**Henry Chen**

**Summary**

* Over 5 years of experience as a Java full stack web Developer, creating **Single Page Web** applications using **Angular 2/4/5/6/7**, **Node.js,** and **React.js** with backend using Spring MVC, Spring Boot, Hibernate, and JPA in various domains.
* Proficient in developing Single Page Applications (SPA) using **Angular 2/4/5/6/7** and **React.js**.
* Developed SPA by using Angular related technologies such as **TypeScript**, Angular **Components, Directives**, **Pipes, Services**, **HTTP Client**, **Observables**/**Promise**, **Reactive/Template** driven forms, **Two-way data binding**, **Life Cycle Methods**, **Routing,** and **Routing Guards** within multiple components.
* Developed SPA by using React related technologies such as **JavaScript ES6**, **Redux**, **CSS Module**, **Dynamic Styling**, **Axios**, **React Components**, **Props**, **States**, **Two-way data binding**, **Promise**, **Conditional Expression**, **Life Cycle Methods**, **High Order Component**, **Props Validation**, **Routing,** and **Router Guard** within multiple components.
* Extensive experience in using front-end technologies like **HTML5**, **CSS3**, **Bootstrap**, **JavaScript**, jQuery, and **AJAX** for building dynamic, user-friendly, and high-performance web applications.
* Applied **responsive web design** to fit all users and devices from desktop to tablet and mobile phone by using **Grid Systems** and **Media queries.**
* Extensive experience in Backend Server-Side Programming using **J2EE technologies** like **Java Core**, **Spring Core**, **Spring MVC**, **Spring Boot**, **Spring AOP**, **XML**, **Hibernate**, **JPA**, **JDBC,** and some other technologies **Node.js**, **Express.js**, and **LAMP** (Linux, Apache, MySQL, PHP).
* Proficient in developing **RESTful API** using J2EE Technologies and testing **APIs** using **Postman**.
* Excellent in **Mobile development** with **Android SDK** and **Android Studio**.
* Expertise in database system like **SQL Server**, **Oracle**, MySQL, and NoSQL database **MongoDB.**
* Proficient in creating Dynamic SQL and performing CRUD and Other SQL operations using Relational Database like **SQL Server**.
* Expertise in Unit Testing by using **Karma**, **Mocha**, **Jasmine**, and **Jest** in frontend, while **JUnit** and **Mockito** in backend.
* Proficient in debugging and troubleshooting existing code using **JIRA**, Developer tools, and **GIT** as Version Control System.
* Involved in all stages of Software Development Life Cycle (**SDLC**) as well as working in **AGILE** Methodologies based development environment, participating in spring/iterations, and **SCRUM** meetings.
* Basic idea in cloud computing with **Amazon Web Services** (AWS).

**Skills**

|  |  |
| --- | --- |
| **Languages** | Java, Python, JavaScript, TypeScript, PHP, C#, C/C++, R, shell scripts, SQL |
| **Databases** | SQL Server, Oracle, MySQL, MongoDB, SQLite, Redis |
| **Web Technologies** | HTML, CSS, Bootstrap, JavaScript ES5/ES6, jQuery, AJAX, Angular 2/4/5/6/7, React.js, Redux, SASS, Gulp, Webpack, Socket.io |
| **Server-Side Technologies/ Frameworks** | J2EE, Servlet, JSP, Spring Core, Spring AOP, Spring MVC, Spring Boot, Spring Cloud, Hibernate, JPA, JDBC, Maven, XML, Node.js, Express.js, LAMP |
| **Data Science** | NLTK, Pandas, Numpy, Scikit-learn, Seaborn |
| **Mobile Development** | Android SDK, Android Studio, Gradle |
| **Tools and IDEs** | Eclipse, Visual Studio Code, Sublime Text 2/3, WebStorm, PhpStorm, Azure Data Studio/SQL Server Management Studio, Oracle SQL Developer, MySQL Workbench |
| **Version Control** | GIT, GitHub |
| **Debugging/Testing Tools** | Jasmine, Karma, Jest, Mocha, JUnit, Postman, Mockito |
| **Operating System** | MAC OS X, Ubuntu/Linux Mint, Windows |

