

# Chunhua Li

## SUMMARY

609-358-4185 – [chunhua.li.chen@gmail.com](mailto:chunhua.li.chen@gmail.com) – <https://github.com/jennyleelich/>

---

- Visit my personal website <https://d3jbwtno2fihos.cloudfront.net> to download the latest resume and view Portfolio.
- Over 15 years of experience in the IT industry with 10 years of extensive experience as a Front End UI Developer with solid understanding of designing and development of web application. With 5 years of extensive experience as AWS cloud solution provider and solid understanding of Modern Spring Boot + Angular Architecture.
- Strong Experience of **Angular** and **React JS** to interact with a web service to render the page from data returned from the service.
- Applied Angular to achieve front-end **MVC** in **single page application**, and **update angular version stably**.
- Worked on **MEAN Stack** utilizing **MongoDB, ExpressJS, Angular** and **NodeJS**.
- Worked on **Full Stack** utilizing Modern **Springboot, Angular, MySQL** and **AWS Cloud** deployment .
- Involved in designing the User Interface mainly focusing on easing access to registration, plans and features available on the website using **HTML, HTML5, CSS/CSS3/JavaScript/TypeScript, JSON, RequireJS, HandlebarsJS, AngularJS, PrimeNG, Kendo UI, Syncfusion Library, Angular Material, UnderscoreJS, Modernizr, JQuery** and **AJAX**.
- Strong knowledge in using Object Oriented Programming concepts in JavaScript and full working knowledge of **DOM** model.
- Experience in Section **508 Compliance**, Usability & **WCAG Accessibility** guidelines.
- Strong experience in Web **Accessibility** testing and Digital **Accessibility** solution.
- Experience in **HTML5 API** like Geolocation, Drag/Drop, Session/Local Storage.
- Developed **responsive pages** using twitter **Bootstrap** and **Media Queries**.
- Experience in **LESS** and **SASS** preprocesses to simplify CSS development and maintenance.
- Strong experience on AJAX techniques for manipulating and parsing multiple data types including JSON and XML.
- Experience of Consuming **RESTful** services to handle data for dynamic applications and working on developing applications based on MV\* frameworks for easy to maintain design.
- Experience of working with SQL and NOSQL databases like **MySQL, MongoDB**.
- Working knowledge of sever side **PHP, Java/J2EE, Spring Boot, .net** environments
- Experience in using **Maven, yarn, npm, Bower, gulp, and Grunt** for packages and project tasks management.
- Prepared architectural documents of user manual and functional specifications.
- Worked on all phase of Software Development Life Cycle( **SDLC** ) like Requirement Analysis, Implementation, Release and Maintenance and also extensive experience **agile & waterfall** project environment.
- Good knowledge of **Ionic3 PhoneGap** and **Jquery Mobile**.
- Extensive experience **improving performance** of web pages and making pages **cross platform compatible**.
- Experience working with **off shore team** and **clients** for fulfill their requirements. Also worked with QA team on daily basis for testing, loading and performance checking.
- **Technology lead** experience, worked parallel with QA team to fix the reported bugs and defects and ensure cross platform comparability. Adaptive to emerging applications and technologies. Quick learner and excellent communicator.

