**Software Engineering Program**

**Sino-British Collaborative Education**

**Chengdu University of Technology – Oxford Brooks University**

**Learning Management System**

**Final Report**

|  |
| --- |
| **Student number:** |
| **Date:** June 2nd , 2024 |
| **Group：L5C3** |

Table of Contents:

[Table of Contents: 2](#_Toc167274749)

[1) Introduction 3](#_Toc167274750)

[2) Accessibility measures: 4](#_Toc167274751)

[a) Perceivable 4](#_Toc167274752)

[Additional accessibility features 5](#_Toc167274753)

[3) Mobile Phone User Adaptation: 6](#_Toc167274754)

[4. Test Plans 7](#_Toc167274755)

[5. Framework Implementation: 12](#_Toc167274758)

[6. Conclusion 14](#_Toc167274759)

[7. References: 15](#_Toc167274760)

1. Introduction

With the increasing size of the school, the number of students has increased dramatically, and the amount of information about students has also multiplied. In the face of huge amount of information, it is necessary to have a student information management system to improve the efficiency of student management. Through such a system can achieve standardized management of information, scientific statistics and fast query, modify, increase, delete, so as to reduce the workload of management.

This system is mainly applied to the school student information management, the overall task is to realize the student information relationship systematization, standardization and automation, its main task is to calculate the student information for daily management, such as query, modify, add, delete, in addition to taking into account the student course selection, and the teacher needs to grade the assessments and so on. According to these requirements, a student management system is designed.

The development of student information management system mainly includes the establishment and maintenance of background database and the development of front-end applications. For the former, it is required to establish a database consistency and integrity, good security database. For the latter, the application is required to be fully functional and easy to use.

1. Accessibility measures:

There are 25 ways to improve accessibility( Lembree,D.,2016). Eleven have been completed in this project. The first is consistent layout and structure. The banner, navigation and navigation should appear in the same place throughout the site. They should also be labelled identically, such as using the same heading structure. The second is to use page titles to provide a consistent structure for HTML documents. The third is the correct use of headings, confusion without headings and skipping titles. The fourth is that linked content still makes sense out of context.

The fourth is to use the title attribute sparingly. The fifth is to highlight links (visually) as they gain focus. This indicates to the eye keyboard user where they are on the page. The sixth is to make progressive enhancement (PE) more feasible by separating content, style (or presentation) and behaviour through the use of HTML, CSS and JavaScript. Progressive enhancement allows websites to elegantly degrade over older browsers and technologies. The seventh is the list using the list element. Number eight is not using only colour to pass colour. The ninth is the correctly labelled data table. Number 10 is the use of easy-to-understand language for cognitively impaired and non-native speakers. Number eleven is to use a variety of methods to test the usability of your website.

The site is designed to provide a platform for system administrators, instructors and students who can improve the student information management of the school and facilitate the daily information management of instructors. It can not be accepted by the majority of disabled people, but users with mobility difficulties, cognitive impairment, colour blindness and other users can use it

# Perceivable

The first is to add alternative text to images,The Alt attribute of the image element provides a literal description of the image. Alt text can be used to access the "content" of images when they are not visible, such as blind users, search engines, disabled images, and broken links.

The second is that you can skip links. For example, a "skip" link is a special type of anchor link that moves the cursor from one area of the page to another. It is very beneficial to keyboard users, allowing them to skip chunks of content and links.

The third is to inform the user if a link is not opened in the default way, that is, outside of a web page in the same browser window. Includes links to open in a new window, links to non-HTML files such as PDF and video files. One or more methods should also be used depending on the type of disability and browser support.

The fourth is to keep the underline, mainly because such links are still visible to color-blind users.

The fifth is to make the Web form accessible by labeling the field. Most sections are already marked, so simply append the label element to the associated text. Give the field a unique ID and reference that ID through the for attribute of the label element.

# Additional accessibility features

The first is that many people don't use a mouse. Some assistive technologies rely on keyboard input or virtual keyboards rather than mouse or pointer devices. For this reason, links should not be required to be mouse-only. To comply with the standards, the link should be an A element with an href attribute.

The second is to add ARIA Landmark Roles. Barrier-free Rich Internet Applications (ARIA) is a relatively new W3C specification that provides a way to make modern applications accessible to assistive technologies

The third is to add enough color contrast to ensure benefits for low vision and color blind users. The ninth is to use ARIA Live Regions to alert users to dynamic changes in content on a web page.

