НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

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

**систем**

**Лабораторна робота №3**

з дисципліни

**«Web-дизайн»**

Тема: «Розробка Web-додатка за допомогою Javascript/VueJ»

Виконала: студентка III курсу

ФПМ групи КВ-03

Макаревич О. Ф.

Перевірив: Петрашенко А. В.

Київ – 2023

**Зміст**

**Мета:** ознайомитись із засобами фреймворка VueJS та навчитись створювати асинхронні запити до Web-сервера.

**Загальне завдання:** адаптувати програмний код ЛР№2 до вимог фреймворка VueJS та забезпечити завантаження необхідних даних з Web-сервера

**Інструменти розробки:** Javascript, VueJS, NodeJS, Express.

**Зауваження:** студент може обрати інший серверний фреймворк (наприклад, Django).

**Завдання**

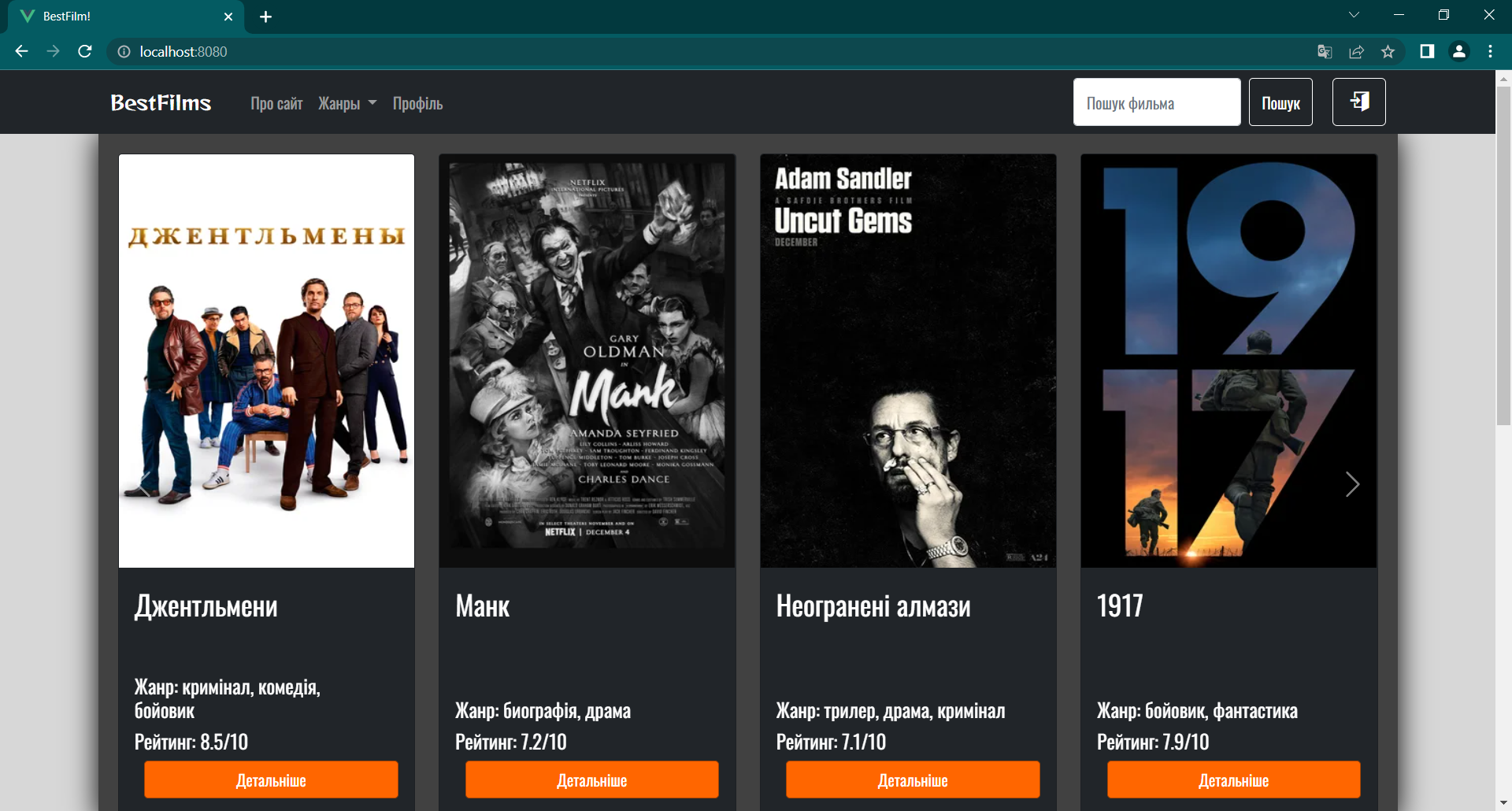
**Тема:  Адаптація лабораторної роботи №2 до вимог Vue. Cтворення веб-серверу за допомогою NodeJS + Express.**

