Tingchao Xu

Phone: 520-270-0886 || Email: williamxu.cs@gmail.com
https://www.william-xu.com/ || 📫20362 Via Napoli, Cupertino, CA

Summary

- Over 7+ years of experience in Front-End and Back-End design and development. Rich experience in front and back-end technology such as **JavaScript**, **ES6**, **HTML**, **CSS3**, **React**, **Redux**, **React-Native**, **Vue, Vuex**, **Node JS**, **Express**, **JSON**, **NoSQL**, **JQuery**, **SQL**, **MongoDB, Restful API, GraphQL, Microservices Architecture, Micro Front-End Architecture, Unit Test, Functional Test**.
- Have rich experience in designing **Restful API**, setting up back-end servers that perform interaction with front-end, passing, saving, deleting and fetching data with **MongoDB** through **Mongoose**.
- Created several real-world applications using **react / vue** and **react-native** as full stack developer. Implementing **Redux / vuex** for managing data flow. Also, setup **API** requests for reaching data online.
- Implementing map, login, and other features. And finally, setting up the back-end with **Express**, **Nodejs** modified data into **MongoDB** through **Mongoose** with **CURD**.
- Experience in building the application using **SASS**, **CSS3** and libraries such as **Bootstrap** and **NativeBase** for UI design and family with **HTML, ES6** and **JavaScript** for web page development.
- Experience in fetching, saving, deleting and finding data with many conductions by using **MongoDB** or **SQL** with **Restful API** for the HTTP request and **CRUD** for communicating servers with databases.
- Strong experience in building web applications and setting up online with **AWS**, and setting up the domain using **GoDaddy**, building servers with **Nodejs**, **Express** and using **AWS** for server side.
- Experience in developing **Single-page applications(SPAs).**
- Experience in writing complex databases using **MongoDB**, can perform relationships between employee and employer, and also can update data related with those relationships through **Restful API**.
- Experience with building communications between servers using **GraphQL**.
- Experience with database queries, designing databases with **SQL, or MongoDB** with **Mongoose**.
- Strong experience of design **web application** and **mobile application**, professional working flow to making developing progress efficient
- Experience in developing web pages using **HTML/HTML5, CSS/CSS3, SASS/LESS**.
- Experience in **React JS** framework to develop the SPA
- Experience in **Redux** architecture with complex **Object-Oriented concepts** in order to improve the performance of the web application.
- Strong experience in working with **React JS** to create client side using **One-way data flow**, **Virtual DOM**, **JSX**, and **Redux**.
- Experience in **MVC architecture**, **Node.js** and **Express** to build the web application.
- Developed **Restful API** using **Node.js** and **Express** and setup **CRUD** at back-end. Using Mongoose to connect **MongoDB**, and using **CRUD** in the front-end through **Redux** to transfer data.
- Experience in building **Redux**, setting up **reducer**, **action**, and **action creators**, connecting state and methods from **reducer** to front-end as props using **connect**.

- Experience in setting up **Async** requests from **Restful API** using **Axios** in **Redux**, and using **thunk** for **middleware** in **Axios** requests.

Technology

- **Language**: JavaScript, Java, Python, HTML
- **IDE**: Mars, Eclipse, Sublime, Visual Studio
- **OS**: Windows, Mac, Linux
- **Frameworks / Libraries**: React, Redux, React-Native, Bootstrap, Vue, Vuex
- **Back-End**: Node.js, Express, Restful API
- **Database**: SQL, MySQL, MongoDB
- **Project Methodology**: Scrum, DevOpe