## TECHNICAL SKILLS

---

|                                 |  |
|---------------------------------|--|
| <b>Web Technologies:</b>        | HTML5, CSS3, XHTML, XML, JavaScript, TypeScript, JQUERY, JSON, AJAX  |
| <b>IDE:</b>                     | Notepad++, Eclipse IDE, Dreamweaver, Sublime Text, Bracket, WebStorm, TextPad, Visual studio code  |
| <b>Libraries and Framework:</b> | ReactJS, Angular, Angular Material, Django, Kendo UI, Syncfusion UI, PrimeNG , jQuery, Node js, Bootstrap, Handlebar js, UnderscoreJS, Modernizr, Spring Boot  |
| <b>Database:</b>                | MySQL, MongoDB   |
| <b>Server Environment:</b>      | Java/J2EE, SpringBoot, JSP, PHP, RESTful, Python   |
| <b>Control/Testing Tool:</b>    | GIT, SVN, Firebug, Chrome Developer Tool   |
| <b>Browsers used:</b>           | Google Chrome, Mozilla Firefox, Safari, Opera, IE  |
| <b>Sample link</b>              | <a href="https://d3jbwtno2fihos.cloudfront.net/">https://d3jbwtno2fihos.cloudfront.net/</a> (my personal website, including resume and portfolio gallery)<br><a href="http://jennyshop.s3-website.us-east-2.amazonaws.com">http://jennyshop.s3-website.us-east-2.amazonaws.com</a> (springboot3 + angular16+ AWS full stack project)<br><a href="https://dq5xwix87sylo.cloudfront.net">https://dq5xwix87sylo.cloudfront.net</a> (angular16 Lazy loading nested Routing and nested Form Demo)<br><a href="https://jennyleelich.github.io">https://jennyleelich.github.io</a> (Bootstrap + AngularJS family website) |
| <b>Sample code</b>              | <a href="https://github.com/jennyleelich/my_portfolio">https://github.com/jennyleelich/my_portfolio</a><br><a href="https://github.com/jennyleelich/spring-angular-sample-project#readme">https://github.com/jennyleelich/spring-angular-sample-project#readme</a><br><a href="https://github.com/jennyleelich/new_dashboard">https://github.com/jennyleelich/new_dashboard</a><br><a href="https://github.com/jennyleelich/jennyleelich.github.io">https://github.com/jennyleelich/jennyleelich.github.io</a>   |

## EXPERIENCE

---

### ANGULAR DEVELOPER

May 2018– now

ETS – Princeton NJ

Involved in ETS IBIS® Application - a state-of-the-art system for the collaborative development—from creation through finalization—of assessment components, including tasks, Forms, Packages, scoring keys, and more. The IBIS Application provides industry-standard connections for use by other assessment technology systems that perform functions such as test delivery, psychometric analysis, and scoring. And also involved in several sub project of IBIS:

- The IBIS® Content Review Tool (CRT) allows Facilitators, Reviewers, and Observers to review Items and associated information in a collaborative environment. It is used to manage and conduct internal and external Item reviews. The CRT makes Item content, metadata, Item statistics, and Item scoring information available for review. It has Session Management, Review and Evaluation, and Decision Management Interfaces.
- The IBIS® Test Assembly Assistant (TAA) supports Test Assemblers in creating Test forms. TAA is comprised of the Manage Configurations, Manage Sessions, and Build Assembly interfaces. The Test Assembly Assistant (TAA) Build Assembly Interface allows users to build Assembly Segments and review them with rules, charts, and graphs. Users can populate Assembly Units (AU) in the IBIS® Application with Items from the segments.
- The IBIS® Open Canvas authoring interface is a basic authoring interface to support efficient content entry and review tasks. Open Canvas is a streamlined interface that supports the most frequently used authoring options for the most commonly used Item Types. The Open Canvas authoring interface allows you to style an Item's content as necessary for delivery and scoring, and provides a real-time generic preview of the content.

### Responsibilities:

- Generate and set up angular project basic architecture, create code base and generate project routing modules and routing path, create module, component, service structure.
- Help team member setting up local development and git CI/CD deployment environment.
- Use Maven/Git/NVM source and version control tools to manage source code, task assignment, document, and other artifacts that placing under version control and sharing with team.