1. Mobile Phone User Adaptation:

I will use Google’s Mobile-Friendly Test tool. It requires your site’s URL or code to run the test.

First of all, ensure that the viewport meta tag is included in the HTML head section with width=device-width, initial-scale=1.0. This sets the width of the page to follow the screen width of the device and sets the initial zoom level.

Second, adjusting the width of elements to fit smaller screens.

Third, enlarging clickable areas for links and buttons to accommodate touch input.

Fourth, adjusting font sizes and line heights for readability on smaller screens.

Fifth, use percentage-based widths for containers and columns instead of fixed pixel widths. This allows them to adapt to different screen sizes.

Sixth, optimize images for mobile devices by using responsive images or CSS techniques like max-width: 100%; to ensure they scale down appropriately without losing quality or affecting page load times.

Seventh, replace hover-based interactions with touch-friendly equivalents. For example, use dropdown menus that expand on tap instead of hover.

1. Test Plans

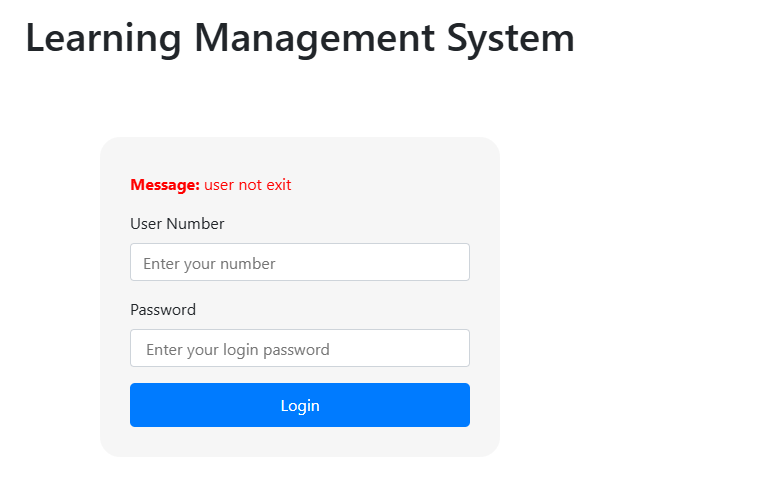
This should include a formal test plan and results as covered on previous programming and software engineering, and any fixes you made. This should include testing both the functionality and accessibility of your site. You should also test that your site works when multiple sessions interact with it at once, by using multiple browsers at once on your computer, you can use screenshots how you tested al functionalities, database, …

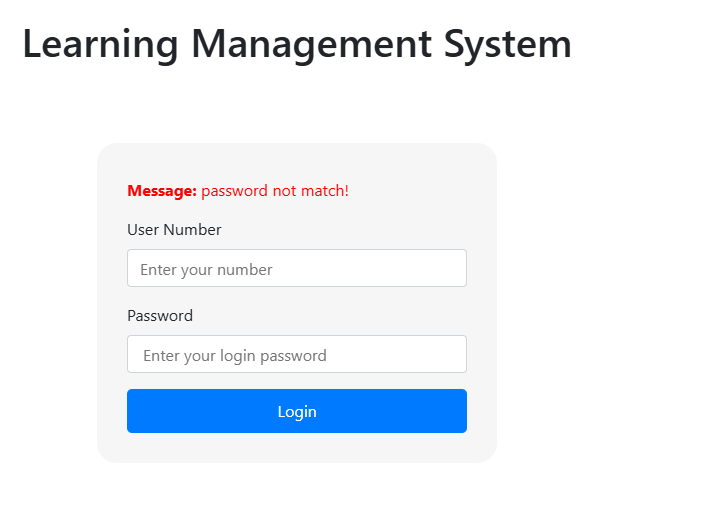
(example)

# login

To log in, users need to enter a user name and password that exists in the database.

