

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп’ютерних систем**

**Лабораторна робота № 2**

з дисципліни **Web-дизайн**

Тема: «Розробка функціональності Web-додатка мовою Javascript»

Виконала: студентка ІІІ курсу

ФПМ групи КВ-12

Ус В. О.

Київ 2024

**Загальне завдання**: розробити функціональність для статичних сторінок Web-додатку першої лабораторної роботи із використанням шаблону MVC.

**Посилання на Github-репозиторій:**

<https://github.com/vickaus/web_lab2>

**Посилання на Github Pages:**

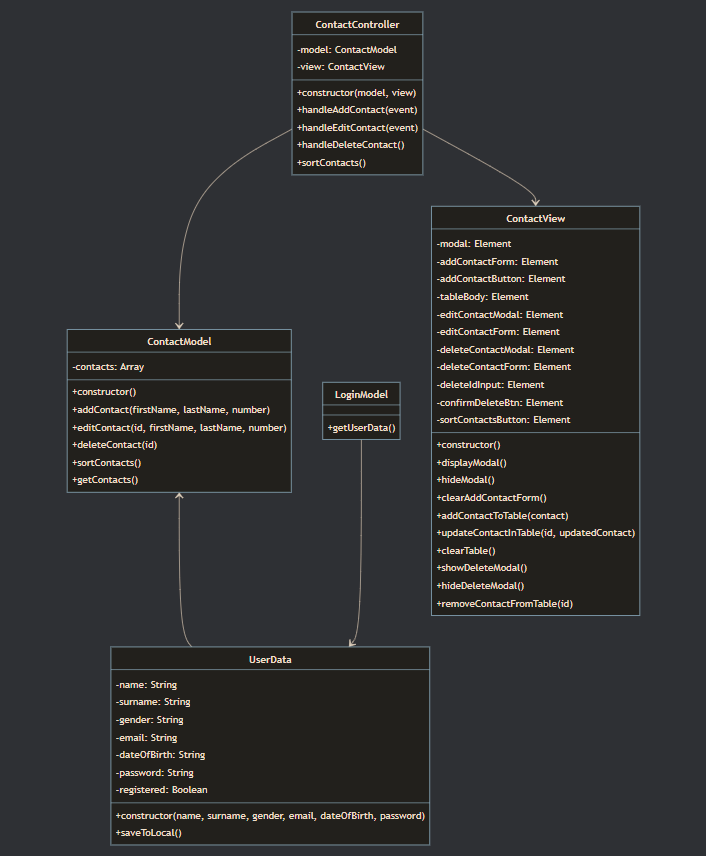
<https://vickaus.github.io/web_lab2/>

**Telegram:**

<https://t.me/vicka_us>

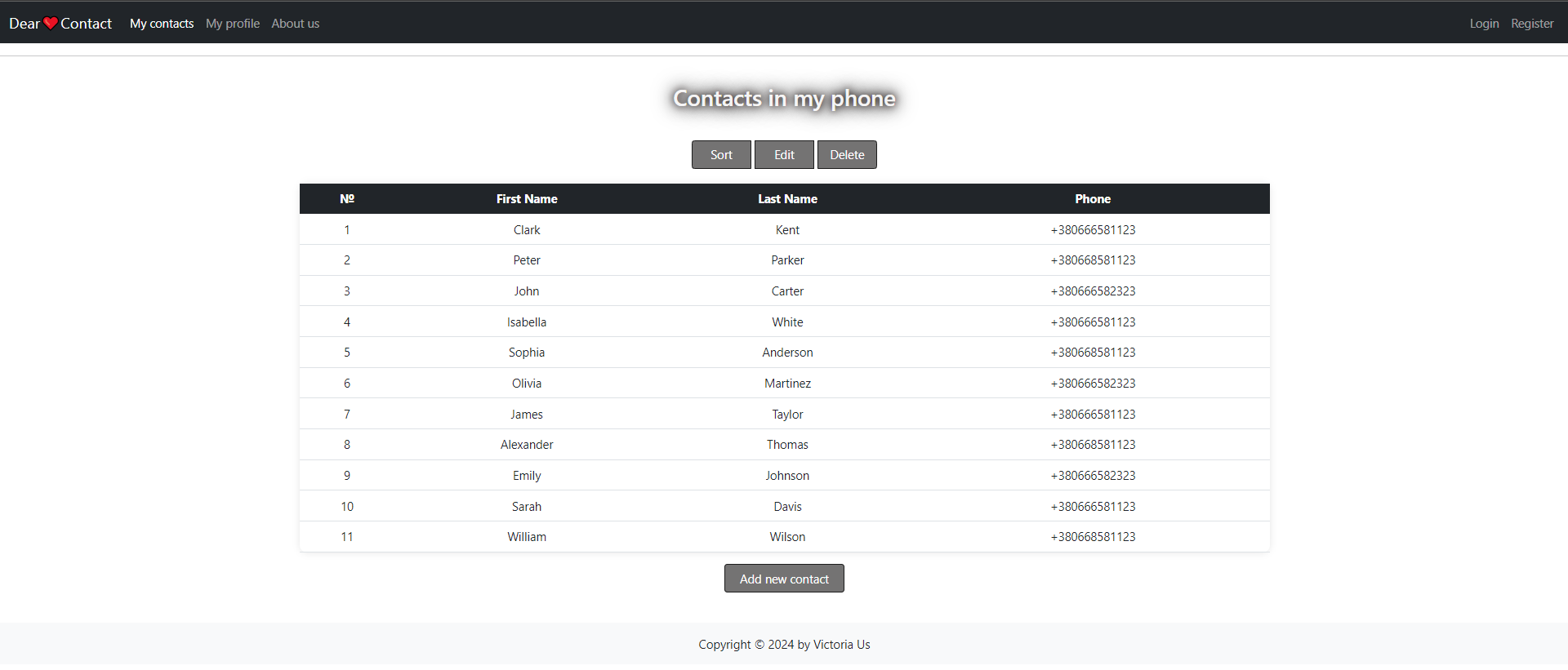
**Dear Contact:** додаток телефонної книги, в якому можна керувати своїми контактами. Завдяки легкому інтерфейсу можна легко переглядати інформацію. Також можна додати нові контакти із такою інформацією, як ім’я, прізвище та номер телефону. Можна редагувати, видалити і відсортувати наявні контакти у алфавітному порядку.