- Deep understand of the concept of the three work areas of the Git project: the Git repository, the working directory, and the staging area. Familiar with the git branch operation, such as git **rebase**, git fetch and merge, git **stash and pop**, git revert etc.
- Config and setup **Sonarqube** in angular project, auto collects and analyzes source code, and provides reports for the code **quality control**.
- Trouble shoot stage and product release issue and CI/CD deployment environment incompatible issue.
- Work with Engineer team for **AWS cloud( S3, cloudfront, API gateway, Lambda function, Beanstalk)** deployment process.
- Use **Kendo UI** and Angular **Material** as a collection and customized of UI components for Angular, such as DropDown, MultiSelect, RadioButton, Table, Paginator, Carousel, Chart, TabView, Accordion, Upload, Messages Module etc.
- Develop and customized **accessible** angular **material** component and lead the ally design and testing.
- Utilized device specific media queries and using Bootstrap for **responsive** layouts.
- Worked through **cross browser compatibility** issues with layout and styles and fixed major bugs for supported browsers and platforms.
- Cooperate with BA and UX team to elaborate the project design and logic , provide technical aspect for more user friendly experience.
- Develop customized form **validators, pipes, directives**. Develop our common Kendo grid component, **accessible** material grid component, accessible material dialog component etc.
- Involve in REST API Request and Response Data structure design, and integrate and test REST API.
- Set up nested and multiple layers **Data Model** structure and **nested Form** structure for UI development foundation.
- Update angular project version, experience **angular 4 to 15 version update**.
- Involve in **Python Django** project, fix and improve item card configuration and improve the project user accessibility experience.
- Create App **Router Module**(including forRoot and forChild Router) and **Router-Outlet** view and associated components, set up **router-link** directive, using **routerLinkActive** directive toggles css active class based on the current RouteState, using **loadChildren** to implement **Lazy Load**.
- Use **Router** navigate to different view and **ActivateRoute** to subscribe router queryParams and detect router change in Angular.
- Tap into **Component Life cycle** and perform operations when required. For example: ngOnChanges, ngOnInit, ngOnDestroy, ngAfterViewInit etc.
- Use **@output** decorator emit the event by using **EventEmitter** API, when changes from child component need to be reflected to its parent component in hierarchy. Use **@input** data binding to pass the external data into component/directive.
- Help with other cross projects trouble shoot the bugs and related issue.
- Use **service** injected into different components to share common behavior and variable
- Use **shared Module** declarations common component and exports common components, sharing common components across different modules.
- Use **HttpClientModule** to initiate HTTP request and process responses and handle error within the application, we can directly access the JSON response by subscribing to the Observable(response result automatic convert from JSON to anonymous Object), tell HttpClient the type of the response to make consuming the output easier and more obvious.

Citi Group – NJ

Involved in Bank internal Credit Finance and Risk Technology projects:

- Alert Dashboard Project, in this iteration implemented Alert and Notifications management, as a global end-to-end integrated control platform, the system could support Internal Audit, Compliance, Operational Risk and O & T Risk and Controls. User have email notification which will show a list of all the alerts that have been triggered. Alerts can be accessed on summary dashboard and users can track the progress of action on the alerts.

- Ethics Office Case Management System, provide database, work flow management and reporting capabilities. All elements of the new platform need to have entitlements-based access which is defined by geography, business and case type. The platform is able to store and report all concerns received by the Ethics Office and all required documentation on the handling of those concerns.

### Responsibilities:

- Use Rational Team Concert ( **RTC** ) source control to manage source code, task assignment, document, and other artifacts that placing under version control and sharing with team.
- Use **Git BitBucket** for version control, deep understand of the concept of the three work areas of the Git project: the Git repository, the working directory, and the staging area. Familiar with the git branch operation, such as git **rebase**, git fetch and merge, git **stash and pop**, git revert etc.
- Set up **TSLint** and **Angular v5 Snippets** extensions in **Visual Studio Code** for development environment.
- Use **PrimeNG** as a collection of UI components for Angular, such as DropDown, MultiSelect, RadioButton, DataTable, Paginator, Carousel, Chart, TabView, Accordion, Upload, Messages Module etc.
- Use **Angular CLI** to create code base and generate components, routes, services and pipes.
- Set up nested and multiple layers **Data Model** structure and **nested Form** structure for UI development foundation.
- Create App **Router Module**(including forRoot and forChild Router) and **Router-Outlet** view and associated components, set up **router-link** directive, using **routerLinkActive** directive toggles css active class based on the current RouteState, using **loadChildren** to implement **Lazy Load**.
- Use **Router** navigate to different view and **ActivateRoute** to subscribe router queryParams and detect router change in Angular.
- Tap into **Component Life cycle** and perform operations when required. For example: ngOnChanges, ngOnInit, ngOnDestroy, ngAfterViewInit etc.
- Use **@output** decorator emit the event by using **EventEmitter** API, when changes from child component need to be reflected to its parent component in hierarchy.
- Use **@input** data binding to pass the external data into component/directive.
- Use three types of **Directives** in Angular: Component directive, Structural directive(\*ngIf, \*ngFor) and Attribute directive(\*ngStyle, \*ngClass).
- Use **service** injected into different components to share common behavior and variable
- Use **shared Module** declarations common component and exports common components, sharing common components across different modules.

