Indra

* Having around 8+ Years of experience as a front end and full-stack development using MEAN (MongoDB, ExpressJS, AngularJS, NodeJS) & MERN (MongoDB, ExpressJS, ReactJS, and NodeJS) stack.
* Experience with **ReactJS** SPA using React-Router, **state**, **props**, events, keys, component lifecycle, DOM manipulation, event handling, forms and third-party react components. Used JSX with ES6 to define components.
* Experience with React Hooks (**useState**, **useEffect**, **useRef**, **useSelector**, **useDispatch** ,**useMemo**)
* Strong understanding of JavaScript (ES5/ES6/ES7/ES8) – **Closures**, **Scope**, **Promises**, Modules, Prototypes, Arrow functions, **functional programming**, classes, symbols, generators, **async await**, and Patterns.
* Built applications using React JS with ES6 which allowed us to render pages on both the client and server using the same codebase, developed web and mobile applications using ES6 features and frameworks such as React and Redux.
* Created SPA by configuring React-Router. Used **BrowserRouter**, **Route**, **Link** for Web.
* Worked with **Promises**, **Fetch API**, **Axios**, and async and await for making HTTP calls. Worked with many redux middlewares and third-party react components.
* Experience with ReactJS new features like Error boundaries, fragments, and new life cycle methods.
* Used **Redux** for state management using Store, Actions, and Reducers. Used pure functions to create reducers and bind Action Creators to dispatch actions.
* Experience in creating **REST API** using Express JS and using MongoDB for persistence and Redis for caching. Used **Mongoose** Schema/Model to create **CRUD** with Express.js controllers.
* Experience with Responsive and Adaptive Design using Media Queries, Twitter **Bootstrap**, and JQuery Mobile.
* Extensively used Node\_modules - **Lodash**, async, request, body-parser, fs, events, promises, **mongoose**, Redis, and socket.io to implement the API.
* Used $http, $location, $q, $log, $scope, $window services for consuming RESTful Web Services, logging and location operations. Used $root Scope, local and session storages, JWT token, $watch Collection and $apply.
* Extensive use of HTML5 APIs - Forms API, Location API, **Storage** API, Web Workers, Canvas, SVG, Video, Offline, and IndexedDB API.
* Used JavaScript data visualization framework (D3.js) to display information in a chart view.
* Experience in UI designing with **Bootstrap**, **CSS3**, CSS Grid Layout, font-awesome icons, SVG Sprites, **Google Analytics**, **Google Performance Metrics**, and **Lazy loading**.
* Experience with CSS3 preprocessors such as **SASS** (**Mixins**, functions, and variables) to create and reusable style sheets and styled-components to style react applications.
* Experience with **Babel**, **Webpack**, and Grunt to perform Transpiling, bundling, minification, compilation, and linting.
* Used **Jest**, Jasmine, Mocha, Chai for Testing.
* Worked in CI/CD process with **Git**, Gerrit, Jenkins, Docker, and Google Cloud.

**Education**

* BTech (CSE) - Pondicherry University, India. 2003

**Projects**

**Client: Move Inc, Santa Clara, CA Apr 2021- Present**

**Position: Full Stack Developer**

**Environment:** **React.js, Redux, React-Router, ES5, ES6, ES7, ES8, Lodash, Express.js, Node.js, MongoDB, Mongoose, Redis, Webpack, Babel, Gerrit, Git, Jenkins, Google Cloud, Docker, NGINX, HTML5, CSS3, Bootstrap.**

**Project Description:** The Move Inc network provides resources and tools for consumers and real estate professionals to help navigate all stages of the home-buying cycle. These tools give consumers access to the most comprehensive selection of existing homes for sale, property records, rentals, mortgage resources, senior housing options, moving resources, and more. The Move network has the deepest consumer engagement among all real estate websites - consumers spend more than 700 million minutes per month on Move websites and mobile experiences.