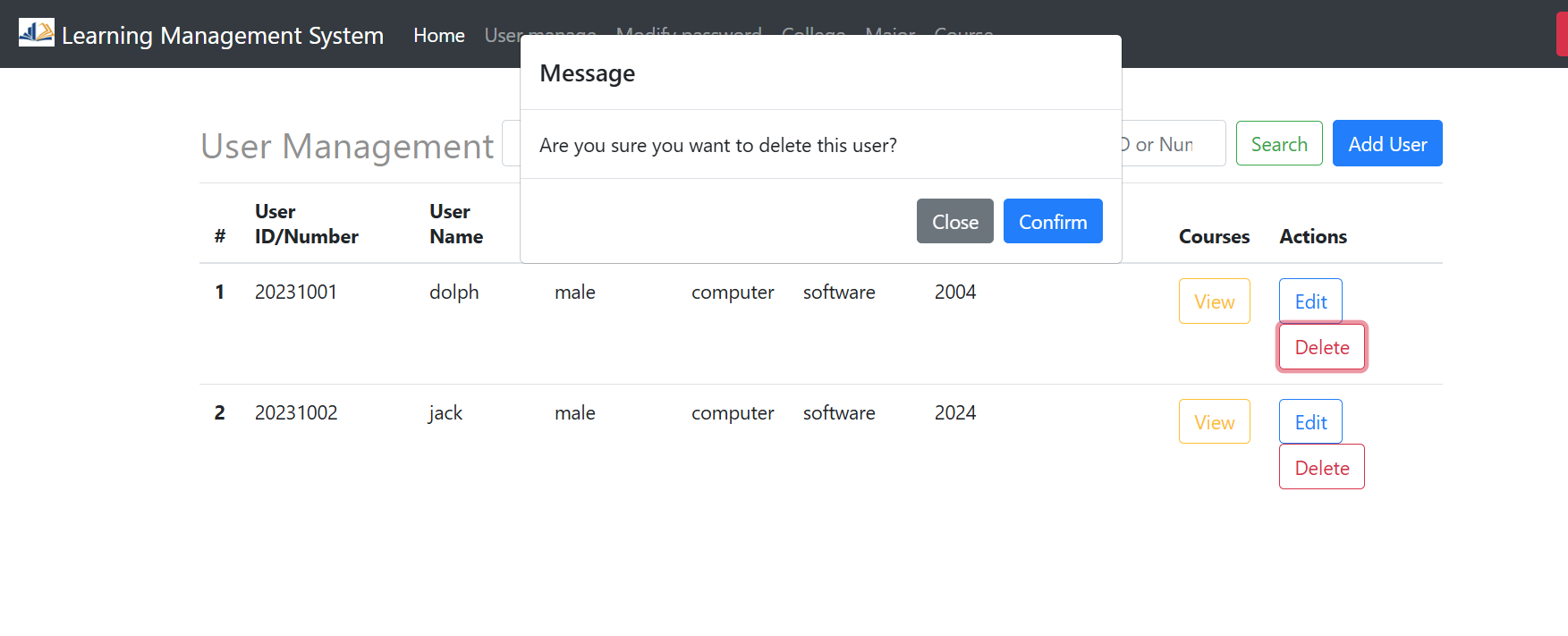
If input password or username not in the database,an error message is displayed.