**Діаграма класів Javascript додатку**

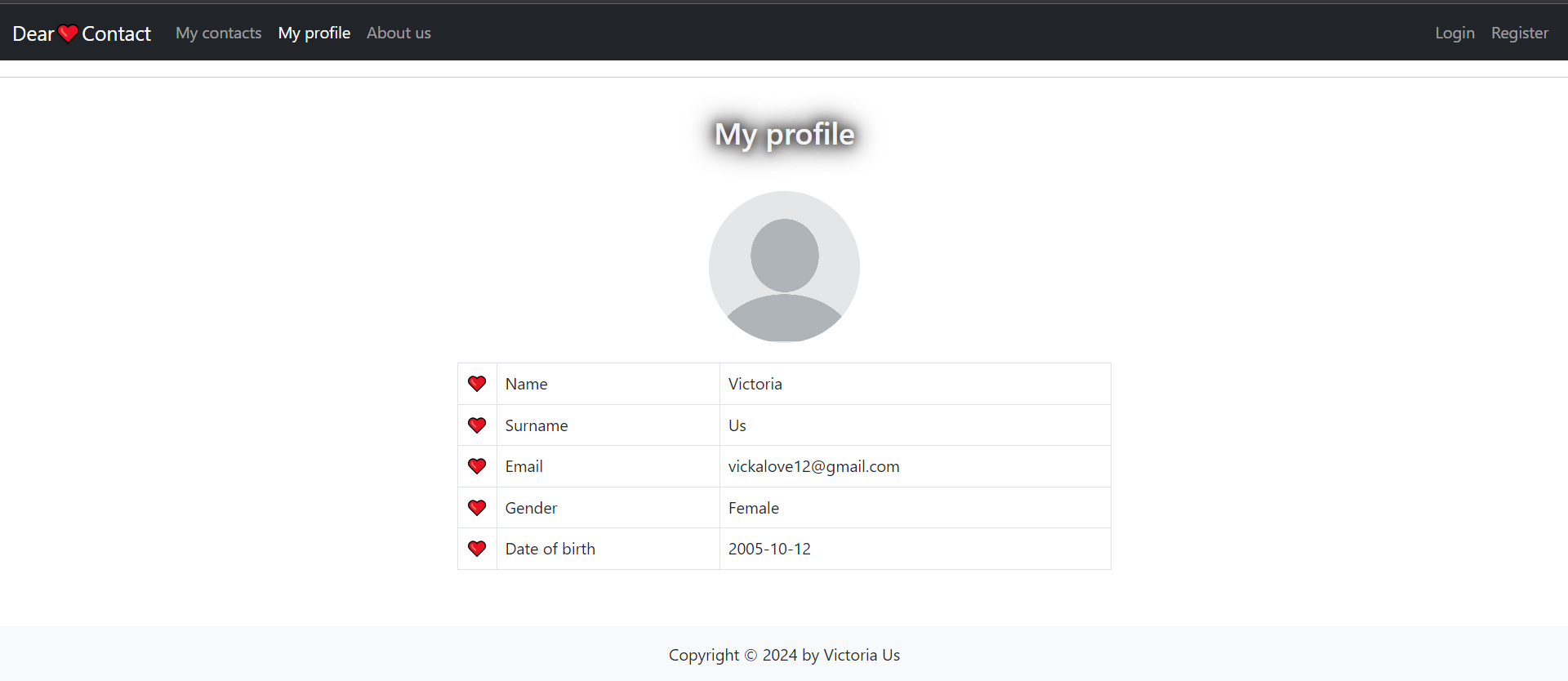


**Копії екрана усіх сторінок статичного Web-додатка**

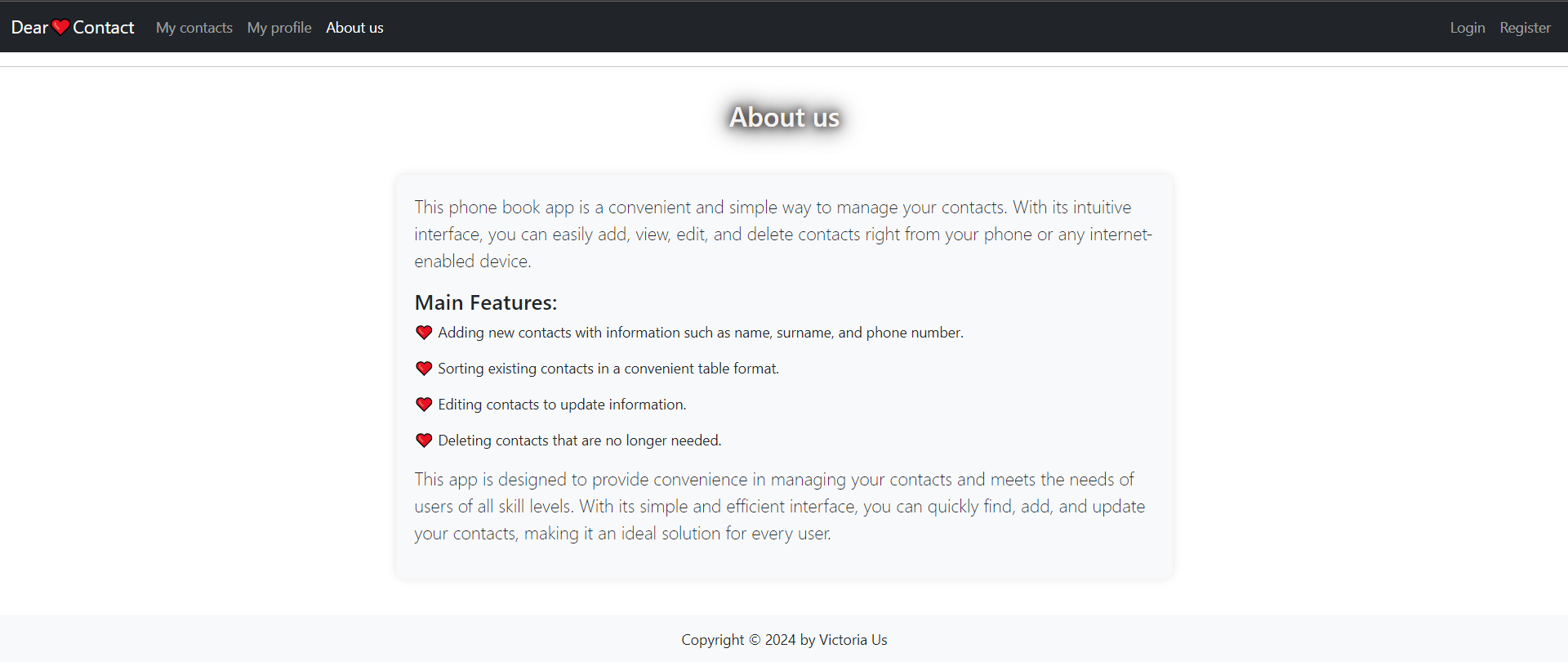
* **My contacts**

****

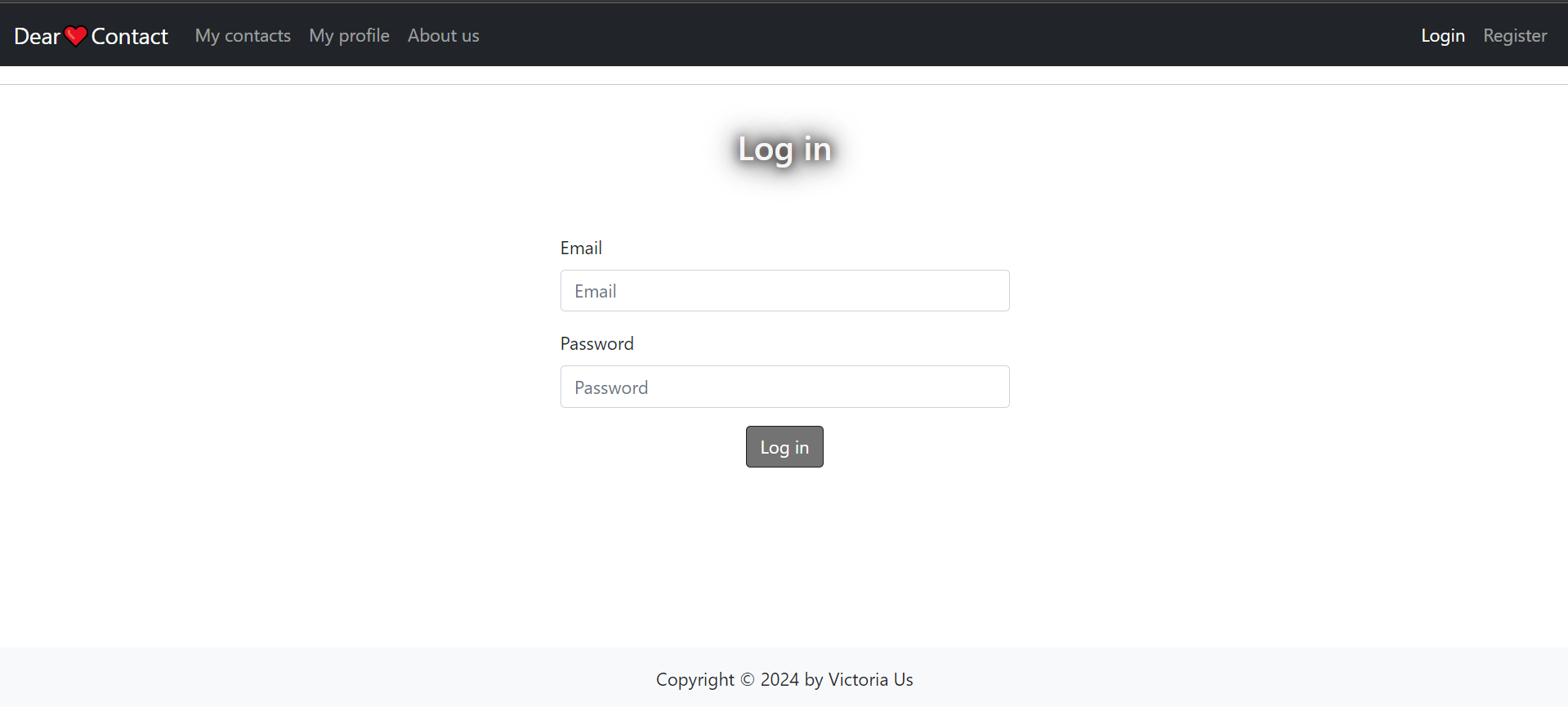
* **My profile**

****

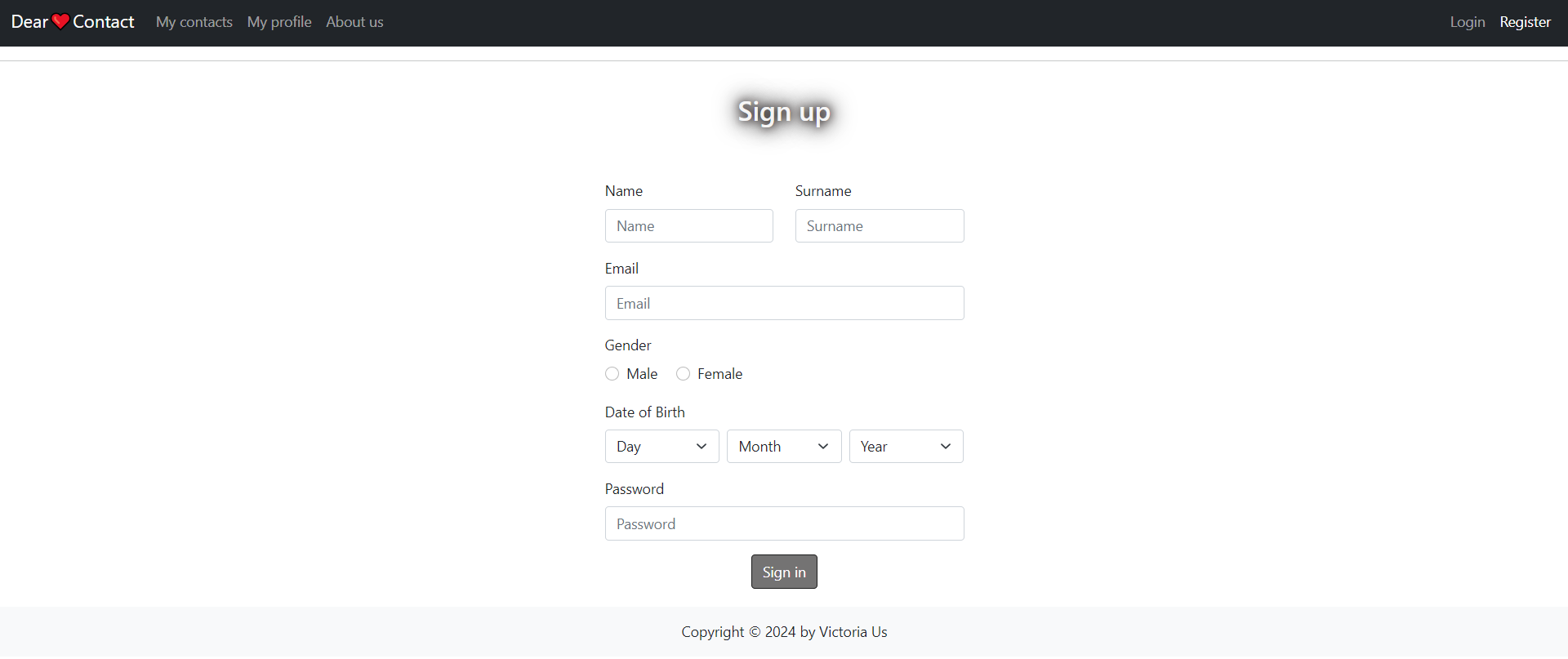
* **About us**

****

* **Login**

****

* **Register**

****

**Вміст Javascript-файлу**

**model.js**

**class** ContactModel {

constructor() {

**this**.contacts = [

{ id: 1, firstName: "Clark", lastName: "Kent", number: "+380666581123" },

{ id: 2, firstName: "Peter", lastName: "Parker", number: "+380668581123" },

{ id: 3, firstName: "John", lastName: "Carter", number: "+380666582323" },

