Міністерство освіти і науки України

|  |
| --- |
| Український державний університет науки і технологій |
| Навчально-науковий інститут  «Український державний хіміко-технологічний університет» |

Факультет комп’ютерних наук та інженерії

Кафедра інформаційних систем

Лабораторна робота №2

З дисципліни Веб дизайн

Тема: «Відображення даних на сторінці»

Виконала:

студент гр. 3-КН-22

Дар’я ДОСОВА

Дніпро, 2024

Мета роботи: додати до існуючої розмітки авторизації можливість отримання та відображення даних з серверу ресурсів.

Практичне завдання

2.2.1 Вивчіть теоретичний і лекційний матеріал до даної теми;

2.2.2 Додайте на веб сторінку можливість виведення даних з серверу

ресурсів після процесу авторизації;

2.2.3 На сервері створіть метод, який очікує від сторінки запит на

завантаження даних з серверу ресурсів;

2.2.4 При отриманні запиту від UI передайте дані на існуючий сервер

ресурсів, щоб відобразити їх;

2.2.5 Створіть звіт про виконану роботу, в який необхідно включити

програмний код, який запускає веб додаток, а також скріншот з браузера до

і після виконання запиту.

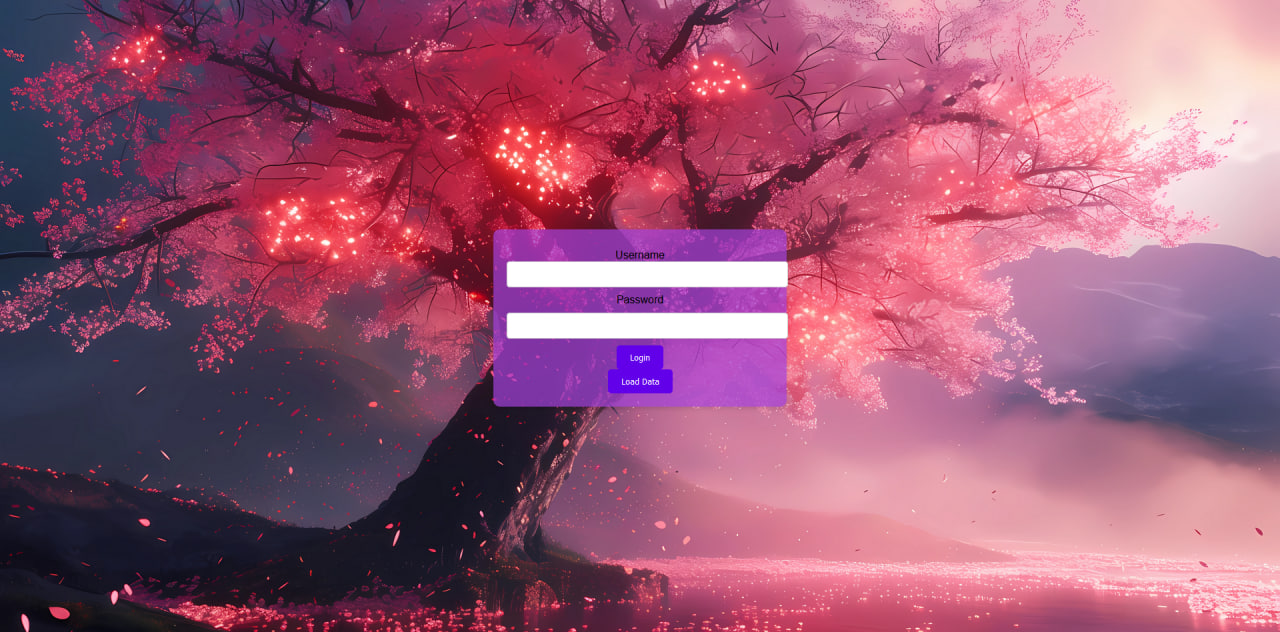
Код main.js:  
  
var JWT = "";  
  
// Компонент для авторизації  
Vue.component('auth-bar', {  
 template: `  
 <div>  
 <div class="field">  
 <label class="label">Username</label>  
 <div class="control">  
 <input type="login" class="input" v-model="login" />  
 </div>  
 </div>  
 <div>  
 <label class="label">Password</label>  
 <div class="field has-addons">  
 <div class="control is-expanded">  
 <input type="password" class="input" v-model="password" />  
 </div>  
 </div>  
 </div>  
 <input type="button" class="button" value="Login" @click="auth" />  
 <div v-if="token" class="token-display">  
 <strong>JWT:</strong> {{ token }}  
 </div>  
 </div>`,  
 methods: {  
 auth: function () {  
 this.$http  
 .get('auth/authorisation?login=' + this.login + '&password=' + this.password)  
 .then((response) => {  
 JWT = response.bodyText;  
 this.token = JWT;  
 });  
 },  
 },  
 data() {  
 return {  
 login: '',  
 password: '',  
 token: '',  
 };  
 },  
});  
  
