Design and Implementation of Music Website Based on Vue Framework Using AI Technologies

Yuan Ma

School of Computer and Software, Jincheng College, Sichuan University, Chengdu 611731, China

Abstract: ***This article briefly describes a music website designed and developed based on the Vue framework. The website refers to popular music websites and apps today, absorbs their advantages, and compensates for their disadvantages. It has a simple but personalized page, simple and convenient operation, complete functions, and high practicality, with the aim of facilitating listeners. This project is developed using the Vue.js framework and is a simple and easy-to-use music website. On this website, users can register, log in, modify personal information, listen to music, watch music videos, and collect their favorite songs, providing users with a comfortable experience.***

Keywords: Front end development; Vue; Website Development.

1. INTRODUCTION

In today's rapidly developing era of the internet, numerous music websites have emerged. However, many of these music websites are often profit driven, and their interface and functional designs often have many flaws that cannot meet the needs of more users. In order to optimize these flaws, facilitate more users, improve user experience, and make more people fall in love with music, it is necessary to create a high-quality and pure music environment for users. While making it convenient for users to operate, they can also enjoy more high-quality songs and music videos. This website provides users with functions such as registration and login, liking and collecting, searching, filtering, listening to music, and watching music videos. This article will introduce the website from the perspectives of main technology selection, feasibility analysis, requirement analysis, design and implementation of various functional modules.

2. INTRODUCTION TO MAIN TECHNOLOGY SELECTION

2.1 Vue.js

Vue.js is a progressive architecture for user interfaces. Compared with other frameworks, Vue adopts a bottom-up incremental development mode. Vue is relatively lightweight, simple, and direct, making it more user-friendly. With HTML, CSS, and JavaScript, it can be quickly mastered; Vue is an MVVM framework that aims to achieve corresponding data binding through a simple API; The core library of Vue is the view layer, which is relatively easy to learn and integrate with other libraries or projects; Vue component-based development can reduce repetitive code writing, improve development efficiency, and simplify debugging steps. Recent advancements in artificial intelligence and machine learning have significantly influenced various domains, including image processing, education, healthcare, and logistics. Yan et al. (2024) proposed a novel mechanism for image super-resolution reconstruction utilizing convolutional neural networks, demonstrating its efficacy in enhancing image quality through deep learning techniques [1]. In the field of education, Long et al. (2024) introduced a transformer-based model combined with InfoNCE loss to improve educational content matching, showcasing its potential in personalized learning systems [2]. Healthcare has also seen innovative applications, such as the work by Huang et al. (2024), who developed a federated learning-based system for multi-agency collaboration in medical image analysis and classification, addressing privacy concerns while improving diagnostic accuracy [3]. Additionally, Li (2025) explored the optimization of clinical trial strategies for anti-HER2 drugs by integrating Bayesian optimization with deep learning, offering a data-driven approach to accelerate drug development [4]. In logistics, Wang and Li (2024) employed long short-term memory neural networks to dynamically optimize transportation networks, highlighting the role of AI in improving supply chain efficiency [5]. Lastly, Tang et al. (2024) conducted a qualitative analysis of regional housing supply and demand imbalances in the US using big data, providing insights into urban planning and policy-making [6]. These studies collectively underscore the transformative impact of AI and machine learning across diverse sectors, paving the way for future innovations.

3. REQUIREMENT ANALYSIS

In contemporary society, music has become an indispensable part of people's daily lives, and there are more "musicians" who specialize in playing music in society. Music can help people relax, feel happy, improve sleep quality, and increase work efficiency. It can be said that music is subtly changing people's lives, and today's music lovers are also a relatively large collective, covering all age groups. They are constantly listening to music and immersed in the wonderful world of music. However, with so many music websites today, people are unable to make the right choices. More websites are profit oriented, with more advertising and marketing, reducing users' visual experience and not meeting the requirements for user experience. At this time, it is necessary to... User centered, with concise pages, Easy to operate, it can cater to the vast majority of music lovers. Therefore, in order to meet the needs of users, the website has set up more practical functions. In addition, the interface design also adopts popular colors and layouts today. The requirements can be divided into: user registration/login, which can eliminate the need for users to worry about their playlists or likes getting stuck on the same computer. When logged in from another device, your playlists will be synchronized to this computer; User information management to ensure the security of your personal information; Search, allowing you to quickly find the songs you want to listen to in a complex playlist to save time; Like and save, in order to find your favorite songs more conveniently and quickly next time; Music playback; MV playback, to play your favorite songs and MVs, etc.

