



Design and Implementation of a School Calendar Adjustment System Webpage Based on React

Yuxuan Chen

College of Electrical Engineering, Zhejiang University, Hangzhou, 310058, China
3220104050@zju.edu.cn

Abstract. In the rapidly evolving landscape of front-end development, frameworks like React have become indispensable tools that enable developers to build complex, responsive web applications. However, during the process of development with React, developers need to address the steep learning curve associated with React. Besides, a mindset of componentization development is required in React application development. Therefore, the research applies the React framework in a common scenario, which is a school calendar adjustment system, to help developers with a common development process. This scenario was chosen due to its relevance to a wide range of educational institutions and its requirement for a dynamic and user-friendly interface. The project meticulously crafted the application's architecture, ensuring the user interface (UI), data flows, and application programming interface (API) interactions were all designed with best practices in mind. This approach resulted in a refined single-page application that boasts a clear and maintainable code structure. The insights gained from this study can help developers understand the nuances of React, enabling them to create more intuitive and maintainable web applications.

Keywords: Webpage Design, React, Componentization, Ant-design.

1 Introduction

With the rapid development of the Internet, the demand for delicate and complex web applications has grown fast [1]. Users are no longer satisfied with static information presentation but expect dynamic web content, responsive interfaces, abundant functionalities, high performance, and good user experiences [2]. Traditional front-end development tools represented by HTML, CSS, and JavaScript can hardly fulfill developers' requirements. Even though they can handle these demands, they require considerable proficiency and can lead to cumbersome code, seriously affecting readability and maintainability. Therefore, to satisfy the growing needs of front-end development, front-end frameworks have emerged and evolved continuously. Meanwhile, abundant libraries based on these frameworks have been developed to handle different situations. These new technologies have overturned the

structure of web projects and changed the traditional web development process significantly.

React developed by Facebook, has experienced a meteoric rise in popularity. This JavaScript library streamlines the creation of web applications by enabling efficient updates to web pages without full-page refreshes—a feature that is especially beneficial for single-page applications [3].

However, adopting React is not without its challenges. The steep learning curve introduces novice developers to new concepts such as the virtual document object model (DOM), component lifecycle, state management, and hooks. Designing a component architecture that is both maintainable and reusable demands both foresight and experience, as missteps can lead to significant maintenance and performance issues. Developers must also navigate a vast ecosystem of accompanying tools and libraries, stay current with frequent updates, and handle complex testing scenarios, particularly when dealing with hooks, states, and contexts.

This research provides a structured approach to overcome these hurdles by embracing a philosophy of incremental learning and hands-on application. It begins with a foundational overview of React's core features—like componentization and community support—presented through straightforward examples to aid comprehension. The study then applies these principles in a practical setting, using a school calendar adjustment system as the thesis project to illustrate how React's component-based architecture can simplify code management and leverage the extensive community support to address development challenges.

By integrating various React-based libraries and code quality tools, this project not only showcases the practical application of React but also aims to distill a design philosophy and development approach that can inform and guide future developers. The goal is to demystify the process of creating React web pages, thereby equipping developers with the clarity and knowledge needed to harness the full potential of this powerful front-end library. The findings from this study have the potential to influence how developers approach similar projects, encouraging a more thoughtful and strategic use of the React framework.

Recognizing the inherent challenges of adapting to new technologies, the limitations encountered during the development process are discussed in this research. By providing a candid account of these experiences, the study offers a balanced perspective that can inform future projects and guide developers in anticipating and overcoming similar obstacles.

2 Methodology

2.1 Web Design and Development Process

Before starting coding, it is necessary to determine the basic content that the website needs to include and to design the structure of the web page. Subsequently, according to the design and functionality, select the suitable front-end frameworks and essential libraries. Once all development is completed, the code must be carefully tested. The

testing includes functional testing, interaction testing, compatibility testing, front-end and back-end integration testing, and more. Finally, the website can be deployed to the server. Even after deployment, it is necessary to regularly check and maintain the website. Fig. 1 shows the development process of this project.

