

Garfield Lee

[550.moe](#) | [LinkedIn](#) | [GitHub](#)

Location: Shanghai, China

Email: career@550.moe

WEB DEVELOPER

Five years of experience as a web developer, I'm familiar with **React, Next.js, and TypeScript**. I also have experience in Web3 and blockchain development, have worked on projects such as an **NFT marketplace**, an **NFT minting platform**, and a **blog platform based on Web3 technology**. Enjoy trying new web technologies and frameworks, actively promoting and using them in team projects.

TECHNICAL SKILLS

Languages	: TypeScript, HTML, CSS, C#, Zig
Frameworks	: React, Next.js, NestJS, Node.js
Libraries	: Zod, wagmi, Tailwind CSS, Prisma
Databases	: PostgreSQL, MySQL
Dev Tools	: ESLint, Visual Studio Code, Git, GitHub Copilot

PROJECTS

Meta Network Homepage	<i>Next.js, TypeScript, Chakra UI, Emotion</i>	Source Code
<ul style="list-style-type: none">• Add new components to the page.• Make the page looks more consistent.		
Meta CMS Backend	<i>NestJS, TypeScript, TypeORM, MySQL, Redis</i>	Source Code
<ul style="list-style-type: none">• Build the basic blocks of the project.• Connect to MySQL utilizing TypeORM.• Setup OpenAPI (Swagger) for web developer to consume.• Integrate with Grafana Loki using Winston, logging simultaneously to the terminal, file, and Loki.• Create a corresponding monitoring dashboard for the queueing system built with Bull and NATS.• Add AuthN (authentication) to the app, using JWT passed by in-house UCenter, Passport and NestJS guard.• Utilize microservices architecture design to request basic user information, storage service access token and social media handle from UCenter.• Allow user entering information about their site, which will be passed to Meta CMS Worker after verification using class-validator.• Integrate with GitHub and Gitee, requesting access to a new or existing repository for Worker to store CMS data.• Persist CMS article metadata with Arweave, use pRetry to handle transient failures.• Cache requests using cache-manager, avoiding unnecessary database queries.• Perform unit testing for critical logic with Jest.• Containerize the whole app with Docker Compose.		
Meta CMS Worker	<i>Node.js, TypeScript, Hexo, Git, execa</i>	Source Code
<ul style="list-style-type: none">• Encapsulate git(1) with execa, since other mainstream git library in Node.js are basically unusable.• Call internal API of Hexo to create, update and delete articles and to generate websites.• Mux custom configuration from the user with the default configuration, generating a new configuration file for Hexo to consume.• Pushing changes to GitHub or Gitee repository, based on information sent from Backend.• Integrate with Grafana Loki using Winston, logging simultaneously to the terminal, file, and Loki, preventing Denial of Service attack.• Perform unit testing for important logic with AVA and Sinon.JS.• Containerize the whole app with Docker.		
ANE NFT Marketplace	<i>React, TypeScript, styled-components, useSWR, Ethers.js, web3-react</i>	Source Code
<ul style="list-style-type: none">• Recreate the design draft using styled-components.		

- Create a decentralized marketplace for ANE with Uniswap Protocol V2 SDK, allowing exchange between ANE token and others, creation of new exchange pairs, and liquidity mining.
- Utilizing Zora Media Protocol to handle the creation, bidding and asking of NFTs.
- Connect to MetaMask and WalletConnect using Ethers.js and web3-react, allowing users to login and to manage their assets.
- Fetch NFT metadata, bids and asks using useSWR.
- Extract custom React Hooks from the UI components.

W3itch Fronted

Next.js, TypeScript, Material UI, Emotion, Ethers.js, useWallet

- Crafted using Material UI and Emotion.
- Different layouts across the site, providing a better user experience.
- AuthN using OAuth 2.0.
- Connect with useWallet.
- Implement user information and settings form using React Hook Form and class-validator.
- AuthZ (authorization) with useContext and useReducer.
- Load and paginate data from the backend using getServerSideProps and useSWR.
- Allow user to filter games by category and/or tags, implemented with React Select and custom Hooks to synchronize the filter state and the URL, allowing users to share the URL with others.
- Manage ERC20 tokens with custom Hooks and Ethers.js.
- A11y (accessibility) with ARIA (Accessible Rich Internet Applications).
- Split existing large components into several components and custom React Hooks, increasing maintainability.
- Prevent unnecessary re-renders using useMemo and useCallback.
- Perform unit testing for business logic with Jest.

AmbrusStudio Mint page

Vue.js, TypeScript, Headless UI, Ethers.js, Web3Modal, wagmi

- Migrate the existing Vue CLI project to Vite.
- Implement the page using UnoCSS and Headless UI.
- Refactor duplicate code in components, decreasing LoC (Lines of Code) in total.
- Author a new Vue component using Vue Composition API, based on Web3Modal HTML and wagmi core, for wallet connection in Vue.
- Mint NFTs with wagmi core and Ethers.js, calling in-house seller contract on Ethereum.
- Perform unit testing for significant logic with Vitest.