{ id: 4, firstName: "Isabella", lastName: "White", number: "+380666581123" },

{ id: 5, firstName: "Sophia", lastName: "Anderson", number: "+380668581123" },

{ id: 6, firstName: "Olivia", lastName: "Martinez", number: "+380666582323" },

{ id: 7, firstName: "James", lastName: "Taylor", number: "+380666581123" },

{ id: 8, firstName: "Alexander", lastName: "Thomas", number: "+380668581123" },

{ id: 9, firstName: "Emily", lastName: "Johnson", number: "+380666582323" },

{ id: 10, firstName: "Sarah", lastName: "Davis", number: "+380666581123" },

{ id: 11, firstName: "William", lastName: "Wilson", number: "+380668581123" }

];

}

addContact(firstName, lastName, number) {

**const** id = **this**.contacts.**length** + 1;

**const** newContact = { id: id, firstName: firstName, lastName: lastName, number: number };

**this**.contacts.push(newContact);

}

editContact(id, firstName, lastName, number) {

**const** index = **this**.contacts.findIndex(contact => contact.id === id);

**if** (index !== -1) {

**this**.contacts[index].firstName = firstName;

**this**.contacts[index].lastName = lastName;

**this**.contacts[index].number = number;

}

}

deleteContact(id) {

**const** index = **this**.contacts.findIndex(contact => contact.id === id);

**if** (index !== -1) {

**this**.contacts.splice(index, 1);

}

}

sortContacts() {

**this**.contacts.**sort**((a, b) => a.firstName.localeCompare(b.firstName));

}

getContacts() {

**return** **this**.contacts;

}

}

**class** UserData {

constructor(name, surname, gender, email, dateOfBirth, password) {

**this**.name = name;

**this**.surname = surname;

**this**.gender = gender;

**this**.email = email;

**this**.dateOfBirth = dateOfBirth;

**this**.password = password;

**this**.registered = true;

}

saveToLocal() {

localStorage.setItem('userData', JSON.**stringify**(**this**));

}

}

**const** LoginModel = {

getUserData: **function**() {

**return** JSON.**parse**(localStorage.**getItem**('userData'));

}

};