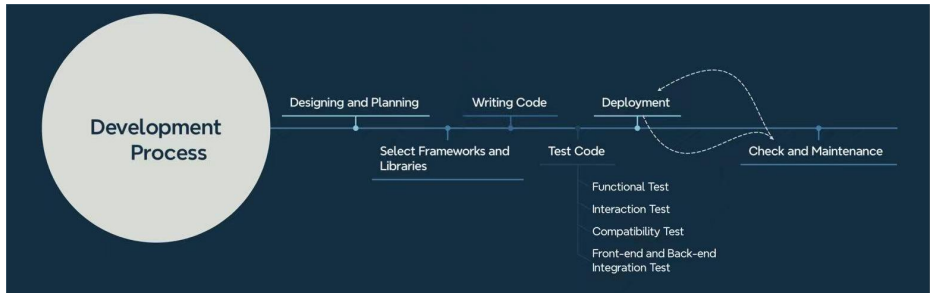


Fig. 1. The process of web application development. (Photo/Picture credit : Original)

2.2 Front End Development Framework Selection

Various tools have been used by researchers in the web design field to meet the diversified demand for web applications [4]. Front-end frameworks are a popular and widely used technology.

Front-end frameworks are pre-written standardized libraries and toolsets, which provide various components for building responsive, interactive interfaces. Using these frameworks can enhance development efficiency and optimize code readability.

The three most popular front-end frameworks are React, Angular, and Vue. They have different characters and benefits. This project selects React as the front-end framework, for it has several unique advantages. React has a large and active developer community, which provides ample third-party libraries to easily build attractive user interfaces. Moreover, React provides a component-based development framework and advocates for embedding HTML and CSS within JavaScript, making the code structure clearer [5].

Besides, the project chooses npm as the package manager to install the required libraries.

2.3 React-based Libraries

React development community provides various libraries and tools for user interface and functionality. To simplify development and achieve better functionality, the project chooses two main additional libraries. For the user interface, the project chooses ant-design as a component library, which provides streamlined and attractive components with fine interactive performance. To realize the switch between the semester information page and the compensatory leave adjustment page, the project

chooses React-Router-DOM, which enables navigation within the app. In addition, the project uses TypeScript for API writing to keep the type of validity.

2.4 Code Quality Evaluation Tools

It is rational to emphasize evaluating the final product, i.e. the code, because the core of development is on the code level [6]. Of course, code quality can be subjectively evaluated from several aspects, such as readability, extendibility, security, etc. However, there is a necessity to use some code quality evaluation tools to attain objective evaluation. This project, as a front-end project, uses ESLint for JavaScript code evaluation, because it supports React code and CSSLint for CSS code evaluation.

3 Results and Discussions

3.1 Fine Framework for School Calendar Web Design

A detailed and reasonable framework is of vital importance for web development. The framework of this project can be divided into three parts: interface layout, data structure, and API interfaces.

Interface Layout Design. The interface layout of a webpage should be designed based on the functionality of the webpage. A good design should come from user needs and go beyond them at the same time [7]. This project mainly has two functional modules: semester management and compensatory leave adjustment. Therefore, the application accordingly has two pages. To utilize the componentization features of the React framework better, the two pages share a head bar and menu bar. The switch between two pages is controlled by the React-Router-DOM library. Besides, to achieve the addition of semester and compensatory leave recording functions, two floating window pages were designed using ant design components. The web structure is shown in Fig. 2.

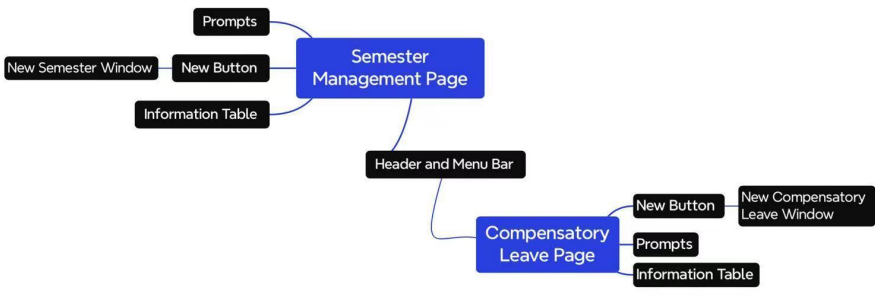


Fig. 2. The webpage structure. (Photo/Picture credit : Original)

Data Structure Design. In considering of storage requirements of back-end, it is important to design a clear, effective, and suitable data structure. The main data in the application are semester information and compensatory leave information. One semester information simply consists of a semester name and its year. One compensatory leave information includes the “from” and “to” dates of the day off and the semester in which it is. Therefore, the semester data structure can be reused. Fig. 3 shows the data structure of the project.