- Use **HttpClientModule** to initiate HTTP request and process responses and handle error within the application, we can directly access the JSON response by subscribing to the Observable(response result automatic convert from JSON to anonymous Object), tell HttpClient the type of the response to make consuming the output easier and more obvious.
- Use **Yarn** for package management tool, it avoids the problem where you remove a dependency from your package.json but the library is still there.
- Use UI toolkit to add CSS style to html page.
- Use **Loadash** library to handle array operation, such as filter, reject, some, get, orderBy etc.

## REACT JS DEVELOPER

Jan 2016– Mar 2017

J.P.Morgan Chase Bank – NY

Involved in Bank internal Big Data project--JPMC Developer Marketplace(<https://basilica-prototype.cfapps.io/>), the responsibility is for design and development of core UI platform that enables client sides delivery and construction processes for the Data Review, Data Discovery, Dispatcher Creating and Analytic group, leveraging emerging big data. It is a React JS based application, more specifically, the user of this UI platform is data scientist, when a data scientist log in the system, they can see their current project and all the jobs inside the project, also they can specify the parameters and create a new dispatcher, when the job finishes, they will see a notification the job is done and also see the dispatcher's IP address.

### Responsibilities:

- Use Pivotal **Tracker** software for **Agile** Project Management.
- Use **Git** for version control, deep understand of the concept of the three work areas of the Git project: the Git repository, the working directory, and the staging area. Familiar with the git branch operation, such as git **rebase**, git fetch and merge, git **stash and pop**, git revert etc.
- Set up **ESLint** for JS-Client built and development environment and JS-client continuous testing environment.
- Use **babel-polyfill** dependency to emulate a full **ES6** environment, also use babel-polyfill solve the **cross browser compatibility problem** which **promise** not working on IE11 while working fine on Chrome and Firefox.
- Implement **TDD** programming process, make debugging easier and optimize and simplify the code. Use **Mocha** and **Enzyme** to **unit test** React Components.
- Use **Sinon** JS to test asynchronous function in react, **stubbing/mock** a react component while still preserving the ability to test the full component lifecycle.
- Use **Nightmare** API to write **feature test** code.
- Create view-state manager with **React routers**, used React-Router to navigate between project list and project detail page.
- Built React Component, using **Props** to communicate between parent components and child components, as a Data Scientist, could view list of projects and instances related to selected project and create a spark&kafka dispatcher.
- Set **state** of higher-level **container component**, trigger the change of the **component's lifecycle** and re-render the component, and write **stateless functional component** for **presentational component** only focus on the UI rather than behavior.
- Use **ES6** Block-scoped variables, Constants, arrow function, template Literals, Export/Import Modules, Classes, and ES6 promise.
- Used **URI JS** library to manipulate with URLs to read and write all regular components using Chaining style.

- Use **Fetch** API for accessing and manipulating parts off the HTTP pipeline, such a requests and responses to fetch resources asynchronously across the network.
- Use lambda function bind when creating event handlers.
- Use **Yarn** for package management tool, it avoids the problem where you remove a dependency from your package.json but the library s still there.
- Use UI toolkit to add CSS style to html page.
- Use **Loadash** library to handle array operation, such as filter, reject, some, get , orderby etc.
- Use **Babelify** to transform ES6 code to ES5 version, then process by **Browserify**, which is to bundle up a simple file along with all of its dependencies.