**Responsibilities**

* Complete exposure with MERN (MongoDB, ExpressJS, ReactJS, and NodeJS) stack with ES6/ES7/ES8.
* Created reusable components with ReactJS. Used JSX for markup, props, and state in the components, life cycle methods, DOM event handling, and created forms.
* Involved designing in web pages usingHTML5, CSS3, **JavaScript**, Bootstrap, SASS,LESS, React. js,**Redux**, Flux, MongoDB.
* Extensively used ES6/ES7/ES8 – block scoping, classes, **arrow functions**, **Rest** and **spread**, generators, modules, **promises**, destructuring, **template literals**, async, and await.
* Extensively used Functional Programming - **Pure functions**, Function composition, **Immutable** data, **Currying**, Partial Applications, and declarative programming. Used Lodash-FP and redux.
* Used React Hooks (**useState**, **useEffect**, **useRef**, **useSelector**, **useDispatch** ,**useMemo**) to create components .
* Created reusable components with React.js.Worked with **props**, **state**, **events**, **keys**, and component lifecycle. Used JSX with ES6 to define components.
* Used Redux middlewares - redux-form, redux-logger, **react-redux**, **redux-thunk**, redux-local storage.
* Used third party react components - react-video, react-select, react-tags input, react-datagrid, react-panels, react-calendar.
* Used React-Router for routing – Used **BrowserRouter** for Web.
* Used **Axios**, a Promise-based HTTP client to send an asynchronous HTTP request to REST endpoints and perform **CRUD** operations.
* Hands-on experience with React Native and POC on Jest for native apps development.
* Used **Express**.js with **MongoDB** for persistence and Redis for caching.
* Built out restful endpoints and lightweight aggregation layers using Node.js Framework such as Express.js, Body parser, cluster, process, Winston, Redis, JWT, passport.js, mongoose, Lodash, async, etc.
* Extensively used **Promises** and EventEmitter in Node.js to avoid callbacks and more readable code.
* Used HTTPS and NGINX web server as a reverse proxy for Node.js application. Used PM2 for deployment.
* Used styled-components to style React applications.
* Used Webpack with ESLint for code cleaning and also for bundling, module injection, and dependencies.
* Worked in CI/CD process with Git, Gerrit, Jenkins, Docker, and Google Cloud.

|  |
| --- |
| **Client: American Express, Palo Alto, CA Sep 2019-Mar 2021**  **Position: Full Stack Developer**  **Environment: React.js, Redux, React-Router, ES5, ES6, ES7, ES8, Lodash, Express.js, Node.js, MongoDB, Mongoose, Redis, Webpack, Babel, Gerrit, Git, Jenkins, Google Cloud, Docker, NGINX, HTML5, CSS3, Bootstrap.** |

**Project Description:** Matrix is a project for a smart monitoring system, in which they have case management, KRI’s (Key Risk Indicators), and governance reporting. Key risk indicators (KRIs) are an important tool within risk management and are used to enhance the monitoring and mitigation of risks and facilitate risk reporting. Operational risk is defined as the risk of loss resulting from inadequate or failed internal processes, people and systems, or external events. Operational KRIs are measures that enable risk managers to identify potential losses before they happen. The metrics act as indicators of changes in the risk profile of a firm.