**Experience**

**Tarte Cosmetics New York, NY**

**Full-Stack Developer (Angular 6/Spring MVC/Hibernate/SQL Server) June 2018 - Present**

***Description*:**

Tarte Cosmetics is a beauty products provider for small business. The internal application was aimed for Products Management by helping our internal customers to check product details based on different cosmetics brands. I was mainly involved in designing various modules by creating corresponding angular components in frontend, developing backend APIs using Spring MVC, and writing some algorithms for data parsing.

***Responsibilities***:

* Contributed to a team in developing a **single page web application** (SPA) based on the **RESTful API** using **Angular 6**, **Spring MVC**, **Hibernate,** and **SQL Server**.
* Designed multiple components such as **product dashboard**, **product filter**, **product list**, **product details**, and **user login/register** with **JWT authentication** and **authorization**.
* Used **Angular Routing** to navigate between different components. Involved **Angular Route Guard** for the **Authentication and Authorization** by passing logged users’ **Json Web Token (JWT)** into Browser **localStorage**.
* Used **Template Form** to validate the user’s input in Login and Register Components, while **React From** for handling complex scenarios.
* Utilized **Directives** in the application. For example, **structure directives** like **\*NgFor** to dynamically exhibit multiple data entries, **attribute directives** like **NgClass** to add and remove CSS classes on an HTML element, and **customer directives** which can be reused.
* Wrote **Customized Pipes** based on business Logics by implementing PipeTransform interface.
* Used **@input**, **@output**, **@Viewchild, local reference,** and **service** for Component Communication.
* Applied **Dependency Injection** and **service** with **@Injectable** decorator.
* Used **Http Client Module** to send http request including **GET**, **POST**, **PUT,** and **DELETE** to call Corresponding **Backend RESTful APIs** with **@RequestMapping** Annotations in Controller Packages.
* Used **Snapshots** to call the router initial data and **Observables (RxJS library)** to handle asynchronous data calls coming from backend REST APIs.
* Designed customized **Interceptors** by implementing HttpInterceptor interface to **add authorization header** with Json Web Token and **handle the global error**.
* Used **SpringMVC architecture** to build the backend project structure layers: **Domain**, **DAO**, **Service**, **Controller**, Exception, and Utils. Configured web.xml by adding spring MVC **DispatchServlet**. Configured Hibernate setting parameters for Database Connection in /src/main/resource/**application.properties** file.
* Utilized Object-Oriented Programming (OOP) design patterns such as **inheritance**, **polymorphism**, **Encapsulation** **interface**, **abstraction**, **overloading,** and **overriding**. Applied **Collections** such as Set, HashMap and List to store data and passed through layers.
* Used **Maven** to handle and update dependencies, run **maven build**, **maven clean install** commands to start, and refresh the backend applications.
* Designed **Object relational Mappings (ORM)** patterns (**Mapping each POJO class to database table**) by creating a corresponding **Hibernate Mapping file(hbm.xml)** and used Java-based **Annotations** inside POJO classes such as **@Entity**, **@Table**, **@Column**, **@Id**, **@GeneratedValue** and so on.
* Extensively applied other Hibernate and database management technologies like **Sessions**, **transactions**, **criteria queries**, Hibernate Query Language (**HQL**), **Native SQL**, **Cache**, **Batch Processing**, and **Interceptors**, to interact with the database.
* Used **@CrossOrigin Annotation** in Controller Layer to fix front-end and back-end different ports data communications problems.
* Developed some Spring customized **Dependency Injections** based on business logics.
* Used **Azure Data Studio** for **SQL Server** Database graphic interface. Implemented **CRUD** operations in DAO layer. Applied **T-SQL** for complex DB operations, wrote SQL **functions** and **stored procedures** for repeated operations. Optionally applied some Advanced T-SQL techs such as **Exception**, **Cursor** and **Trigger**.
* Used **Jasmine** and **Karma** to do front-end Angular Unit test, **JUnit** to do back-end API Unit test, **JIRA** to track the process of the project and joined the **SCRUM meeting** for report, **Slack** to communicate with team leader and members, **Git** and **GitHub** for Version Control and troubleshot with merge conflicts.