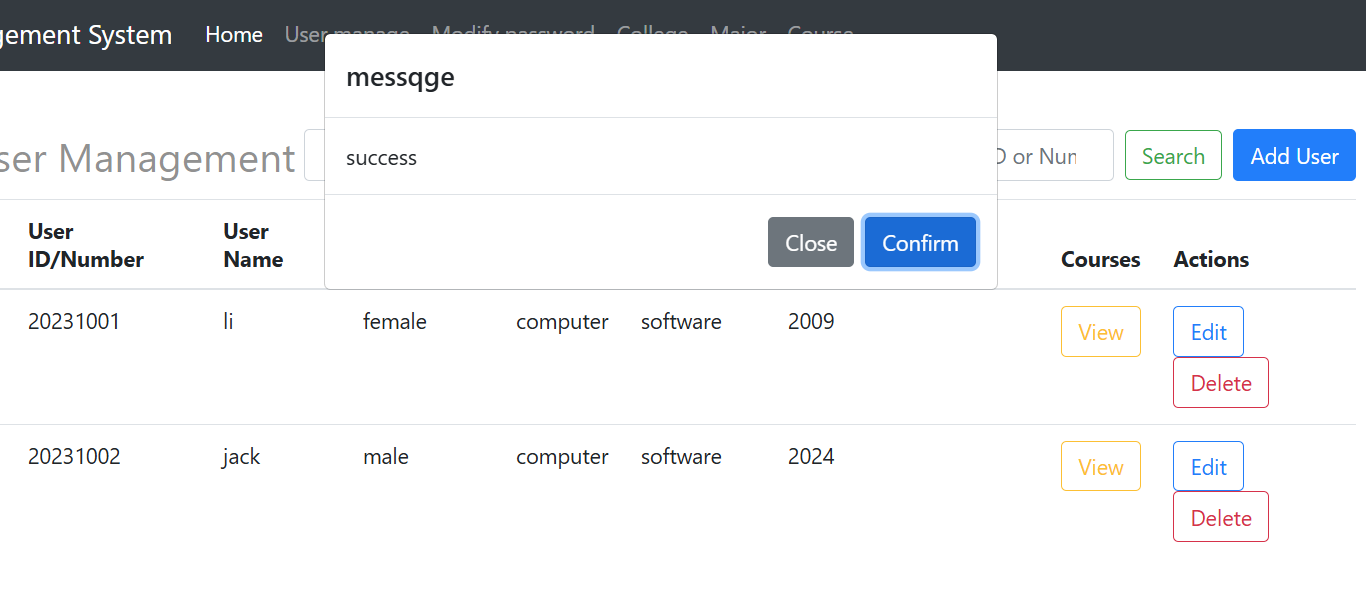


# delete student

if manager want to delete user

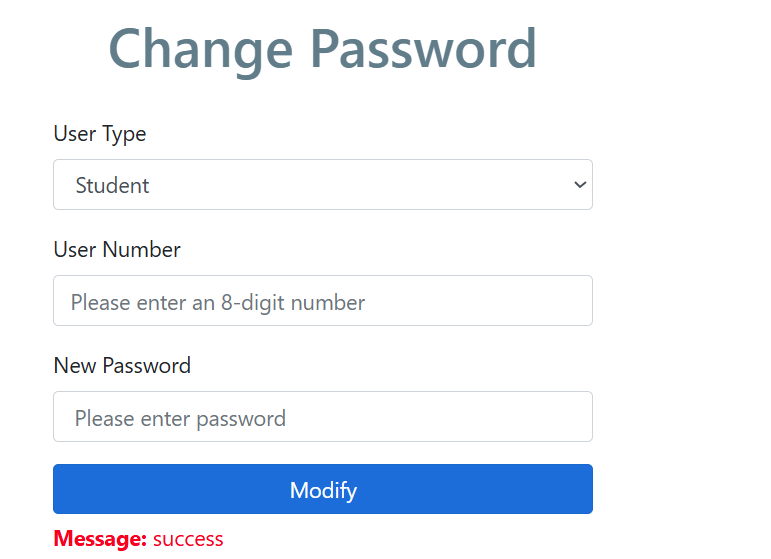


And the return true



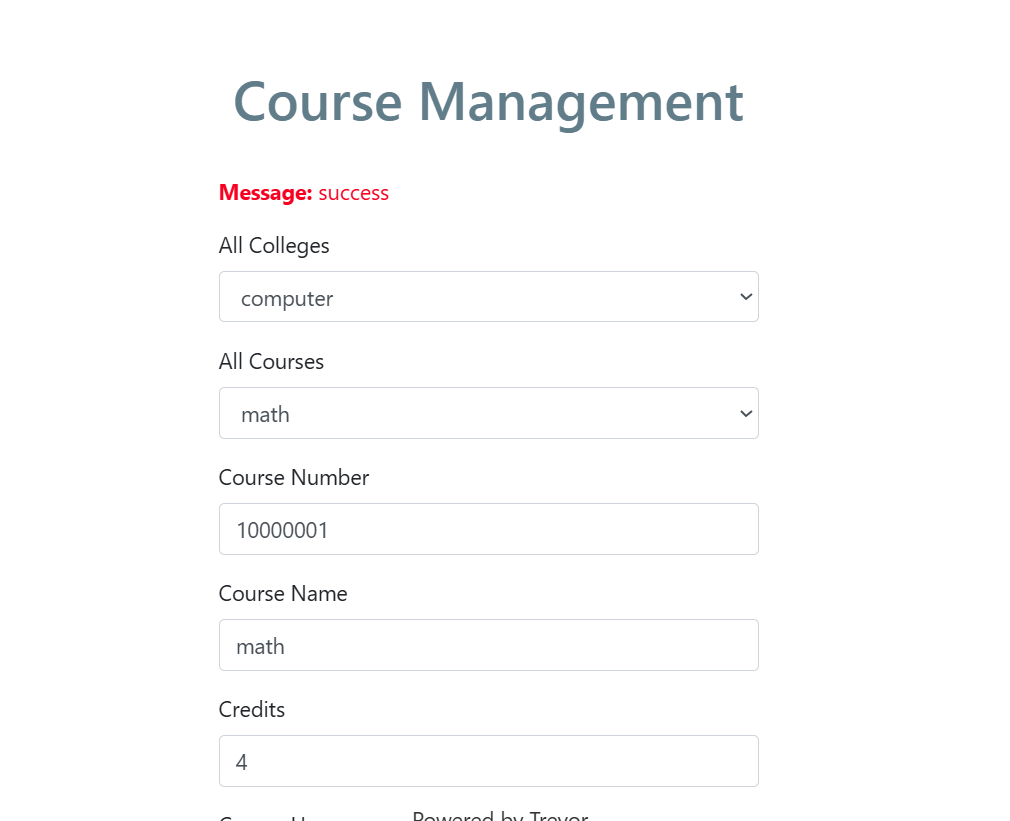
# change password

if you want to change user password, put the new password. It will return success.