This website is mainly divided into user registration/login, user information management, liking and collecting, search, music playback, MV playback and other modules. In order to provide a simple interface and operation, and to provide a better user experience, users can only like and collect songs after logging in. Songs and MVs can be played without the need for user login.

4. MODULE DESIGN

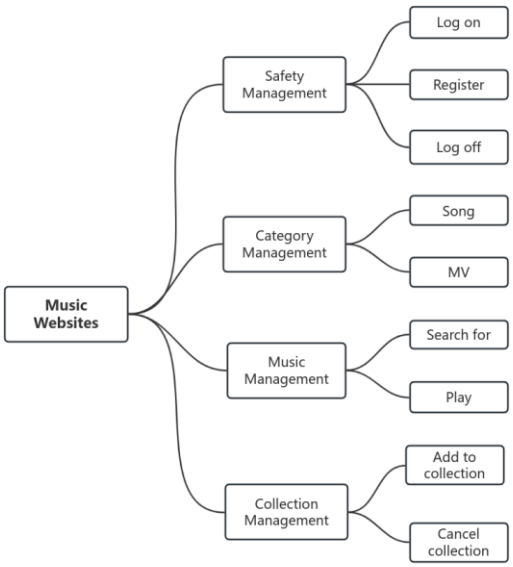

Figure 1: Module Design Diagram

4.1 Function Description

4.1.1 User login/registration/logout login

Registration:

(1) Users can select the login and registration buttons on the login page. The registration function is designed for new users, while existing users can log in directly.

(2) Users need to set their own personal username and password for registration, and the username must be composed of numbers, letters, and underscores, and cannot be repeated. If repeated, words such as "This username already exists, please start anew" will appear. Considering account security, the password cannot be too simple and must be six or more characters. If the username or password does not meet the specified requirements, words such as "Registration failed, please check the correctness of the username or password!" will be displayed to prompt the user that the input is incorrect.

Sign in:

After successful user registration, simply enter the set username and password, click the login button below. If the input information is incorrect or the user has not registered, the message "Login failed, please check the correctness of username or password!" will be displayed. If the input is correct, the message "Login successful" will be displayed and the user will be redirected to their personal homepage. And the corresponding username information will also be passed to the personal homepage after logging in through variables.

Log out:
Exiting login is a function that occurs after a user successfully logs into their personal page. This feature is designed for logged in users. The implementation of the login function is mainly achieved by deleting the user's data from the array of user information, and then redirecting the page to the logged in page on the interface. The specific function of logging out is achieved by setting the value of the login judgment variable, whether it is true or false, to determine whether it is in a login state or an logout state.

4.1.2 Favorite/Cancel Favorite

(1) This feature is not open to users who have not registered or logged in, and is only available to those who have successfully logged in.

(2) After the user logs in successfully, there will be a favorites button below the songs in the music hall interface. Clicking on it will give you likes and favorites. Favorite songs can be viewed on the "My Music" personal homepage. Click the like and save button below the song in the personal homepage playlist to cancel the collection.

(3) The implementation of this feature sets a click event, which triggers the unlike () function to perform conditional judgment. If the like button is clicked, the song data will also be saved to the like song data on the homepage and displayed on the personal information page.

4.1.3 Search

(1) At the top of each song page and MV interface, there is a search box where users can enter the song they want to listen to, click on the MV they want to watch or the singer they like to search, and the playlist or MV below can display the search results. Delete the search text in the input box and click elsewhere to make the input box lose focus and cancel the search.

(2) This function binds a change event to the input box. After the user clicks the search button, the toSearch() function will be called. This function can first determine whether the input box is empty. If it is empty, the function will be skipped and the search function will not be executed. If it is not empty, the function will search for data that matches the data in the input box and display it.

4.1.4 Music playback

(1) There is a playlist in the song interface, and each song has its corresponding area to store the corresponding information of the song, including singer, release time, song time, etc. Clicking on any position in the song area can play the song. At the same time, the corresponding player for the song will appear at the bottom of the interface.

(2) The player includes song images, start/pause buttons, playback bar, volume adjustment button, and download button.