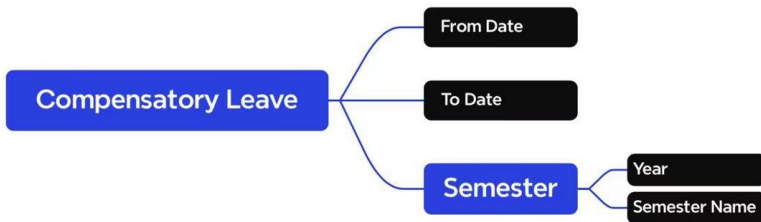


Fig. 3. The data structure. (Photo/Picture credit : Original)

API Interface Design. As a practical web application, this project possesses standardized front-end APIs. They provide a series of interfaces to achieve efficient communication with the back-end. As shown in Fig. 4, regular visitors can use “/api” interfaces to get data from back-end, while administrators can use “/admin” interfaces to set or delete data.

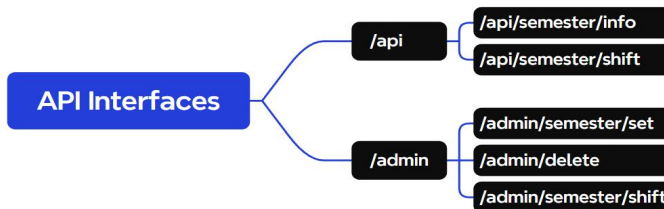


Fig. 4. The API structure. (Photo/Picture credit : Original)

3.2 Classification and Planning of Interface Design Patterns

The main interface of this project consists of two pages, a shared head bar and a menu bar. However, the project is still a single-page application, because there is only one entry HTML file. A successful website is attractive to users, making them feel trustworthy and satisfied [8].

Therefore, it is necessary to use a unified style to demonstrate professionalism. At the same time, the webpage should have some unique styles to avoid homogenization. In this project, the head bar uses gradient colors to show a unique style and attract users as well. The main body below uses the default style provided by ant-design to display a concise interface. In this way, the website can meet the previous two requirements.

3.3 React-based Web UI Interface Display

One of the most salient features of React is componentization. Using component-based development methods, UI is broken down into small components, each with some properties or states. By combining these small components, large components are built, improving code reusability [9]. As shown in Fig. 5, the web page body consists of the following components in Table 1, whose prototype can be obtained from ant-design. Besides, clicking on the button can trigger a floating window with the useState method of React. The window is shown in Figure 6, with components listed in Table 2.

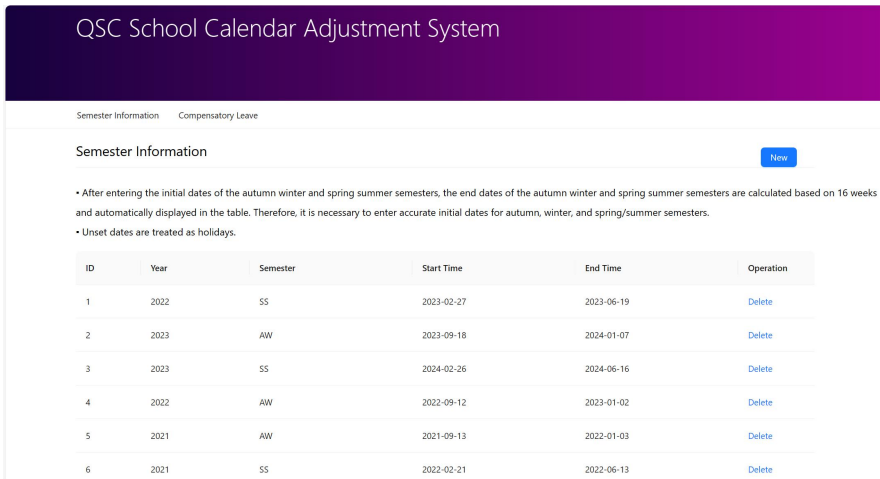


Fig. 5. UI of web page body. (Photo/Picture credit : Original)

Table 1. Components and their functionalities in web page body.

Component	Functionality
Menu	Change its key, which is the index of the page, with a mouse click.
Title	Display page title.
Button	Respond to mouse-click events and open a floating window.
Paragraph	Display page instructions and prompts.

Table

Display data and provide hyperlinks for deleting data.