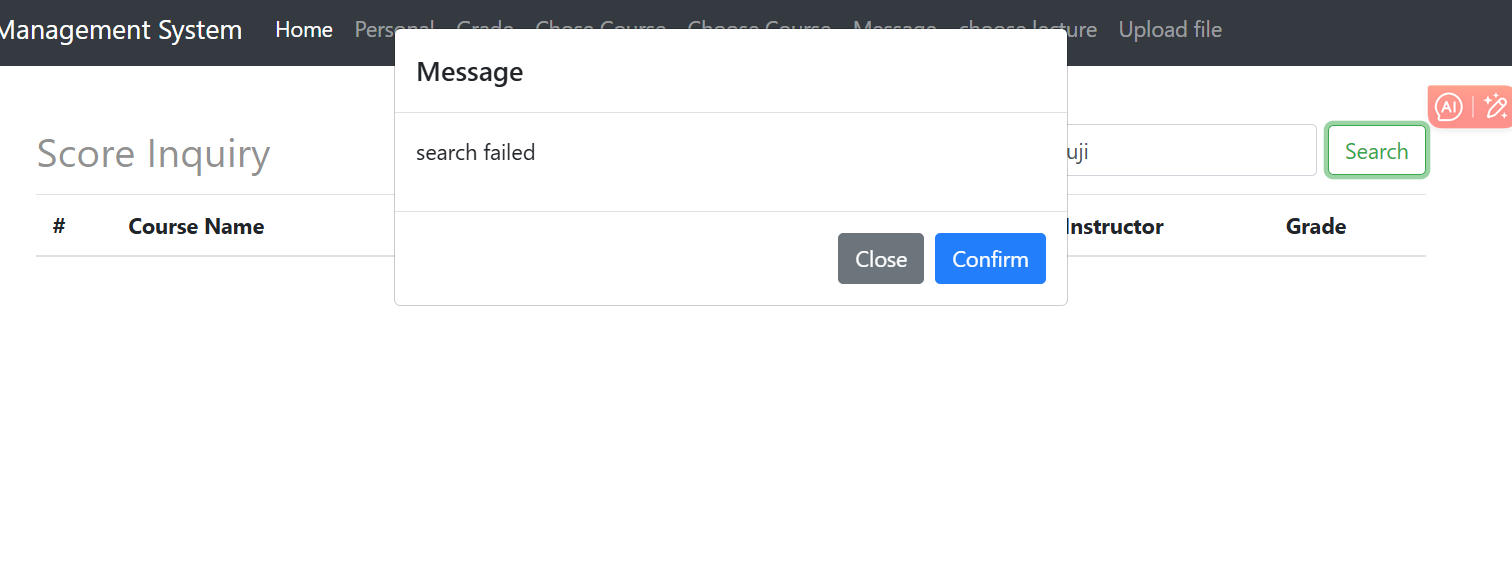
# course manage

if you want to edit the course information, and can execute, return true



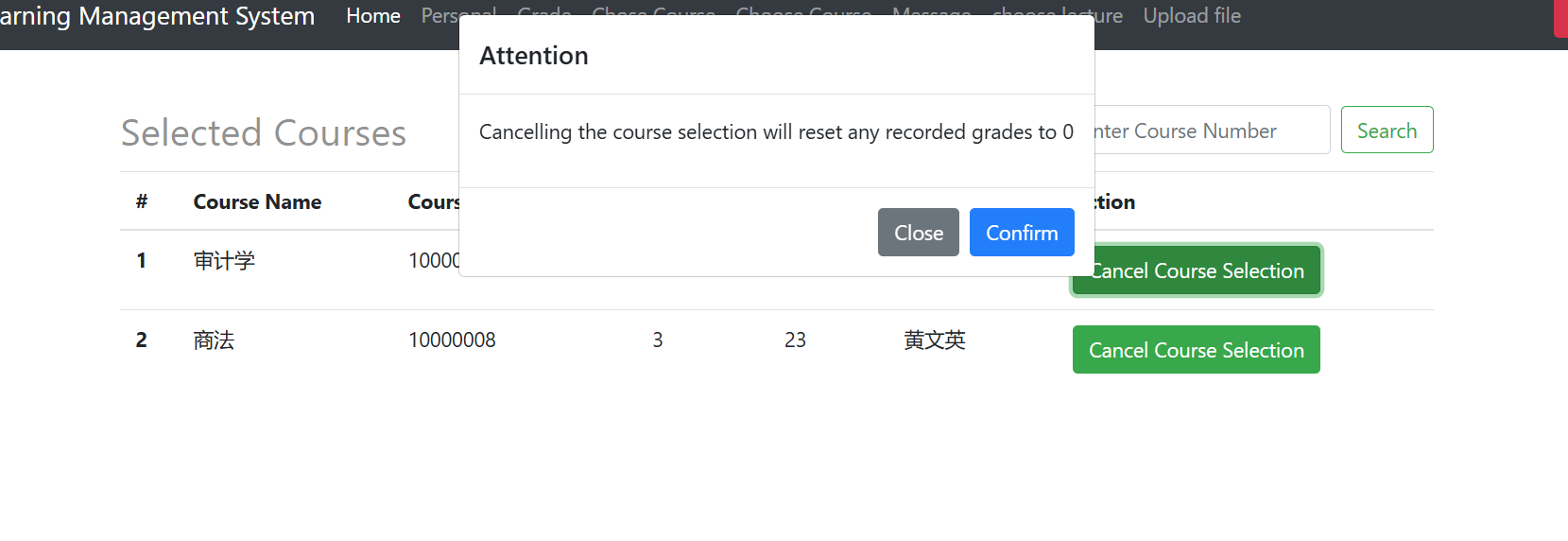
# search course

if you want to search course and type wrong name. it will remind you.



# cancel course

if you want to cancel course, it will remind you do not do this



1. Framework Implementation:

Choose one of the major JavaScript frameworks such as ReactJS, Bootstrap or Angular, and research it only. Describe how you would modify your site to work within this framework and what difference it would make to implementation of your site. Note that your description must be specific to your site - a general description of the framework will score no marks.

1. I'll choose ReactJS as the JavaScript framework for modifying a hypothetical website. ReactJS is a popular choice for building user interfaces, especially in single-page applications.

2. Initialize a new React project (if not already set up) using Create React App or another starter template.

Identify reusable visual elements across the site (like buttons, input forms, navigation bar) and convert them into React components.

const Button = ({ onClick, children, className }) => (

<button onClick={onClick} className={`btn ${className}`}>{children}</button>

);

And implement it in login button.

<Button onClick={handleLogin} className="primary">Login</Button>

3. Using react-router-dom, a robust library for handling routing in React applications:

Define paths that correspond to different components in your application, such as the homepage.