***Environment***: JavaScript, Typescript, HTML, CSS, Bootstrap, Angular 6, Git, Java Core, Spring MVC, Hibernate 5, SQL Server, Windows, and MAC OS X.

**Tompkins Mahopac Bank Brewster, NY**

**Full-Stack Developer (React/Spring Boot/Hibernate/SQL Server) May 2017- May 2018**

***Description*:**

Tompkins Mahopac Bank provides assorted banking services for customers such as saving/checking account, loans, mortgage, etc... The internal application and the loan repayment portal was designed to provide a centralized system for loan data management. I was responsible for various React Modules in frontend such as logging system, loan data dashboard, resource management system, and project task system. Applied Spring Boot and SQL Server database for backend APIs.

***Responsibilities***:

* Developed a **single page web application** (SPA) based on the **RESTful API** using **React** and **Redux**, **Spring Boot**, **Hibernate,** and **SQL Server**.
* Responsible for creating and maintaining reusable **stateful** and **stateless components**.
* Utilized **React Life Cycle Hooks** Methods such as ComponentWillMount(), ComponentDidMount(), ComponentDidUpdate(), and render() in some components.
* Called **setState()** method to update the stateful component global state and then re-render DOM. Used React **Props** for Parent and Child **Component Communication**.
* Used **Axios** to perform **CRUD** operations by sending http requests to call backend RESTful APIs. Utilized **Axios Interceptors** to handle global errors and add authentication header.
* Used **Redux** (**Action, Middleware, Reducer, Store**) for managing **Application Global State** and handling user **authentication**.
* Dispatched **multiple actions** in Action layer and tackled actions in Reducer layer based on **action types**.
* Applied **react-router**, **react-router-dom,** and **react-router-native** libraries to handle Routing among different components. Applied **conditional rendering** in **Router Guard** for **user Authentication and Authorization** by checking **Json Web Token (JWT)** in browser **localStorage**.
* Performed **form validations** by adding input fields and validation labels inside <form> element and populating onSubmit function to globally validate user input.
* Designed middleware by using **redux-thunk** to tackle asynchronous code.
* Applied **JavaScript ES6 features** like **const**, **let**, **arrow function**, **template literals** etc.
* Involved **CSS module** for every component and utilized **Radium** and **High Order Component (HOC)** for **CSS styling**.
* Used **@trendmicro/react-sidenav** library for Navigation bar.
* Broadly **Applied Spring Boot** to build the backend project structure: **Domain**, **DAO**, **Service**, and **Controller**.
* Used **Maven** as build Tool to write the dependencies and maintain the version of the dependencies.
* Implemented **POJO classes**, **JpaRepository interface**, **business logics**, and **back-end API functions for front-end http requests (Axios)** under Domain, Dao, Service, and Controller layer respectively.
* Followed Spring Data Respository (**JpaRepository**, **CrudRepository**) Principles to define customer functions in the DAO layer.
* Applied **@RequestMapping**, **@GetMapping**, **@PutMapping**, **@DeleteMapping** annotations in Controller layer to determine which url and method to be set.
* Defined **Customized Exception** and **utilities** classes.
* Run the back-end Application with **a class @SpringBootApplication Annotation** or using command **spring-boot:run** or **installing spring tool suites**.
* For frontend unit testing, wrote some **Jest** mocks **for react testing.** For backend unit testing, utilized **Postman** and **Mockmvc** to test REST controller, while **JUnit** assertEquals() and **Mockito** when() to test service layer. Applied **Agile** Environment and used **JIRA** as the product management tool for task process tracing and joined the **SCRUM** meeting for report. Used **Git** and **Github** for version control and troubleshot with merge conflicts.