// Компонент для завантаження даних  
Vue.component('persons-info-bar', {  
 props: ['persons'],  
 template: `  
 <div>  
 <input type="button" value="Load Data" @click="load" />  
 <div v-for="person in persons" :key="person.id">  
 <div>  
 <i>({{ person.id }})</i>  
 {{ person.lastName }} {{ person.firstName }}  
 </div>  
 </div>  
 </div>`,  
 methods: {  
 load: function () {  
 if (!JWT) {  
 alert("Please log in to access data.");  
 return;  
 }  
  
 const headers = {  
 'Content-Type': 'application/json',  
 'Authorization': 'Bearer ' + JWT,  
 };  
  
 this.$http  
 .get('http://localhost:45000/classes/persons', { headers })  
 .then((response) => {  
 response.json().then((data) => {  
 this.persons.splice(0);  
 data.forEach((person) => this.persons.push(person));  
 });  
 });  
 },  
 },  
});  
  
// Створення додатку Vue  
new Vue({  
 el: '#app',  
 template: `  
 <div>  
 <auth-bar />  
 <persons-info-bar :persons="persons" />  
 </div>`,  
 data: {  
 persons: [],  
 },  
});

Код index.xml :

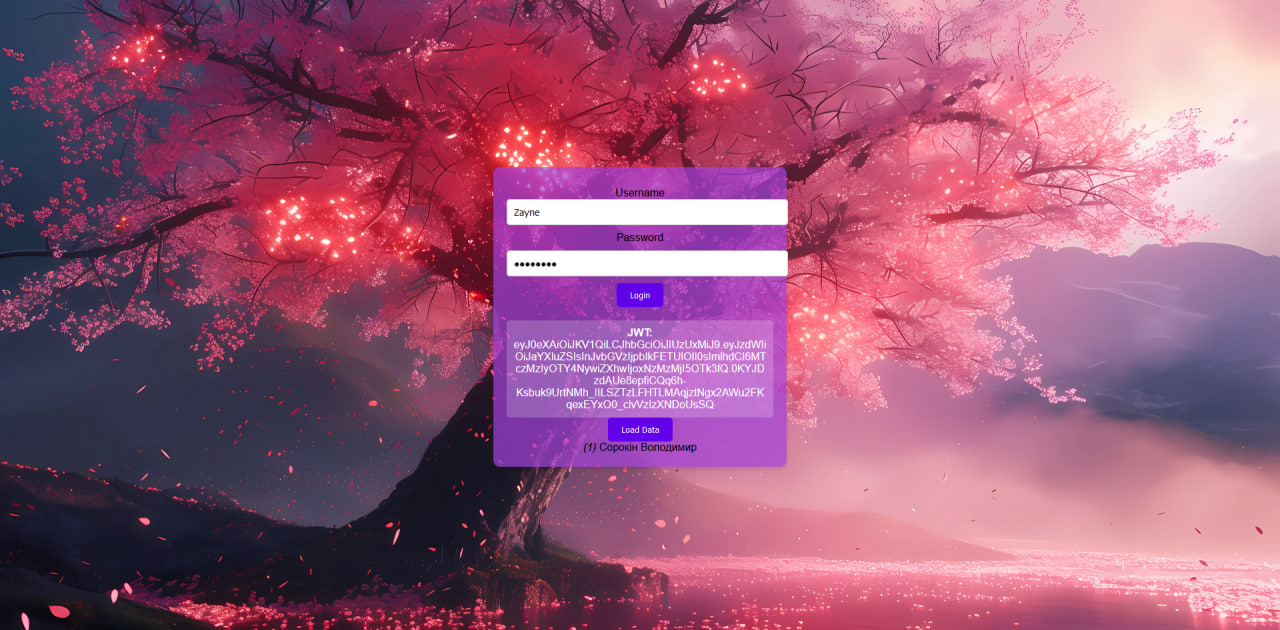
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>DB Editor</title>  
 <!-- Підключення бібліотек Vue.js -->  
 <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>  
 <script src="https://cdn.jsdelivr.net/npm/vue-resource@1.5.1"></script>  
 <style>  
 body {  
 background-image: url('https://images.alphacoders.com/135/1350953.png');  
 background-size: cover;  
 background-position: center;  
 background-repeat: no-repeat;  
 font-family: Arial, sans-serif;  
 margin: 0;  
 padding: 0;  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 height: 100vh;  
 overflow: hidden;  
 }  
  