Education

B. S. Computer Science, University of Arizona, AZ

Work EXPERIENCE

AutoDesk, San Jose

Full Stack Developer Jan 2022 – Current

- Develop Autodesk Digital Payment System and application with **React, Redux, SASS, CSS3, Bootstrap, Nodejs, Express, JavaScript, ES6, Webpack, Microservices Architecture, AWS, Dynatrace, Micro Front-End Architecture, Restful API, GraphQL, Unit Test and Functional Test.**
- Develop Payment system front-end with **React.js, SASS, Internal Bootstrap library** and **HTML**.
- Develop and manage new Systems data flow with **React useContext, React useState, Redux and localStorage**.
- Create **React Custom Hooks** for accessing react ecosystem, handling data storage, data flow, and monitoring data change through **useEffect**, **useState**, **useContext** and **useMemo**.
- Experience with event-driven and scheduled **AWS Lambda** functions to trigger various AWSresources.
- Develop back-end with **Node.js, Express** and apply Microservices Architecture, Reaching different data from different services, such as authorization service from login page, calculating the tax based on user geolocation services and so on.
- Analyze and solve **cross-browser compatibility** challenges and debugging generated **HTML**, applied styles, and JavaScript in browsers using browser debug of different browsers like Safari, Firefox, Google Chrome, and Internet Explorer.
- Design and develop **RESTful API** with **CRUD** to exchange data from front-end to server.

- Experience to set up Async requests using **Axios** and **Promise** in **Redux**, and **Redux Thunk** in **middleware**.
- Experience of building **GraphQL** to fetch or transmit specific data structures.
- Experience writing end to end test with **Cypress automation test**.
- Testing code logic and flow with **Unit Tests JEST** for each react component, custom hooks and back-end services.

LogDNA, Mountain View

Software Engineer Feb 2020 – Oct 2021

- Develop LogDNA websites application with **Vue.js, Vuex**, **SASS**, **CSS3**, **Nodejs**, **Express**, **JavaScript, ES6, Webpack, Microservices Architecture, Restful API**, **Unit Test, MongoDB**.
- Create the LogDNA web app with **Vue.js** and **Vuex** as Front-End development.
- Lead a team to implement **Drift Chat** services into the main application, setting up different themes, messages, response, and email reply with **Dirft APIs**.
- Implementing **Stripe payment** services into the application by using **Stripe payment apis**, creating **invoices, plans, subscriptions** and others for various use cases, and also setting up the test account.
- Develop application graph sidebar with **Vue.js, CSS, Bootstrap** and **HTML**.
- Develop web UI interface using **CSS3**, **SASS** and **Libraries**.
- Develop websites use **Microservices Architecture**, reaching different data from different services, such as graph data from our backend service running with **ES search**, different time periods of logs, , user info and search result from backend.
- Use lazy loading to load logs, and sort or filter different log out by using **Vue** to display out in UI, **Vuex** for managing the data flow and multiple API calls for continuous fetching data from backend.
- Creating different util functions for handling time, such as time zone issue, time difference and others.
- Created and maintained **MongoDB** and **Nodejs** using **mongoose** and **Express**.
- Fetch the services **API** calls from back-end and map all the data in front end using **Vuex** with **Vue.js**.
- Use **Vuex** to manage the applications state and flow.
- Unit Test website component by using **Jest**, and **Vue test library.**
- Functional Test for web flow and React components use the **Cypress testing framework**.

Walmart Labs, Sunnyvale

Software Engineer Jun 2019 – Feb 2020

- Develop Sam's Club websites with **React, Redux**, **SASS**, **CSS3**, **Bootstrap**, **Nodejs**, **Express**, **JavaScript, ES6, Webpack, Microservices Architecture, Micro Front-End Architecture, Restful API**, **Unit Test and Functional Test**.
- Responsible for login page, two-factor authentication, forgot email page, reset password flow.
- Responsible for register flow, creating a new account with existing membership id and user info.
- Responsible for membership page, join membership page and payment methods.

- Develop website use **Microservices Architecture**, reaching different data from different service, such as authorization service for login page, payment service for authorize payments and so on.
- Develop website use **Micro Front-End Architecture** for developing separate page and components.
- Develop web UI interface using **CSS3**, **SASS** and **React internal Library**.
- Develop **React HOC** for login page, register page and membership page.
- Create **Redux Reducer**, and **Action Create** for storing data and control the page flow.
- Use **React Lifecycle methods** to control the web page flow.
- Build communications between servers by using **GraphQL** to pass specific nested data.
- Use **React Router** to redirecting the page, and **Query String** to sharing the website page.
- Develop website based on **WAI** and test website based on **WAI**.
- Unit Test website component by using **Mocha**, **Enzyme**.
- Functional Test for web flow and React component use **Nightwatch.js**.