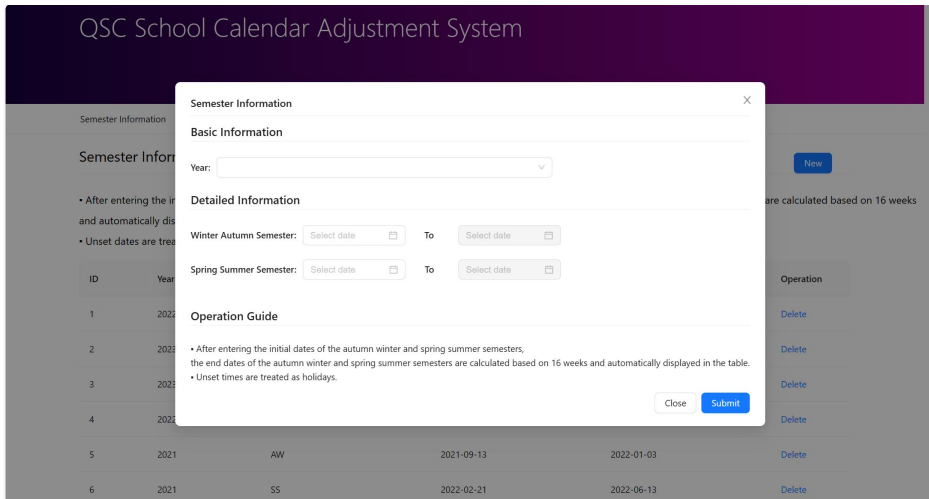


Fig. 6. UI of the float window. (Photo/Picture credit : Original)

Table 2. Components and their functionalities in the float window.

Component	Functionality
Title&Text	Display prompts.
Selector	Select the year.
Date Picker	Pick the start date and end date for the semester.
Button	Submit the data to the back-end.
Component	Functionality

3.4 API Library Designed with TypeScript

TypeScript has received growing attention since it was published in 2012. It became the fifth most popular programming language all over the world [10]. TypeScript is a superset of JavaScript. In detail, it is a JavaScript-like language with strict type requirements. Therefore, it is especially suitable for writing APIs, because a tiny type mistake causes a big error in back-end and database. In this project, all APIs were developed based on a universal web fetch API, so that a set of rules was encapsulated in the reused request header. When modifying cross-domain policies or back-end URLs, only the general request header needs to be modified. The specific APIs only need to add the routing and body of the request to this universal interface. In this method, the amount of code and maintenance difficulty are greatly reduced.

3.5 Evaluation of Web Design Results

The evaluation of a website is useful for ensuring the quality of web pages. Methods to evaluate a website can be divided into 3 categories: based on the user, based on the evaluator, and based on automatic website evaluation tools [11]. This project was mainly evaluated by the evaluator-based method in two aspects: functional integrity and code quality.

Functional Integrity. As a functional webpage, the integrity of functionality is the key element to evaluate the quality of the project. There is a list of basic functions the webpage is required to implement:

- 1) Get semester data from the back-end and display it to the user.
- 2) Get compensatory leave data from the back-end and display it to the user.
- 3) Add new semester data or new compensatory leave data and push it to the back-end.
- 4) Delete a semester data or a compensatory leave data and push it to back-end.

After joint testing at the front and back ends, the project has ensured that all functions can be implemented. Moreover, in some aspects, functions were improved in comparison to the plan as follows:

- 1) Due to the fixed length of the semester, automatic calculation of the end date of the semester has been achieved when inputting semester information.
- 2) Delete data in the table by clicking the delete button before the request is sent to back-end.

Nevertheless, there is still room for improvement in the functionality as follows:

- 1) There are not too many compensatory leave data, otherwise a searching function is necessary.
- 2) The batch addition function is not implemented.

Code Quality. It is essential to evaluate code quality after development. In this project, both subjective and objective methods are used to evaluate code quality.

Subjectively, code quality can be assessed from these perspectives: readability, maintainability, performance, extensibility, security, etc. In this project, performance is not considered, because complex algorithms or multithreading are not included in the project. In terms of readability and maintainability, because of the componentization characteristics of React, the project is modular, therefore increasing the readability. This is not conducive to subsequent maintenance. As for extensibility, the project uses an extended API writing style, which makes the project extensible. In the security aspect, when sending a request to the back-end, a session token cookie is appended to the request head. In this way, the server can validate the identity of the user to avoid unauthorized modification.

Objectively, two code quality inspection tools were used in this project, namely ESLint, and CSSLint. ESLint can detect syntax errors and irregularities. Running ESLint in the command line, it did not find any error, nor gave any warning. Similarly, the project used CSSLint to examine CSS code. No errors were detected,

while some warnings were reported. According to the characteristics of React, CSS needs to be written carefully, because IDs and classnames are shared throughout the entire project. It would be better to use embedded CSS in the React project.