**Responsibilities**

* Complete exposure with **MERN** (**MongoDB**, **Express JS, ReactJS, NodeJS**) full stack programming.
* Experience with **ReactJS** new features like **Error boundaries**, **fragments,** and **new lifecycle methods.**
* Worked with React - props, state, keys, refs, events and fetch API to make calls to the server.
* Used **Redux** for state management. Created reducers with pure functions and actions to dispatch events and stores to maintain the state.
* Worked with **React-Router for routing –** Used BrowserRouter for Web and NativeRouter for mobile.
* Worked with **Redux tools** - react-redux, react-router-redux, redux-thunk, redux-logger and redux-form.
* Used third party react components – react-video, react-typeahead, react-tagsinput, react-datagrid, react-calendar, react-tabs and react-autocomplete.
* Worked with **Javascript** **(ES5/ES6/ES7)** - closures, hoisting, prototypes, block scoping, rest, spread, destructuring, classes, promises, iterators, generators, async, await, maps, and sets.
* Used **functional programming** – pure functions, immutable objects, and declarative programming. Used lodash-FP and redux.
* Wrote **Express.js** and used **MongoDB** for persistence and Redis for caching.
* Used **Mongoose** Schema/Model to create the query, update and delete with Express.js controllers.
* Extensively used Promises and EventEmitter in **Node.js** to avoid callbacks and more readable code.
* Used **ZMQ** distributed in-memory Message Queue to connect different Node.js utilities and applications
* Used **Babel** for transpilation and **Webpack** for bundling, module injection, and dependencies. Used NPM scripts for build definitions.
* Experience with **Grunt** to perform **minification**, compilation, and linting and **NPM** to maintain package dependencies.
* Worked in **CI/CD** process with **Git, Gerrit, Jenkins, and Google Cloud.**

**Client: Sears, Chicago, IL. Sep 2017- Aug 2019 Position: UI Developer**

**Environment: HTML5, CSS3, SASS, DOM, Bootstrap, JavaScript, AngularJS, Angular UI, AJAX, Restful API, JSON, Git, Grunt.**

**Project Description:** Sears is one of the famous retail chains operating in many countries and has a large number of users. Buy Online is an online e-commerce shopping web application developed for this chain that allows online shopping features to its users. Using this website, the customers can log in as a guest, view items-their description, price, etc., add/remove items to their cart and purchase the item using various payment methods like credit card/debit card/PayPal, etc.

**Responsibilities**

* Used AngularJS as a framework to create a Single Page Application (SPA) which can bind data to specific views, synchronize data with the server, and two-way binding.
* Injected AngularJS $http, $q as the promise library for asynchronous data operation.
* Good with scope isolation, transclude, templating, in creating (elements, attributes, classes) using custom directives.
* Developed AngularJS Custom Services/Factory for reusable functionality, Custom Directives to provide shared components.
* Experience in developing web service applications using SOAP and REST APIS.
* Improved dependency injection for modular development and component isolation using AngularJS.
* Used link function to define directive’s API and attached event listeners to DOM elements, watching model properties for change, and updating the DOM.
* Used $http, $location, $log, $q, $timeout, $window services using dependency injection, logging and location operations.
* Implemented HTML5 features like semantics, local storage, data list, required attributes, and sliders.
* Created various Mixins, Variables, mixins, nesting, import, Custom Fonts, and Directives in SASS.
* Implemented CSS3 features like CSS3 Media Queries, Pseudo-classes, multi-column layout, gradients, and web fonts.
* Used CSS Sprites to decrease the number of HTTP requests and load time of web pages to improve page performance.
* Used AJAX, JSON, and XML to send a request to the server to check the functionality of the website.
* Experience with Grunt to perform minification, compilation, and linting.
* Worked with the source version control tool Git to manage the code repository.

**Client: Sherman Health Systems Inc - Elgin, IL Feb 2016 – Aug 2017 Position: UI Developer**

**Environment: DHTML, HTML5, CSS3, XHTML, Linux, jQuery, AJAX and JavaScript.**

**Project Description:** Sherman provides healthcare services to improve service delivery and patient outcomes. Sherman works with leading healthcare insurers to better coordinate care, reduce healthcare costs, and improve long-term health. The online portal allows members and brokers to look for information about health care plans. Members can enroll in an insurance plan using an online portal. It has a payment module, report module, etc. This web portal allows members to view the claims, change Doctors, and add new services.

