together to complete tasks and achieve the sprint goals. The team is responsible for estimating the effort needed to complete the tasks and determining the best approach to solve the problem.
* Used **Jenkins** and **Docker** to run automation tests and deploy them to Development.
* Involved in the development of programming user interface design and front-end (HTML) for the web application using HTML, **Typescript** & **CSS** to the requirements of the client.

**CLIENT: IBM,** **San Jose, CA APR 2016- DEC-2018**

**ROLE: SR. UI/ REACT DEVELOPER**

**Responsibilities:**

**Project Description:**The IBM Quantum Experience launched in May, giving researchers, students, and enthusiasts cloud-based access to an experimental quantum computing platform. As computing moves beyond the classic binary system, bits with infinite positions will unlock infinite possibilities. Platform users can learn quantum principles, experiment with new algorithms, and more.

**Key Result Areas:**

* Created web pages using HTML5, CSS3, JavaScript, jQuery, Ajax.
* Involved in complete SDLC life cycle – Designing, Coding, Testing, Debugging, and Production Support.
* Created and maintained the framework and layout of each portal with Cascading Style Sheets (CSS) and SASS, Webpack build in tools.
* Implement automatic scaling for the React application using **Azure App Service** scaling features to handle increased traffic or load, ensuring high availability.
* Experience with Node.JS, Bootstrap JS. Responsible for developing the partner Homepages using HTML5, CSS3, JavaScript, React JS, **Node** **JS,** and jQuery, SCSS.
* Used Object Oriented Programming concepts to develop UI components that could be reused across the Web Application.
* Set up modern frontend testing environments using **Vitest**, **Cypress**, or **Playwright**.
* Implement snapshot testing, unit testing, and end-to-end testing workflows in the Vite ecosystem.
* Ensure test performance and parallelization in large-scale applications.
* Use **Vite with SSR frameworks** like **Vite + React + Express**, or tools like **Vike** or **Remix**.
* Implement **static site generation (SSG)** if SEO is a priority.
* Configure routing strategies accordingly (e.g., client-side vs server-side rendering).
* Used **Grunt** as a task runner to test the **JavaScript** with **JS hint.**
* Experience of working on CSS Preprocessors like **LESS and SASS** along with JavaScript task runners like **Grunt JS**.
* Deploy React applications globally using Azure's multiple regions, ensuring minimal latency for users across different geographies.
* Ensure that UI components are consistent with the design guidelines and branding of the application.
* Ensure that MUI components are optimized for performance by utilizing techniques like lazy loading and memorization.
* Optimize the performance of React apps with techniques like lazy loading, code splitting, and caching strategies, leveraging Azure’s static hosting and edge distribution.
* Use pre-built MUI components (e.g., buttons, dialogs, menus) to build the user interface efficiently.
* Experienced in using the NPM module for managing UI dependency.
* Built Web pages that are more user-interactive using JQUERY plugins for Drag and Drop, JavaScript, Bootstrap.
* Combine multiple MUI components to build complex UI elements, like navigation bars or form layouts.
* Experience in Node.js and Java script development.
* Write unit, integration, and end-to-end tests using tools like Jest, Mocha, Jasmine, etc.
* Works in using **React.js** components, Forms, Events, Keys, Router, Animations, and Flux concept.
* Worked on **React.JS** Virtual Dom and React views, rendering using components which contains additional components called custom HTML tags.
* Use MUI components (e.g., <TextField>, <Select>) for form controls while integrating them with React state management libraries like Redux or Formik for handling state and form validation.
* As a React developer working with Docker containers, your main responsibilities are centered around ensuring that your application is deployed securely and efficiently using containerization technology.
* Developed the flux pattern by using the **redux** framework as a core dependency.
* Used **React Router** for routing the redux app to connect redux and react to support routes.

**CLIENT:** BENCH TECH SOFTWARE PVT LTD**, BENGALURU, INDIA JAN-2013 – DEC-2015**

**ROLE: UI DEVELOPER**

**Responsibilities:**

**Project Description:**Ben Tech offers full-service technology support from the switch to wire, wire to Wi-Fi, server to desktop, cloud to the employee, employee to productivity. Benjamin Keith brings 20+ years of technical knowledge, professionalism, and business experience to you.

**Key Result Areas:**

* Used HTML5, and CSS, JavaScript for front end web designing.
* Used the functionalities to write code in HTML5/HTML, CSS3/CSS, JavaScript, jQuery, JSON, and Bootstrap.
* Involved in dealing with CSS animations, Backgrounds, Layouts, Positioning, Text, Border, and Padding.
* Deployment of web, enterprise Java components, messaging components and multi-threading.
* Implemented various Validation Controls for form validation and implemented custom validation controls with JavaScript validation controls.
* Performed Unit testing using **Jasmine and Karma**.
* Analyse system requirements and break them down into classes, objects, methods, and attributes.
* Identify key entities in the system and model them using object-oriented concepts like inheritance, polymorphism, encapsulation, and abstraction.
* Collaborate with stakeholders to ensure that system design meets business requirements.
* Apply design patterns (e.g., Singleton, Factory, Observer, Adapter) to solve common design problems.
* Implement proper inheritance and interface design to create polymorphic behavior across the system, promoting code reusability and flexibility.

**2012 BTECH IT Scient Institute of Technology**

**2017 MS CSE Chicago state University**

**Portfolio Link: https://madhuresume.netlify.app/**