Apple, Sunnyvale
Software Engineer Jul 2018 – Jun 2019

- Develop several websites with **React, Redux**, **SASS**, **CSS3**, **Bootstrap**, **Nodejs**, **Express**, **JavaScript, ES6, JQuery, Webpack** and **Restful API**.
- **Develop React Library** with **dynamic designing structure** and published it on NPM.
- Design the website structure using **HOC** and **pure component** to build a website based on json file.
- Develop the website with **dynamic concept** to let those does not have knowledge of **JavaScript** or react to build their own website.
- Designed the website using **CSS3**, **SASS** and **Bootstrap** as **UI** development.
- Create web app with **React.js** and **Redux** as Front-End development.
- Set up the back-end with **Node**.js and **Express** to react the data from server with react **socket**, and auto push the data to every linked client side.
- Design **Dynamic Redux Action Factory** to create **Redux Actions** and **Repository** dynamically.
- Use **React Router** to redirecting the website and **Query String** to sharing the website.
- Use **thunk**, **Axios**, **ajax** to reaching the data from server through api.
- Develop the website with **high order component** and **pure component** structure.
- Design the structure using **React recompose, Lodash**, and other useful react library to improve efficiency of developing website.
- Use **Code-splitting** with **Webpack bundler** and **Dynamic Imports** code design for **Lazy-Loading** maximize website performance.
- Use **Restful API** to communicate with server at the back end to fetch life data every 5 seconds.

WeHousing, Pleasanton
Software Engineer Jun 2017 – Jun 2018

- Created **React.js** / **Redux** front-end UI for displaying user information and user's.
- Designed the **UI interface** with grouping of **button**, **tags** and input to let user interact in client side.
- Designed and developed the data structure and functionality with **Redux**.
- Designed **Store**, **Action** and **Reducer**, ever times UI will interact the **state** through **Action**, and update the **React virtual DOM** with new state return in **Reducer**.
- Used **thunk** in the **Axios** so that it can **dispatch action** in the **Axios middleware**.
- Set up the **Restful API** at the back end using **Nodejs** and **Express**.
- Created and maintained **MongoDB** and **Nodejs** using **mongoose** and **Express**.
- Designed and developed the web page using **HTML5**, **CSS3** and **JavaScript**.
- Used **Restful API** to communicate and modify the **MongoDB** at the back end.
- Designed client-side UI with **CSS3**, Bootstrap and React components.
- Worked with implementing **APIs** with **MVC** pattern.
- Used **Axios** to communicate with database through **API**.
- Designed a dynamic and complicated page using **HTML5**, **CSS**, **JavaScript**, and **ReactJS**.
- Developed page layouts, navigation, buttons and tags and icons.
- Fetch the services **API** calls from back-end and map all the data in front end using **Redux** with **ReactJS**.
- Designed **GET**, **POST**, **FIND**, **DELETE** within the **API**, performance the functionality to communicate front-end with **MongoDB** through **Node.js** and **Express**.
- Developed mobile application with **React-Native**, **Redux**, **REST API**, **Node.js**, **Express**, **MongoDB**, **mongoose** and other **React-Native libraries**.

Daily-Diary, (Own Startup) JAN 2016 to May 2017

- **PROJECT**: Developed diary mobile app which can take everyday diary and auto save current location, weather info and other information. Has map which can display and number and location of where you made the diary.
- Designed the application's **UI** by using **Bootstrap**, **CSS3**, **SASS**.
- Built the application with nested **React-Native** components such as **input**, **button**, **tags** and others.
- Designed and store the database in **MongoDB** and communicating between **UI** and database using **Restful API** and **Axios**.
- Developed **Restful API** in the back-end with **Node.js**, **Express** and **thunk** for the **middleware**.
- Used fetching to **fetch** data from other website through their **API** such as weather info, geolocation and other information.
- Designed different pages with **React-Native**, **Bootstrap** and used **Router** to direct to different pages.
- Setting up the **Restful API** at the back end with **Node.js** and **Express**, designed all the data schema within it.
- Used lifecycle methods such as **componentDidMount** to setting up all the necessary data or components before rendering and used **componentDidUpdate** for re-rendering when **props** have been changed.
- Used **props** to pass the method, object and data from parent component to child **component**.