AmbrusStudio User Center

React, TypeScript, Headless UI, Ethers.js, Web3Modal, wagmi, ImmutableX SDK

- Built with Vite, UnoCSS, Headless UI, React Hook Form, Web3Modal and wagmi.
- Construct engineering system with Turborepo and pnpm workspace, to share UI components and logic between projects.
- Encapsule ImmutableX SDK into a React Hook.
- Use AbortController to optimize rendering and network requests.
- Refactor filter logic in the gallery, increase rendering performance.

AgentSwap Civitai

Next.js, TypeScript, tRPC, NextAuth.js, Web3Modal, wagmi, SIWE

[Source Code](#)

- Utilize NextAuth.js and SIWE (Sign-In With Ethereum) to implement login function.
- Share SIWE logic between components using custom React Hooks.
- Simplify front-end Hosting Worker API calls using tRPC.
- Verify API parameters with Zod.
- Create webhook to receive events from Hosting Worker, update the database accordingly.

AgentSwap Hosting Worker

Node.js, TypeScript, H3, Zod, execa, Octokit, Pino

[Source Code](#)

- Implement necessary APIs using Node.js and H3.
- Output logs to file and terminal with Pino, Pino HTTP and Pino Pretty.
- Verify environment variables and API parameters with Zod and h3-zod.
- Encapsulate git(1), docker(1) and systemctl(1) (for controlling Caddy) with execa, to build and deploy Gradio projects.

- Check if the repository exists, using Octokit and Hugging Face JS libraries.

Taro	<i>React, TypeScript</i>	Source Code
<ul style="list-style-type: none"> • Triage issues on GitHub. • Seasonal contributor to the codebase. 		
Taro UI	<i>Taro, React, TypeScript</i>	Source Code
<ul style="list-style-type: none"> • Triage issues on GitHub. • Seasonal contributor to the codebase. 		

EXPERIENCE

Web Developer <i>Qiubi Technology (Shanghai) Co., Ltd.</i>	Nov 2020 – Jun 2023 <i>Remote – Shanghai, China</i>
<ul style="list-style-type: none"> • I added new features for an enhanced Civitai project, to allow users to use Ethereum wallet login. I created a custom NextAuth.js Credentials Provider and used SIWE for signature verification. Web3Modal and wagmi were used on the front-end to connect to the wallet. To implement the hosting of Gradio projects, I developed a back-end service using H3, validated environment variables and API parameters with Zod, and wrapped Git and Docker commands with Execa to package Gradio projects from GitHub or Hugging Face into Docker images, hosted on the server with custom ports and domains so that the front-end can display Gradio projects. • Collaborated with our CTO to build a back-end service for a web3 blog toolkit. Using NestJS, TypeORM, MySQL, and TypeScript, I completed the back-end API for blog information, storage targets, and theme configuration, and integrated Arweave to store user post's metadata. I built a website generator based on Hexo using Node.js, which can automatically generate and deploy Hexo websites according to the presets provided by the back-end. Received 200 new registrations just on the first day. • I assisted a web3 MOBA game company, developing NFT minting websites and user center websites. I used Vue.js, Headless UI, UnoCSS (a Tailwind CSS alternative), and TypeScript to build the minting website. As Vue does not have a good wallet connection framework, I utilized the Vue's Composition API to encapsulate Web3Modal HTML and wagmi Core to achieve similar functionality as Web3Modal React. During my maintenance of this project, the company successfully conducted four rounds of NFT issuance activities. Together with my colleagues, we built the user center website using React and TypeScript. I split UI elements into components for efficient reuse and also helped optimize some React Hooks to improve page performance. To better reuse UI components and share business logic, I used monorepo to reorganize code from multiple projects into independent packages which solved problems such as copy-pasting and logic synchronization. 	
Web Developer <i>Shanghai Hayi Information Technology Co., Ltd.</i>	Nov 2018 – Oct 2020 <i>Shanghai, China</i>
<ul style="list-style-type: none"> • I am responsible for the development of the company's WeChat Mini Program project. I utilize Taro to standardize the development process and to improve code reuse rate. With Taro's comprehensive support for React, I can easily handle the development and maintenance of different mini-program platforms, effectively reducing development costs. In addition, I also promote TypeScript and ESLint to improve code quality and reduce bugs during production. With my assistance, the company's projects have achieved more efficient development and higher quality code implementation. • I am responsible for customizing the October CMS theme for the company, using TypeScript and Sass to write front-end logic and styles, and using Webpack and PostCSS to package scripts and resource files. By removing messy script tags and style tags, I have reduced issues such as functional failures and style conflicts while increasing code maintainability. Additionally, by integrating commonly used October CMS components and styles into an npm package, I have achieved better code sharing which has improved team development efficiency. With my assistance, our clients have received a better user experience with higher quality project implementation. 	
Software Developer Intern	Aug 2017 – Feb 2018
<ul style="list-style-type: none"> • Assisted lead developer maintaining and developing new features using C# and ASP.NET Core. 	

EDUCATION

Shandong Vocational College of Information Technology <i>Software Engineering</i>	Weifang, Shandong, China <i>Jul 2015 – Jul 2018</i>
---	--