***Environment***: JavaScript ES6, HTML, CSS, Bootstrap, React, Redux, Git, Java Core, Spring Boot, JPA, SQL Server, Windows, and MAC OS X.

**TriState Finance Hasbrouck Heights, NJ**

**Full-Stack Developer (JSP/Servlets/Hibernate/Spring MVC/Oracle) April 2016- April 2017**

***Description*:**

TriState Finance offers Auto Lending, Auto Refinancing, Automotive Sales, Car Financing, Car Loan, and Auto Loan. The internal application was about auditing sales and loan Non-performing asset information. I was mainly involved in designing JSP page modules in front-end like user/admin logging system, non-performing asset information management dashboard, auditing management, non-performing asset details, blacklist management, Overdue loan SMS collection etc. and used Spring MVC for backend.

***Responsibilities***:

* Designed a J2EE Web application with **Spring Core**, **Spring MVC** architecture, **JSP**, **Servlets**, **jQuery**, **Hibernate** and **Oracle**.
* Coded both front-end and back-end inside Eclipse under the created **Dynamic Web Project** structure by the project manager. Put All front-end JSP pages, JavaScript, CSS, and images under src/**WebRoot** directory
* Utilized **form submission** or **jQuery Ajax calls** to call the required APIs inside Controller layer.
* Developed back-end MVC architecture with **Domain**, **Dao**, **Service**, and **Controller** layers.
* Wrote **POJO classes**, hibernated **CRUD operations**, **business logics**, **REST API functions** for front-end http requests under Domain, Dao, Service, and Controller layer respectively.
* Applied **Servlet Life Cycle** Methods like **init(), service(), doGet(), doPost()** for Controller layers. Realized **page redirection** by using **request.getRequestDispatcher().forward()** with HttpServletRequest Class.
* Configured **Build Path** by adding required **external JARs** for specific development. Configured **hibernate.cfg.xml** for database connection settings with several properties inside <session-factory> tag.
* Realized **Object relational Mappings (ORM)** patterns (**Mapping each POJO class to database columns**) by creating a corresponding **Hibernate Mapping file (hbm.xml)** for each POJO class, or use **Annotations** inside POJO classes.
* Extensively applied other Hibernate and database management technologies like **Sessions**, **transactions**, **criteria queries**, Hibernate Query Language (**HQL**), **Native SQL**, **Cache**, **Batch Processing**, and **Interceptors** to interact with the database.
* Initialized **Spring beans** and called the bean xml files path by creating a new instance of **ClassPathXmlApplicationContext** class. Get the required bean with **getbean()** method under ApplicationContext class.
* Developed some Spring customized **Dependency Injections** based on business logics.
* Applied **PL/SQL** for complex DB operations and wrote SQL **stored procedures** for repeated operations.
* Applied **JUnit** tool for writing Unit /Integration test cases. Used **JIRA** to track and report the issues in the application. Applied **Git** for Version Control and code management.

***Environment***: HTML, CSS, Bootstrap, JSP, Servlets, Git, Java Core, Spring Core, Spring MVC, Tomcat 7.0, Hibernate 4, Windows 7, and Oracle 11g.

**National Camera Exchange Golden Valley, MN**

**Full-Stack Developer (React/Node.js/Express.js/MongoDB) March 2015 - March 2016**

***Description*:**

National Camera Exchange is a Photographic and Video retailer for small business. The internal application was aimed for Camera Equipment Management by helping our internal customers to check camera product details based on different brands. I was mainly involved in designing React Components in front-end such as the logging system, camera product dashboard, product list, product filter, and product details. Applied Node.js and MongoDB for backend APIs designation.

