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 №1

З дисципліни Веб дизайн

Тема: «Створення сторінки авторизації»

Виконала:

студент гр. 3-КН-22

Дар’я ДОСОВА

Дніпро, 2024

Мета роботи: створити розмітку сторінки для виконання авторизації.

Код роботи:

1)application.ym

server:  
 port: 45000

2) pom.xml

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.3.5.RELEASE</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
  
 <groupId>org.example</groupId>  
 <artifactId>LAb1</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>11</maven.compiler.source>  
 <maven.compiler.target>11</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 </dependencies>  
  
</project>

3) AuthRestController

package org.example.rest;  
  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.client.RestTemplate;  
  
@RestController  
@RequestMapping("auth")  
public class AuthRestController {  
 private static final String *AUTHORIZATION\_URL* =  
 "http://localhost:8004/authorisation";  
 @GetMapping("authorisation")  
 public String auth(  
 @RequestParam(value = "login", defaultValue = "")  
 String login,  
 @RequestParam(value = "password", defaultValue = "")  
 String password  
 ) {  
 RestTemplate restTemplate = new RestTemplate();  
 String local\_url = *AUTHORIZATION\_URL* + "?login=" + login  
 + "&password=" + password;  
 ResponseEntity<String> result =  
 restTemplate.getForEntity(local\_url, String.class);  
 return result.getBody();  
 }  
}

4) index.html

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <title>DB editor</title>  
 <!-- Підключення бібліотек vue →-->  
 <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-color: rgb(164, 66, 220);  
 font-family: Arial, sans-serif;  
 margin: 0;  
 padding: 0;  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 height: 100vh;  
 }  
  
 .container {  
 text-align: center; /\* Center the child div horizontally \*/  
 height: 100vh; /\* Full viewport height \*/  
 }  
  
 div.text {  
 display: inline-block;  
 width: 200px;  
 height: 200px;  
 background-color: lightgreen;  
 text-align: center;  
 }  
 </style>  
</head>  
  
<body>  
<div class="container">  
 <div class="text"><!-- Створення екземпляру додатку vue -->  
 <div id="app"></div>  
 <!-- Підключення файлу js з vue скриптами -->  
 <script src="js/main.js"></script>  
 </div>  
</div>  
</body>  
  
</html>

5) main.js

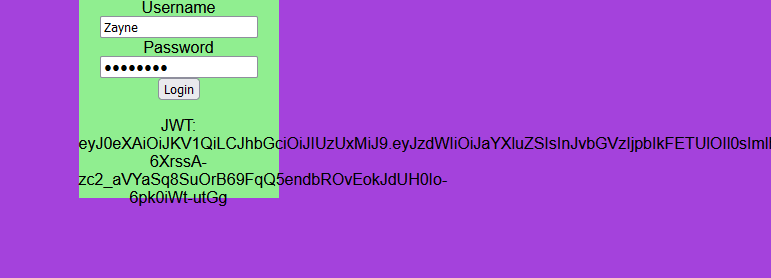
Vue.component('auth-bar', {  
 template:  
 '<div>'+  
 '<div class="field">'+  
 '<label class="label">Username</label>'+  
 '<div class="control">'+  
 '<input type="text" class="input" v-model="login"/>'+  
 '</div>'+  
 '</div>'+  
 '<div>'+  
 '<label class="label">Password</label>'+  
 '<div class="control">'+  
 '<input type="password" class="input" v-model="password"/>'+  
 '</div>'+  
 '</div>'+  
 '<input type="button" value="Login" @click="authorize"/>'+  
 '<p>JWT: {{ JWT }}</p>'+  
 '</div>',  
 methods: {  
 authorize() {  
 this.$http.get('auth/authorisation?login=' + this.login + '&password=' + this.password)  
 .then(response => {  
 console.log('body: ' + response.bodyText);  
 this.JWT = response.bodyText;  
 });  
 }  
 },  
 data() {  
 return {  
 login: '',  
 password: '',  
 JWT: ''  
 };  
 }  
});  
  
var app = new Vue({  
 el: '#app',  
 template: '<auth-bar />'  
});

Результат виконання:

1)До виконання авторизації



2) Після виконання авторизації:



Висновок: Vue.js — це потужний фреймворк для створення інтерфейсів користувача, що використовує реактивну систему та ефективне управління DOM. Він підтримує шаблони на основі HTML, які компілюються в рендерингові функції віртуального DOM, оптимізуючи ре-рендеринг. Vue також надає можливість створювати переходи між елементами за допомогою CSS анімацій, JavaScript маніпуляцій або сторонніх бібліотек. Завдяки своїй реактивній системі, Vue спрощує керування станом і рендерингом компонентів, роблячи розробку інтуїтивною та ефективною.