 .container {  
 text-align: center;  
 height: auto;  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 }  
  
 div.text {  
 display: inline-block;  
 width: 400px;  
 background-color: rgba(164, 66, 220, 0.7);  
 text-align: center;  
 border-radius: 10px;  
 overflow-wrap: break-word;  
 padding: 20px;  
 box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
 }  
  
 #app {  
 font-size: 14px;  
 color: white;  
 }  
  
 .field {  
 margin: 10px 0;  
 }  
  
 .input {  
 width: 100%;  
 padding: 10px;  
 font-size: 14px;  
 border-radius: 5px;  
 border: 1px solid #ccc;  
 }  
  
 .button {  
 background-color: #6200ea;  
 color: white;  
 padding: 10px 20px;  
 border: none;  
 border-radius: 5px;  
 cursor: pointer;  
 }  
  
 .button:hover {  
 background-color: #3700b3;  
 }  
  
 .token-display {  
 margin-top: 20px;  
 padding: 10px;  
 background-color: rgba(255, 255, 255, 0.2);  
 border-radius: 5px;  
 overflow-wrap: break-word;  
 word-wrap: break-word;  
 color: white;  
 }  
 </style>  
</head>  
  
<body>  
<div class="container">  
 <div class="text">  
 <!-- Головний контейнер Vue -->  
 <div id="app"></div>  
 </div>  
</div>  
  
<!-- Підключення файлу main.js -->  
<script>  
 var JWT = "";  
  
 // Компонент для авторизації  
 Vue.component('auth-bar', {  
 template: `  
 <div>  
 <div class="field">  
 <label class="label">Username</label>  
 <div class="control">  
 <input type="login" class="input" v-model="login" />  
 </div>  
 </div>  
 <div>  
 <label class="label">Password</label>  
 <div class="field has-addons">  
 <div class="control is-expanded">  
 <input type="password" class="input" v-model="password" />  
 </div>  
 </div>  
 </div>  
 <input type="button" class="button" value="Login" @click="auth" />  
 <div v-if="token" class="token-display">  
 <strong>JWT:</strong> {{ token }}  
 </div>  
 </div>`,  
 methods: {  
 auth: function () {  
 this.$http  
 .get('auth/authorisation?login=' + this.login + '&password=' + this.password)  
 .then((response) => {  
 JWT = response.bodyText;  
 this.token = JWT;  
 });  
 },  
 },  
 data() {  
 return {  
 login: '',  
 password: '',  
 token: '',  
 };  
 },  
 });  
  
 // Компонент для завантаження даних  
 Vue.component('persons-info-bar', {  
 props: ['persons'],  
 template: `  
 <div>  
 <input type="button" class="button" value="Load Data" @click="load" />  
 <div v-for="person in persons" :key="person.id">  
 <div>  
 <i>({{ person.id }})</i>  
 {{ person.lastName }} {{ person.firstName }}  
 </div>  
 </div>  
 </div>`,  
 methods: {  
 load: function () {  
  
 if (!JWT) {  
 alert("Please log in to access data.");  
 return;  
 }  
  
  
 const headers = {  
 'Content-Type': 'application/json',  
 Authorization: 'Bearer ' + JWT,  
 };  
  
  
 this.$http  
 .get('http://localhost:45000/classes/persons', { headers })  
 .then((response) => {  
 response.json().then((data) => {  
 this.persons.splice(0);  
 data.forEach((person) => this.persons.push(person));  
 });  
 });  
  
 },  
 },  
 });  
  
 // Головний Vue екземпляр  
 new Vue({  
 el: '#app',  
 template: `  
 <div>  
 <auth-bar />  
 <persons-info-bar :persons="persons" />  
 </div>`,  
 data: {  
 persons: [],  
 },  
 });  
</script>  
</body>  
  
</html>

Результат виконання:

Вебсайт до логіну:



Після логіну :



Висновок:

У ході виконання роботи було додано можливість отримання та відображення даних із серверу ресурсів до існуючої системи авторизації. Це дозволяє забезпечити повноцінну взаємодію клієнта із захищеними ресурсами, підвищуючи функціональність та зручність використання системи.