**RISHABH MEWAR**

**+91-7768086761| rishabh.mewar.academic@gmail.com | https:[//www.](http://www.linkedin.com/in/rishabh-mewar-4646969a)linkedin.com/in/rishabhmewar**

**SKILLS🡪 UI ARCHITECTURE:**

* **Web Frameworks:** Angular2+ & React.js.
* **CSS Frameworks:** Foundation, Bootstrap.js, Semantic UI, Materialize & Skeleton.
* **CSS Preprocessors:** Stylus, Saas, Less.js & Post CSS.
* **Data Interchange Format Language:** JSON & XML.
* **Application Architecture:** Cross origin resource sharing (CORS) & Responsive design.
* **State Management Libraries:** Flux, Redux & RXJS.
* **Data Visualization Libraries:** n3-Charts, D3.js (Data driven document) & HighCharts.
* **Persistent Immutable Libraries:** Immutable.js, Lodash, Rambda & Modernizr.
* **Package Managers Tools:** Yeoman, Bower, NPM & Yarn.
* **Task Runner Tools:** Grunt & Gulp.
* **Module Bundlers Tools:** Web pack & Browserify.
* **JavaScript Templating Engines:** Underscore.js, Mustache.js & Handlebars.
* **Testing Frameworks:** Cucumber, Mocha, and Karma & Jasmine.
* **Documentation Software:** Swagger.
* **Linting Frameworks:** ESLint, JSLint, TSLint & Prettier.
* **Debugging Frameworks:** Chrome Developer Tools.
* **Version Control:** GitHub, GIT & SVN.
* **Cross Browser Compatibility Tools:** Cloud Testing, Cross Browser Testing & Testing Bot.
* **Security:** IAM Protocols🡪OAuth2.0 &OpenID Connect.

**SKILLS🡪 PROGRAMMING/DATABASE ARCHITECTURE:**

* **Programming Language:** JAVA.
* **Programming Paradigm:** Object Oriented & Reactive.
* **Programming Principles:** SOLID design principles, Data Structures & Algorithms, Time & Space complexity, Refactoring, Performance Tuning, Separation of Concerns, Avoid premature Optimization, Scaling, Multithreading, Concurrency and Synchronization.
* **Application Architecture:** Cross origin resource sharing (CORS), Design Patterns & Domain Driven RESTful Micro Services.
* **Java Frameworks:** J2EE, Spring Boot, Spring AOP, Spring JPA, Spring Batch, Spring Security, Spring Cloud.
* **JavaScript Framework:** Node.js
* **Container Orchestration:** Google Kubernetes & Docker Swarm.
* **Network Orchestration:** SDN
* **Message Broker:** Rabbit MQ.
* **Protocol Format Exchange/Data Interchange Format:** JSON & XML.
* **APIs Contract/ Design/Test:** Swagger & Postman.
* **DevOps Tools:** Jenkins.
* **Integration & Unit testing:** Arquillian, Junit5 & Cucumber.
* **Collaboration Tool:** Atlassian Tools
* **Source Control Tools:** GitHub, GIT & SVN.
* **Build Tools:** Maven.
* **SQL Database:** Oracle DB.
* **Server Side Development:** Express.js.

**EXPERIENCE**

**Senior Software Engineer, Capgemini Technology Services India Ltd., Bangalore Nov’17-Present**

1. **ECRD PROJECT🡪** An application to generate prices of repairs. Also, work in client location, GE, Whitefield, Bangalore.

* Implement Angular Components, Services, Content Projection, Forms, Lazy loading and Prefetching the data using Resolvers.
* Implement State Management Libraries: RxJS & Redux (Angular NgRX) for handling asynchronous requests.
* Integrating the UI with Node.js Restful Web Services and Perform Cross Browser Compatibility.
* Implement Reusable Angular Services Logging & Configuration Management.
* Utilize Karma & NPM to build and perform Unit Tests & Integrated Tests of Angular Components.
* Implement Security Access control using Open ID Connect Auth2 & JSON Web Tokens.
* Worked on CSS Frameworks, CSS Preprocessors, and Angular Materials & Prime Ng.

1. **ADDITIVE MANUFACTURING PROJECT🡪** It is a concept laser based software for printing 3D parts in aviation sector.

* Building React Components, Implementing React Hooks and Integrating Authentication with React Hooks.
* Implement Redux State Management and Securing the Application with Auth0.
* Implement testing with Jest and Client Side React Router 4 and Perform Cross Browser Compatibility
* Implement Persistence Immutable Libraries: Immutable.js, Lodash, Rambda & Modernizr.
* Creating Redux Middleware, Redux Thunk and Loading Data from the server with Redux Saga.
* Working with Asynchronous and Context APIs and Higher Order Components.

1. **MATERIALS PROJECT🡪** A supply chain application aims to generate P.O (purchase Orders) for clients.

* Architectural (Design), Technical (Test Cases) & Functional (Deployment) understanding of the Interfacing Modules.
* Implement Collections, Multithreading, Concurrency, Synchronization, Design Patterns, SOLID coding methodology using JAVA.
* Implement RESTful Domain Driven Design Micro-Services (Learn Architectural Design Patterns).
* Build Application’s Micro services using Spring Boot/Spring Architecture with Maven Architype.
* Contain, Monitor, Manage Load Balancing & Networking Data Stateful Applications using Docker.
* Creating, Finding, Configuring, Mapping Services using Circuit-breaker, Client side load balancing & Routing.
* Use Swagger to build APIs contract and Perform Junit5 testing and Integration testing using Arquillian.
* Used POSTMAN to test RESTful APIs for HTTP requests and Design the Project Flow in UML.

1. **SUPPORT APPLICATION PROJECT🡪** Supports a legacy application of GE Aviation.

* Analyze Business requirements and perform Database & Java Enhancements.
* Understand & Implement Modifications (CDI, Bean validation & Persistence) in J2EE application architecture.
* Implement Materialized Views and PL/SQL procedures according to the Modules (Business Domain).
* Create a Scheduled Chain of Jobs (PL/SQL procedures) that Migrate Data among the Applications.

**EDUCATION & INTERESTS**

* **B.Tech** in Electronics & Telecommunication- **CGPA:** **2.84/4 Narsee Monjee Institute of Management Studies, Mumbai |** 2017
* Senior Secondary School, Modern School, Rishikesh: **78%** **May’12**
* High School, Modern School, Rishikesh: **82%** **May’10**
* Read Articles/Theories (CFA) in Finance.