**view.js**

**class** ContactView {

constructor() {

**this**.modal = document.getElementById('addContactModal');

**this**.addContactForm = document.getElementById('addContactForm');

**this**.addContactButton = document.querySelector('.btn-add button');

**this**.tableBody = document.querySelector('.table tbody');

**this**.editContactModal = document.getElementById('editContactModal');

**this**.editContactForm = document.getElementById('editContactForm');

**this**.deleteContactModal = document.getElementById('deleteContactModal');

**this**.deleteContactForm = document.getElementById('deleteContactForm');

**this**.deleteIdInput = document.getElementById('deleteId');

**this**.confirmDeleteBtn = document.getElementById('confirmDeleteBtn');

**this**.sortContactsButton = document.getElementById('sortContactsButton');

}

displayModal() {

**this**.modal.classList.add('show');

}

hideModal() {

**this**.modal.classList.remove('show');

}

clearAddContactForm() {

**this**.addContactForm.reset();

}

addContactToTable(contact) {

**const** newRow = document.createElement('tr');

newRow.innerHTML = `

<td>${contact.id}</td>

<td>${contact.firstName}</td>

<td>${contact.lastName}</td>

<td>${contact.number}</td>

`;

**this**.tableBody.appendChild(newRow);

}

updateContactInTable(id, updatedContact) {

**const** rows = **this**.tableBody.querySelectorAll('tr');

**const** rowToUpdate = Array.from(rows).find(row => row.querySelector('td:first-child').textContent == id);

**if** (rowToUpdate) {

rowToUpdate.children[1].textContent = updatedContact.firstName;

rowToUpdate.children[2].textContent = updatedContact.lastName;

rowToUpdate.children[3].textContent = updatedContact.number;

}

}

clearTable() {

**this**.tableBody.innerHTML = "";

}

showDeleteModal() {

**this**.deleteContactModal.classList.add('show');

}

hideDeleteModal() {

**this**.deleteContactModal.classList.remove('show');

}

removeContactFromTable(id) {

**const** rows = **this**.tableBody.querySelectorAll('tr');

**const** rowToDelete = Array.from(rows).find(row => row.querySelector('td:first-child').textContent == id);

**if** (rowToDelete) {

rowToDelete.remove();

} **else** {

**alert**('Row not found.');

}

}

}

document.addEventListener('DOMContentLoaded', () => {

**const** profileTable = document.querySelector('.table-bordered');

**const** storedUserData = localStorage.**getItem**('userData');

**if** (storedUserData) {

**const** userData = JSON.**parse**(storedUserData);

**const** nameCell = profileTable.querySelector('tr:nth-child(1) td:nth-child(3)');

**const** surnameCell = profileTable.querySelector('tr:nth-child(2) td:nth-child(3)');

**const** emailCell = profileTable.querySelector('tr:nth-child(3) td:nth-child(3)');

**const** genderCell = profileTable.querySelector('tr:nth-child(4) td:nth-child(3)');

**const** dateOfBirthCell = profileTable.querySelector('tr:nth-child(5) td:nth-child(3)');

nameCell.textContent = userData.name;

surnameCell.textContent = userData.surname;

emailCell.textContent = userData.email;

genderCell.textContent = userData.gender === 'male' ? 'Male' : 'Female';

dateOfBirthCell.textContent = userData.dateOfBirth;

} **else** {

console.**log**('User data not found. Please log in or sign up first.');

}

});

document.addEventListener('DOMContentLoaded', () => {

**const** registerForm = document.querySelector('.needs-validation');

**if** (registerForm) {

registerForm.addEventListener('submit', (event) => {

event.preventDefault();

event.stopPropagation();