***Responsibilities*:**

* Designed a Single Web Application using **React**, **Redux**, **Bootstrap**, **CSS**, **Node.js**, **Express.js**, **MongoDB,** **mongoose**, and **Google API**.
* Utilized **passport** to Call Google APIs for authentication if the users selected to login with Google Accounts.
* Used **Redux** (**Action, Middleware, Reducer,** and **Store**) for managing Application **Global State**.
* Designed reusable and feasible component **containers** for inserting different components pages.
* Used React **Props** for Parent and Child Component Communication. Called **setState()** method to update the stateful component global state and the re-render DOM.
* Implemented **Express Routing** by calling get(), post(), put(), and delete().
* Handled libraries with **NPM** tools, initialized package.json file with npm Init, run the application with **NPM start** command by configuring the start scripts setting.
* Configured **MongoDB Connections** and implemented **CRUD operations** with mongoose.
* Utilized JavaScript Asynchronous Mechanism with **Promise** or **Asyn/Await** and handled exceptions with multiple **try catch blocks** for Asyn/Await usage.
* Applied **JavaScript ES6 features**, for example, used **let** and **const** for variable definition, and imported packages with required() method.

***Environment***:HTML, CSS, Bootstrap, React, Redux, Git, Node.js, Express.js, MongoDB, Visual Studio Code, and Ubuntu 14.04/Linux Mint 17.

**Eden Prairie Center Eden Prairie, MN**

**Mobile Developer (Android SDK / LAMP) February 2014- February 2015**

***Description*:**

Eden Prairie Center is a retail provider for Restaurants, Entertainment, and Shopping Center. The internal mobile application is to help internal customers to check living products and shopping.I was mainly involved in designing Android Activity modules in Android Studio like user/admin logging system, adding products, adding coupons, checking product details, checking coupon details, shopping cart, etc. and used LAMP (Linux, Apache, MySQL, PHP) for designing backend APIs.

***Responsibilities*:**

* Developed a Native Android APP by using **Android SDK** and **LAMP**.
* Responsible for Developing **Activity Modules** and **UI layout** based on project requirements. Binding UI layouts and Activity class with **setContentView()** method inside OnCreate() method. Mainly Designed UI layout with **LinearLayout** and **RelativeLayout**.
* Implemented **Activity Life Cycle** Methods OnCreate(), OnStart(), OnResume(), OnPause(), onStop(), OnDestroy(), and onRestart() in UI Components.
* Configured **AndroidManifest.xml** by adding multiple activities and uses-permission. Changed the launching activity by moving the <intent-filter> tag inside into different activities.
* Applied UI Controls inside UI layouts for designing different pages, like **TextView, EditText, Button, ImageView, CheckBox, RadioGroup/RadioButton, AutoCompleteTextView, Spinner** etc.
* Designed **customer Adapter** by extending **BasedAdapter** interface. Used **setAdapter()** method to bind the listView with the customer Adapter. Used **notifyDataSetChanged(**) method to tell the ListView that the items in the dataset have changed.
* Optionally applied **Services**, **Broadcast Receivers,** and **Content Provider** based on business logics. Wrote customized Animation, Color Style, Images, Menu, and Values inside **resource** package. Utilized **Handler** and **switch case blocks** to send and process different thread’s Message from MessageQueue.
* Use **Intent** interface for Page Routing and its internal methods like putExtra() and getExtra() for **Component communication** (**data transferring**).
* Developed **server-side APIs** using PHP inside Sublime Text/ PhpStorm and Called mysql\_connect() method for PHP MySQL connection.
* **Retrieved JSON data** by **calling server-side APIs** with **URLConnection** class in UI Components.

***Environment***:Java Core, Android, MySQL, PHP 5.6, Apache server, Android Studio, Sublime Text 2/PhpStorm, Navicat, Windows 7, and Ubuntu 14.04/Linux Mint 17.

**Education**

Master of Science in Computer Science, Stevens Institute of Technology