## ANGULAR JS DEVELOPER(INTERNSHIP)

May 2015– Jan 2016

TD Ameritrade – NJ

TD Ameritrade is an online broker, which is well known for their competitive and most advanced web-trading platform. I worked on several projects along with major redesign with technologies like Javascript, jQuery, AngularJS, RequireJS, UnderscoreJS, HandleBars, SASS, Twitter Bootstrap.

### Responsibilities:

- Worked on redesigning of existing Application to mobile friendly single page application with **AngularJS & AngularUI**.
- Created quick prototypes with AngularJS & ReactJS.
- Implemented Model View Controller (**MVC**) architecture by using AngularJS.
- Rigorously used Angular **Directives** for developing personalized components.
- Used AngularJS **\$http, \$q** services to request **RESTful** Web Service and handle response. Proficiently used Angular concepts like **Services, Factory and Promises**, which helped in consuming and transforming the JSON data provided by the backend API.
- Used **NodeJS** server for developing the application.
- Data Modeling, Analysis and Data Visualization using **HighCharts** .
- Developed test cases and performed unit testing using **karma-Jasmine**.
- Used Grunt for automated **SASS** compile, Build & Test.
- Developed HTML5 Mobile application with HTML5, Java Script, jQuery, JSON and RESTful web service.
- Well aware of WCAG (Web Content Accessibility Guidelines) including Section 508 and aria.
- Closely Worked with Server side developers to integrate UI code with Java/J2EE/JSP.
- Actively participated and mentoring in requirements gathering, analysis, design, and development and testing phases.
- Established user interface design standards, internationalization considerations, framework guide and Developed HTML pages from functional specifications & wire frames also validated forms using AngularJS.
- Implemented **image lazy loading** to improve the page performance.
- Worked on **Java/J2EE** based Server side environments for multiple projects.

- Worked parallel with QA team to fix the reported bugs and defects and to ensure **cross-browser compatibility**.

## Involved Project

### **1. Connections for Employees:**

It's an internal portal mainly used for knowledge sharing for employees. It is also used for managing events and employee promotions. It is an AngularJS based application. I added multiple features periodically and also maintained the existing features based on changed requests.

- Implemented Multilingual feature to the site so that the users can interact/read in their own language, responsible for internationalizing(**i18n**) the web application based on W3 Standards.
- Customized the favorites feature to keep track of the content they like on the site by easily adding them to their favorites list.
- Integrated a customized Poll's feature to the entire community or for just a set of their friends.
- Used **AngularJS** to create an attractive, fun, and engaging user experience for an interactive quiz.
- Provided instant feedback when user answered questions and also created customized sharing links so users can challenge their friends through email.
- Created a responsive HTML5 Video Playback using AngularJS and Bootstrap and made it look intuitive and same in all browsers, unifying the user experience.
- Built customized playback controls, seek bar and time display.
- Utilized device specific media queries for responsive layouts.

### **2. Help Centre Module:**

- Converting existing applications with new single page application using HTML5, CSS3, JavaScript, jQuery and **AngularJS**. Implemented MVC pattern of AngularJS framework, created different modules, each module include specific controllers, services, custom filters and directives.
- Provided instant feedback when user asked questions and also created customized sharing links so users can easy to get help.
- Responsible for **internationalization(i18n)** the Web Application based on W3 standards, inject **angular-translate** and **angular-dynamic-locale** library, configure the translation files and change \$locale after the language change.
- Designed schedule appointment functionality, Injected **\$state** into controller, navigate the route to different state turn to different view when click buttons.
- Developed **custom directives** for drop down selected menu using **\$compile** service to implement complication.
- Implemented Angular **Service** as **Singleton**, share data and method between different components.
- Worked on top menu navigation bar, drop down menu, simple validations, slide show, alert boxes.
- Implemented angular **Dependency Injection**, created components(directive,service,filter,controller), resolve their dependencies, and provide them to other components as requested.

- Utilized device specific media queries for responsive layouts.also used **Bootstrap** components including table, grid to create responsive layout for new pages.