**if** (registerForm.checkValidity()) {

**const** inputName = document.getElementById('inputName').value;

**const** inputSurname = document.getElementById('inputSurname').value;

**const** inputEmail = document.getElementById('inputEmail').value;

**const** inputPassword = document.getElementById('inputPassword').value;

**const** gender = document.querySelector('input[name="gender"]:checked');

**const** inputDay = document.querySelectorAll('select.form-select')[0].value;

**const** inputMonth = document.querySelectorAll('select.form-select')[1].value;

**const** inputYear = document.querySelectorAll('select.form-select')[2].value;

**if** (!gender) {

**alert**('Виберіть стать');

**return**;

}

**const** dateOfBirth = `${inputYear}-${inputMonth}-${inputDay}`;

**const** userData = **new** UserData(inputName, inputSurname, gender.id, inputEmail, dateOfBirth, inputPassword);

userData.saveToLocal();

**alert**('Акаунт успішно створено!');

window.location.href = 'contact.html';

}

registerForm.classList.add('was-validated');

});

}

**const** loginForm = document.querySelector('.login-form');

**if** (loginForm) {

loginForm.addEventListener('submit', (event) => {

event.preventDefault();

**const** inputEmail = document.getElementById('inputEmail').value;

**const** inputPassword = document.getElementById('inputPassword').value;

**const** storedUserData = localStorage.**getItem**('userData');

**if** (storedUserData) {

**const** userData = JSON.**parse**(storedUserData);

**if** (userData.registered && inputEmail === userData.email && inputPassword === userData.password) {

**alert**('Вхід в акаунт успішний!');

window.location.href = 'contact.html';

} **else** {

**alert**('Неправильний логін чи пароль. Спробуйте ще раз.');

}

} **else** {

**alert**('Користувач не знайдений, зареєструйтесь.');

}

loginForm.classList.add('was-validated');

});

}

});

**controller.js**

**class** ContactController {

constructor(model, view) {

**this**.model = model;

**this**.view = view;

**this**.view.addContactButton.addEventListener('click', () => **this**.view.displayModal());

**this**.view.addContactForm.addEventListener('submit', (event) => **this**.handleAddContact(event));

**this**.view.editContactForm.addEventListener('submit', (event) => **this**.handleEditContact(event));

**this**.view.confirmDeleteBtn.addEventListener('click', () => **this**.handleDeleteContact());

**this**.view.sortContactsButton.addEventListener('click', () => **this**.sortContacts());

}

handleAddContact(event) {

event.preventDefault();

**const** firstName = **this**.view.addContactForm.elements['inputFirstName'].value;

**const** lastName = **this**.view.addContactForm.elements['inputLastName'].value;

**const** number = **this**.view.addContactForm.elements['inputNumber'].value;

**this**.model.addContact(firstName, lastName, number);

**this**.view.addContactToTable({ id: **this**.model.contacts.**length**, firstName, lastName, number });

**this**.view.clearAddContactForm();

**this**.view.hideModal();

}

handleEditContact(event) {

event.preventDefault();

**const** id = **parseInt**(**this**.view.editContactForm.elements['editId'].value);

**const** firstName = **this**.view.editContactForm.elements['editFirstName'].value;

**const** lastName = **this**.view.editContactForm.elements['editLastName'].value;

**const** number = **this**.view.editContactForm.elements['editNumber'].value;

**this**.model.editContact(id, firstName, lastName, number);

**this**.view.updateContactInTable(id, { firstName, lastName, number });

**this**.view.hideEditModal();

}

handleDeleteContact() {

**const** id = **parseInt**(**this**.view.deleteIdInput.value);

**if** (!**isNaN**(id)) {

**this**.model.deleteContact(id);

**this**.view.removeContactFromTable(id);

**this**.view.hideDeleteModal();

} **else** {

**alert**('Please enter a valid contact ID.');

}

}

sortContacts() {

**this**.model.sortContacts();

**this**.view.clearTable();

**this**.model.getContacts().**forEach**(contact => **this**.view.addContactToTable(contact));

}

}

window.addEventListener('load', () => {

**const** model = **new** ContactModel();

**const** view = **new** ContactView();

**const** controller = **new** ContactController(model, view);

});