Frontend Developer

Wang Shouren, Male, Han Nationality, 26 years old, 5 years of experience

Skills

- Strong knowledge of JavaScript and TypeScript.
- Proficiency in React and a deep understanding of React's core principles, rich experience in developing and optimizing performance in React and TypeScript projects.
- Experience with HTML5, CSS3, and responsive design principles.
- Familiar with object-oriented programming and test-driven development (TDD) concepts, and knowledgeable about common design patterns.
- Possesses a good coding style and prefers to write clean code, practical experience applying SOLID principles to development.
- Solid coding ability, having solved over 250 problems on LeetCode, and understanding TypeScript type gymnastics.
- Strong debugging and problem-solving skills.
- Familiarity with reactive programming principles for complex UIs.
- Good spoken English (B1).

1. Language Skills
    - Proficient in HTML, CSS, JavaScript, TypeScript
    - Familiar with Less
2. Frameworks
    - Proficient in React
    - Proficient in using Webpack and Vite
    - Familiar with Next.js, Vue.js, etc.
3. Libraries
    - Familiar with Redux, React Router, Ant Design, Rxjs, Mobx, ahooks, Tailwind CSS, Lodash, Axios, Storybook, etc.
4. Backend Skills
    - Familiar with Node.js
    - Knowledgeable about MySQL, MongoDB, HTTP, Prisma, WebSocket
5. Package Managers
    - Familiar with npm, pnpm, and yarn
6. Integration Tools
    - Basic understanding of Github Actions, Docker, and Docker Compose
    - Basic understanding of AWS, including EC2 and Route 53
    - Basic understanding of Linux
7. Artificial Intelligence
    - Using AI tools to enhance development efficiency, such as Github Copilot, ChatGPT, Claude, Cursor, V0, etc.
    - Experience in deploying open-source AI models
8. Web3
    - Understanding of basic blockchain principles
    - Familiar with Solidity, Ethers.js, and able to deploy contracts to blockchain test networks using Hardhat or Foundry
9. Testing Frameworks
    - Familiar with Vitest, mastering component testing and unit testing
    - Understanding of Playwright, capable of performing E2E testing

Work Experience

1. Daguan Data 2020 - Present, serving as a frontend technical expert in a team of about 70 frontend developers
    - Team leader managing about 10 frontend developers
    - Technically using Webpack and React to develop and build frontend applications
    - Assisting junior frontend engineers in solving development issues
    - Contributing components and code to the company's frontend component library
    - Won first prize in the frontend skills competition organized by the company
    - Responsible for frontend interview work
    - Participated in the frontend engineering design of company projects
    - Organized documentation for multiple open-source projects, which can be viewed on Github
    - Led the team to achieve an on-time delivery rate of over 95% for completed projects
    - Guided junior engineers to independently undertake development tasks within an average of 3 months
    - Deployed open-source AI models using ollama and open-webui to provide AI-assisted development for the team

Project Experience

Knowledge Middle Platform

Technologies used: React, Ant Design, Rx.js, Mobx, Axios, ECharts, ahooks

- Successfully reproduced the homepage dashboard design, receiving client praise
- Integrated single sign-on functionality
- Created multi-window communication capabilities
- Led the design of backend chart data structures
- Optimized frontend code using the chain of responsibility pattern, simplifying over 20 consecutive if-else statements
- Implemented decision tree visualization: created a visual representation of decision trees using tree data structure objects
- Built a desktop application using Electron
- Implemented lazy loading for various parts of the homepage
- Resolved browser compatibility issues to support Firefox 52 and Chrome 69

Standardized Frontend Design

Technology stack: React, Dumi, Monaco Editor, etc.

- Built the project from scratch, including documentation website and Playground website, writing development and usage documentation.
- Automatically generated navigation and sidebar structure by scanning the source code directory, achieving zero-configuration navigation menus.
- Integrated login page, route switching, state storage, internationalization, theme switching, notification bar, mock data, and about 10 example pages.
- Used code parsing tools to automatically generate API documentation, enhancing development efficiency.
- Improved build speed through turbo build caching tools.

Poster Low-Code Platform

This project mainly interfaces with poster marketing business, aiming to reduce repetitive labor and development costs. Developed a frontend web editor based on simple drag-and-drop, and self-developed a rendering engine capable of rendering Vue or React components, with features including UI editor, light code, function as a service (FaaS), and preview page.