(3) The data is obtained through v-for traversal, and the corresponding cells of the song are bound with click events. The data is transmitted through each click to play the corresponding song, and the Boolean value is switched to change the pause state of the song.

4.1.5 MV playback

(1) There is a similar introduction area in the MV interface and song, which includes MV images, singers, release time, etc. Clicking on any position in this area will pop up a playback box on this page to play the MV. Click on the top right corner to close the playback box.

(2) The data is obtained through v-for traversal, and the obtained data is displayed in the area according to the template. Clicking on play triggers the play () function, which displays the hidden playback box and retrieves mp4 data locally to display in the playback box. If no data is obtained, the play() function will not be triggered, and the MV cannot be played. The MV playback method defines a variable to place the path. Each time the play button is clicked, the function is triggered to update the value of the variable, and then the path is passed to src through the variable to achieve MV playback.
(3) Clicking the close MV button will trigger the close() function, which will pause the playback of the MV and hide the playback box, thereby stopping the playback of the MV.

(4) The key codes for MV playback function are as follows:

Close () {let video = document.get Elements By Class Name (& apos; video & apos;) [0] video.style.display = & apos;none & apos;}, play(value) {let video = document.get Elements By Class Name (& apos; video')[0] video.style.display = & apos; block & apos; this.mv Ur l = require(`../ assets / mv / $ {value.url}.mp4`)}.

4.1.6 Filtering function

(1) At the top of the song page, there is a filtering section for the initial letters (A~Z, #), language (Mandarin, Japanese, Korean, European and American), singer gender (male, female, group), and song type (pop, rap, rock, electronic, folk, light music, jazz, other). Clicking can perform single-layer or multi-layer filtering, and after filtering, the corresponding filtered songs will appear in the playlist.

(2) This function retrieves the current cell content by clicking on the event, triggers the check() function, sets a new array, and if the data matches the clicked content, loads the data into the new array and reloads the table to achieve filtering function.

(3) Cancelling filtering is resetting the data.

(4) The key code for the filtering function is as follows:

Checkone (e) {console.log(e.current Target.inner Text) let arr = [] this.table Data.forEach(function (item) {if (e.current Target.inner Text = item. language) {arr.push(item) // e.current Target.style. Color = » # 613560 »; }}) console.log(arr) this.table Data = arr}.

5. CONCLUSION

This article briefly describes a music website designed and implemented using Vue.js. The website focuses on user experience and aims to be convenient for users. The interface is simple and clear, and the operation is easy and convenient, suitable for all age groups. Currently, the website is still in the development stage, and there are still some loopholes and shortcomings that need to be improved. In the subsequent development process, interactive functions such as user comments on songs and MVs will be added to attract more users or singers to this website, allowing the website to have private message communication and interaction between users and singers like mainstream music websites, increasing the fun of the website. In addition, the website will also obtain the most commonly listened to song types for users, create a daily recommended playlist for users, and recommend more songs that the user likes. Type of music. In the subsequent user experience process, we will use methods such as survey questionnaires and street interviews to collect user opinions or suggestions, continuously update and upgrade the website, and strive to create a music website that belongs to everyone.

REFERENCES

[1] Yan, H., Wang, Z., Xu, Z., Wang, Z., Wu, Z., & Lyu, R. (2024, July). Research on image super-resolution reconstruction mechanism based on convolutional neural network. In Proceedings of the 2024 4th International Conference on Artificial Intelligence, Automation and High Performance Computing (pp. 142-146).

[2] Long, Y., Gu, D., Li, X., Lu, P., & Cao, J. (2024, September). Enhancing Educational Content Matching Using Transformer Models and InfoNCE Loss. In 2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE) (pp. 11-15). IEEE.

[3] Li, T. (2025). Optimization of Clinical Trial Strategies for Anti-HER2 Drugs Based on Bayesian Optimization and Deep Learning.

[4] Huang, S., Diao, S., Wan, Y., & Song, C. (2024, August). Research on multi-agency collaboration medical images analysis and classification system based on federated learning. In Proceedings of the 2024 International Conference on Biomedicine and Intelligent Technology (pp. 40-44).

[5] Wang, J., & Li, X. (2024). Dynamic Optimization of Transportation Networks in Logistics Using Long Short-Term Memory Neural Networks.

[6] Tang, Y., Zhao, S., & Yanjun, C. (2024). Regional Housing Supply and Demand Imbalance Qualitative Analysis in US based on Big Data.