Environments: HTML/HTML5, CSS/CSS3, JavaScript, jQuery, jQuery UI, UnderscoreJS, RequireJS, AngularJS, SASS, Modernizr, XML, FireBug, Agile, Waterfall, Git, Notepad++, Brackets, HighCharts, Google map API, Grunt, NodeJs, Java/J2EE

## UI DEVELOPER

MAR 2011 – Sep 2013

Fuzhou Digital Creative Information Technology Co. , China

Involved in Logistics order processing system project, this customized web application provides the whole processing of creating order and order stacking for the company, integrates consignment management, receipt management, distribution management, shipping management, inventory management, vehicle management and other modules.

### Responsibilities:

- Involved in the redesign, update of the consignment, receipt and distribution management modules. Created requirement analysis, design, coding using **JavaScript, jQuery, HTML5, CSS3** in sublime editor.
- Updated UI implementing using advanced **HTML5, CSS/CSS3, JavaScript and jQuery** to make user interfaces and web components richer and dynamic.
- Created Responsive layouts using **Bootstrap**.
- Implemented MVC pattern of AngularJS framework to develop Single Page Application, managed changes in application views and complex nested views using **UI-Router**.
- Utilized device specific media queries for responsive layouts.
- Used Angular **UI Grid** for data edit and display.
- Developed custom AngularJS **directives** for reusable components across the application.
- Created custom AngularJS **services** for lean controllers and maintainable/testable code.
- Data Modeling, Analysis and Data Visualization using **HighCharts** .
- Created client side validation tool using Angular JS and Javascript.
- Used **NPM Bower and Grunt** for package and project task management.
- Performed test cases using **jasmine and Karma**, and used Protractor for End2End test.
- Developed API using AJAX, JSON to implement more robust and efficient browser and server transfer applications.
- Developed the user interface with fully cross browser compatibility for IE8 +, Safari, Firefox and Chrome.
- Worked on **Java/J2EE** based Server side environments for multiple projects.
- Maintain, enhance and recommend solutions that best suit business and technology needs from UI standpoint.

Environment: CSS/CSS3, HTML, JavaScript, jQuery, SQL Server, JSON, AngularJS, HighCharts, AJAX, GIT, SQL Server,Java/J2EE

## WEB DEVELOPER

Aug 2009 – MAR 2011

Fuzhou Jitong Information Technology Co. LTD – China



As a web developer, mainly involved in creating the front-end components, implementing UI designs, converting the clients requirements into responsive a web application. I worked on different projects like using LESS and SASS pre-processor of CSS and making a site dynamic with jQuery, Javascript, AJAX, and upgrading site appearance.

### **Responsibilities:**

- Developed organization's Web site and multiple custom web applications for various clients.
- Extensively responsible for the productive design and development of the user interfacing screens using **HTML, CSS, JavaScript, jQuery, AJAX**.
- Developed non-functional HTML, CSS pages from the mock ups and involved in UI review with UI **mockups** and Business Units.
- Consumed **Ajax** calls to build connection to database and retrieve updated data in **JSON** format.
- **Wireframes** are transformed to the full-fledged program code using the object oriented JavaScript, HTML and CSS.
- Built a website using **PHP**, HTML and **MySQL** to give the customers access of the data.
  
- Designed dynamic client-side JavaScript codes to build web forms and simulate process for web application, page navigation and form validations using jQuery and JavaScript.
- Developed jQuery and JavaScript code extensively for operating Document Object Model (DOM) and style sheets.
- Also Converted data from database into JSON format files and retrieve useful information using AJAX calls to display them on the browser.
- Worked through **cross browser compatibility** issues with layout and styles and fixed major bugs for supported browsers and platforms.

Environments: HTML 4.01, CSS, JavaScript, jQuery, DOM, AJAX, JSON, MySQL, PHP

### **RS(REMOTE SENSING)/GIS ENGINEER**

Aug 2007 – July 2009

Fuzhou Huihong InformationTechnology Co. LTD – China

Involved in Remote Sensing image processing(using ENVI/IDL ,ARCGIS) and online publication based on WebGIS image publication system in ArcIMS .

## **EDUCATION**

---

### **FUJIAN NORMAL UNIVERSITY - FUJIAN, CHINA 2007**

Master of Science in Cartography and Geographic Information System Science