import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';

import Home from './Home';

const App = () => (

<Router>

<Switch>

<Route path="/" exact component={Home} />

</Switch>

</Router>

);

4. Local State Management (useState): Handle UI state like input values, toggle states

const [email, setEmail] = useState('');

const handleEmailChange = event => setEmail(event.target.value);

1. Global State Management (useContext, Redux): Manage data across multiple components, like user authentication

const AuthContext = React.createContext(null);

return <AuthContext.Provider value={{ isAuthenticated, user }}>{pesonal}</AuthContext.Provider>;

1. Use fetch with useEffect to handle data fetching in homepage:

useEffect(() => {

axios.get('/api/student/hoempage').then(response => setUser(response.data));

}, []);

1. Conclusion

This report focuses on Accessibility measures, Mobile phone user adaptation, Test for both of Functionality and accessibility and JavaScript Framework Implementation introduce the existing functions of this project. Accessibility and adaptability and room for improvement. In the future, this project can focus on mobile phone and continue to improve compatibility functions, fix existing bugs, and realize the expectation of compatibility between web pages and mobile phones.

1. References:

Dennis E. Lembree. (2016)25 Ways To Make Your Website Accessible. Available at: https://www.webhostingsearch.com/articles/25-ways-to-make-your-site-more-accessible.php. (Accessed: 1 june 2024).