Technology stack: React, Ant Design, Node.js, Express

- Responsible for the domain model design and implementation of the entire frontend system, building the project from scratch, developing the editor, renderer, FaaS, and light code functionality.
- The editor implemented alignment guides, node position/size adjustments, node deletion, bring forward/send backward, copy/paste, and history features.
- Also wrote docker-compose files to configure MySQL and Nginx environments, and used GitLab, Jenkins, and Docker for continuous integration.
- Resolved redundancy issues in Webpack configuration files for multiple frontend projects, wrote build scripts, and used the strategy pattern to customize package.json file content to distinguish project types and follow different build processes.
- Each development required manually starting multiple frontend and backend projects, causing repetitive labor; wrote Node.js scripts to use PM2 for one-click startup of frontend development servers and backend services.
- Resolved the coupling issue between the editor UI model and business logic code, designed a solution using the publish-subscribe pattern and state machine to achieve bidirectional binding and decoupling of the UI model and view, reducing the amount of frontend business logic code by about 60%, enhancing code readability.
- Adopted optimization solutions such as lazy loading, preloading, hot loading, code splitting/compression, CSS file extraction, and CDN loading of third-party modules, achieving second-level loading times for light code and preview pages.
- Reduced web development costs, improving delivery efficiency from the original 3-4 days to 1-2 days, with an efficiency increase of about 50%.
- Distinguished between testing and production environments, set version control for materials and pages, enhancing project stability.
- Used Vitest for unit testing, achieving a statement coverage rate of over 90%.
- Used Playwright for E2E testing of highly interactive pages with drag-and-drop features, ensuring a smooth and error-free user experience.

Personal Projects

Full-Stack Open Source Project Collection

Project link: GitHub - Next.js Practical Cases

Technology stack:

- Frontend: Next.js 14, React 18, TypeScript, Tailwind CSS, Storybook
- Backend: Prisma, PostgreSQL, Docker
- Tools: Turborepo, pnpm, Vitest, Playwright

Project Highlights:

1. Monorepo Architecture Design
    - Built a monorepo project using Turborepo + pnpm workspace
    - Abstracted shared component libraries and tool libraries to improve code reuse
    - Unified dependency version management to reduce maintenance costs
2. Stack Overflow Clone Application
    - Implemented core features such as user authentication, Q&A, tags, and voting
    - Built data models and relationships using Prisma + PostgreSQL
    - Used ClerkJS for identity verification and authorization
3. Large File Upload Application
    - Implemented features such as file chunk uploads, resumable uploads, and instant uploads
    - Used WebSocket for real-time progress display
    - Adopted streaming processing to optimize memory usage
4. Engineering Practices
    - Used Storybook for component-driven development
    - Configured ESLint + Prettier to standardize code style
    - Used Vitest + Playwright for unit testing and E2E testing
    - Achieved automated deployment with Docker

Performance Optimization:

- Implemented on-demand component loading
- Configured build caching to improve compilation speed
- Optimized database query performance

Project Achievements:

- Open-sourced the project and received GitHub Stars
- Almost every public component has complete documentation and examples
- Supports one-click deployment to production environments

ERC20 Token Contract Example

This is a demonstration project for a decentralized application (dApp) based on the ERC20 protocol, built using Hardhat, RainbowKit, and PNPM Workspace.

Project link: GitHub - Web3 Learning Journey

Technology stack:

- Smart Contracts: Developed and locally deployed using Hardhat, supporting deployment to the Sepolia test network.
- Frontend: Used RainbowKit, wagmi, ethers.js for smart contract interaction.

Functionality:

- Implemented basic functions of ERC20 tokens using ethers.js and wagmi, including wallet connection, transfers, and balance inquiries.

Social Links

- Github: https://github.com/wangshouren7
- Youtube: https://www.youtube.com/@%E9%A6%96%E4%BA%BA%E7%8E%8B

Languages

1. English (B1)
2. Chinese (Fluent)

Location

- Chengdu, China

Contact Information

- wangshouren116@gmail.com

Education

- Chengdu Neusoft University, Bachelor of Computer Science and Technology