4 Conclusion

The project adopted a detailed and reasonable development methodology to design and implement a school calendar management system. By carefully selecting front-end development frameworks and libraries, and utilizing code quality assessment tools, a fully functional and user-friendly website was established. React was chosen as the front-end framework mainly because it has strong development community support and provides a clear component-based development approach. By combining the ant design component library and react-router-DOM to achieve navigation between pages, the website has successfully achieved both aesthetic and practical results.

In terms of data structure and API interface design, the project has adopted clear and effective strategies to ensure efficient communication with the backend. By using TypeScript to write APIs, type safety is enhanced, and the possibility of serious errors caused by type errors in the back-end and database is reduced.

The design of the UI interface adopts a responsive design pattern, which not only enables the website to maintain functionality and aesthetics on different devices but also improves the reusability of the code. The componentized approach makes the interface easy to maintain and has strong scalability.

The functional integrity assessment shows that this project not only meets the basic requirements but also exceeds initial expectations in certain aspects, such as the automatic calculation function of the end-of-semester date and the ability to directly delete data from the table before sending back-end requests.

Overall, by integrating modern front-end technology and best practices, this project has successfully demonstrated an efficient, maintainable, and user-friendly school calendar management system. In future work, enhancing website functionality will be a key consideration to enhance user experience.

This study demonstrates an efficient and modular design and implementation methodology in the field of front-end development, providing a clear framework and case reference for future related research and practice. The successful implementation of this project highlights the importance of component-based development, interaction design, and code quality assessment tools in achieving dynamic and user-friendly web applications using React and its libraries in the ecosystem. In addition, the application of TypeScript further emphasizes the value of type safety in improving the stability of large-scale applications, providing useful insights for future researchers and developers when choosing technology stacks and coding practices.

Although this study has achieved significant results in the front-end development process and technology selection, there is still room for improvement in performance optimization and cross-platform adaptability. For example, the potential impact of server-side rendering (SSR) on SEO and loading time, as well as optimization

strategies for mobile devices, has not been thoroughly explored. Future research can focus on these aspects, especially in improving performance under large-scale user access, as well as further enhancing compatibility and user experience across devices and browsers. In addition, considering the evolving nature of web development, continuous attention and integration of emerging technologies and frameworks is an important task for future research.

References

1. Kaluža, M., Krešimir, T., Bernard, V.: Comparison of Front-end Frameworks for Web Applications Development. *University Scientists at the River* 6(1): 261-282 (2018).
2. Dinh, D., Zhuanyan, W.: Modern Front-end Web Development—How libraries and frameworks transform everything. MSc. Thesis, Turku, Turku University of Applied Science (2020).
3. Aggarwal, S.: Modern Web-Development using ReactJS. *International Journal of Recent Research Aspects* ISSN: 2349-7688 5(1), 133-137 (2018).
4. Shrestha, D., Tan, W., Sumina M., et al.: A UML Based Approach for Analysis and Design of Tourism Web Portal. In: *International Conference on Smart Electronics and Communication 2020*, pp. 236-243. IEEE, Trichy, India (2020).
5. Chen, S., Upendar-Rao, T., Venkata-Koteswara-Rao, B.: Front-End Development in React: An Overview. *Engineering International* 7(2), 117-126 (2019).
6. Stamelos, I., Lefteris, A., Apostolos, O., et al.: Code Quality Analysis in Open Source Software Development. *Information Systems Journal (Oxford, England)* 12(1), 43-60 (2002).
7. Liu, N., Liu, Z.: The Application of User-centered Design Concept in Web Design. *Guangdong Technology* 22(08), 36-49 (2013).
8. Flavian, C., Gurrea, R. and Orús, C.: Web design: a key factor for the website success, *Journal of Systems and Information Technology* 11(2), 168-184 (2009).
9. Pei, X., Hong, M.: Development of Java EE+React Mini and Micro APP Program. *Computer Programming Skills and Maintenance* 24(18), 37-39 (2017)
10. Nivasalo, M.: Full stack development in TypeScript with tRPC and React Native. MSc. Thesis, Helsinki, Metropolia University of Applied Sciences (2023).
11. Zahran, D., Al-Nuaim, H., Rutter, M., et al.: A Comparative Approach to Web Evaluation and Website Evaluation Methods. *International Journal of Public Information Systems* 10(1), 20-39 (2014).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