**Responsibilities:**

* **Responsible for creating efficient design and developing User Interaction screens using HTML5, CSS3, JavaScript, jQuery AJAX and JSON.**
* Analyzed requirements and designed the task flow using flow charts and designed flow between pages of the UI.
* Designed and developed code using HTML/HTML5, XHTML, CSS/CSS3, DOM and DHTML.
* Utilized JavaScript and JQuery to improve the overall design and to update a portion of a GUI page thus reducing bandwidth usage and load time for web pages to get user input and requests.
* Worked in the development and adherence of User Interface look/feel, schemes, and layouts.
* Prepared and defined required tasks, estimating durations for tasks, sizing team for allocation effort for all phases of SDLC, deliverables, and other digital activities.
* Responsible for ongoing technical maintenance and support for assigned applications.
* Analyzed, debugged, and revised existing system logic difficulties and documentation as necessary.
* Designed and created standards-compliant web pages that are interactive, attractive, and intuitive.
* Created HTML pages from mockups and visual wireframes, working closely with the design/product team.
* Used CSS3 Sprites to decrease the number of HTTP requests and increase the load time of web pages.
* Produced semantically appropriate, standards-compliant markup, and produced rich user interfaces.
* Used JavaScript and jQuery for validating the input given to the user interface.
* Created web content and tested all the websites with multiple browsers, operating systems, and screen resolutions.
* Developed and maintained the layouts using CSS2/CSS3 for all intranet applications compliant with W3C.
* Performed validation of completed sites, which included debugging and testing code.
* Tested cross-browser compatibility to ensure the look and feel of the UI designs worked the same in various versions of different browsers (IE, Firefox, Google Chrome, Safari, and Opera).

**Client: AT&T - Dallas, TX Feb 2014 – Jan 2016**

**Position: UI Developer**

**Environment: HTML5, CSS3, XHTML, JavaScript, jQuery, AJAX, JSON, Java,**

**Project Description:** ATT is one of the largest telecom service providers in the U.S. ATT Retail Line of Business provides the End User with all Classical and Next Generation communication devices (Wireless & Wireline) like iPhone, DTV, High-Speed Internet Access using a Sales Front End called OPUS.

The web-based application controls the Provisioning aspects and is primarily used in Cingular wireless Retail stores. Was involved in the Service Provisioning Domain analysis and testing ensuring the system to be flawless.

**Responsibilities:**

* **Created UI documentation and added HTML/DHTML help to the interface.**
* Developed HTML, CSS based page layouts using the prototypes, wireframes, and images for vertical re-design of the websites.
* Customized the HTML search page and designed pagination.
* Used jQuery and JavaScript to manipulate DOM, registering callbacks, or watching the model changes.
* Used jQuery UI controls as Accordion, Tabs, Dialog, and used jQuery mobile data attributes to create mobile-friendly pages.
* Designed and implemented a dynamic image rendering web service that is consumed by JSP, jQuery, JavaScript.
* Especially working with AJAX data consisting of JSON/XML/JSONP responses, parsing the same, and presenting on the UI layer.
* Worked on Cross-Browser compatibility and fixed the bugs for several browsers.
* Leveraged knowledge of W3C standards, cross-browser development techniques.

**Technical Skills:**

**Web Development :** HTML5, CSS3, Grid Layout, Bootstrap, SASS, ES5, ES6,

ES7, ES8, jQu

ery, JSON, AJAX, XML, AngularJS, MVC,

DOM.

**Databases :** MySQL, MongoDB.

**Programming Languages :** ES5, ES6, ES7, ES8.

**IDEs worked on :** WebStorm, Sublime, Visual Studio, Brackets.

**Application Servers :** Apache, Tomcat, NodeJS.

**Node Libraries :** Lodash, Async, JWT, ExpressJS, Body-Parser, ExpressJS, Underscore JS, Mongoose.

**Front End Libraries :** Modernizr, Angular-UI, jQuery,ReactJS,Redux,Jest,Jasmine.

**Version Control :** GIT, GitHub.