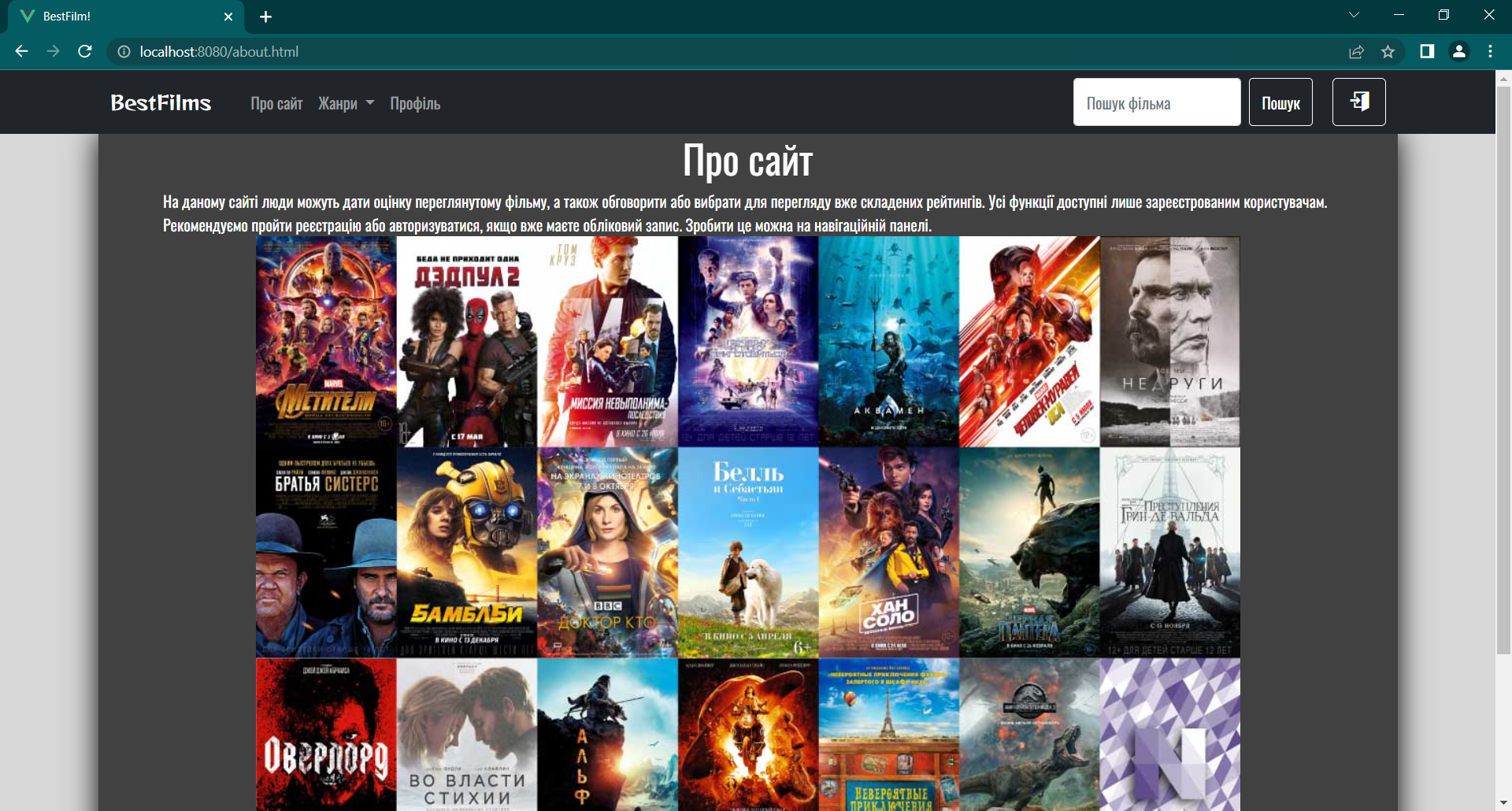
       У цій лабораторній роботі було створено додаток за допомогою фреймворка Vue. Було використано vue-router для навігації та vuex для взаємодії з данними.

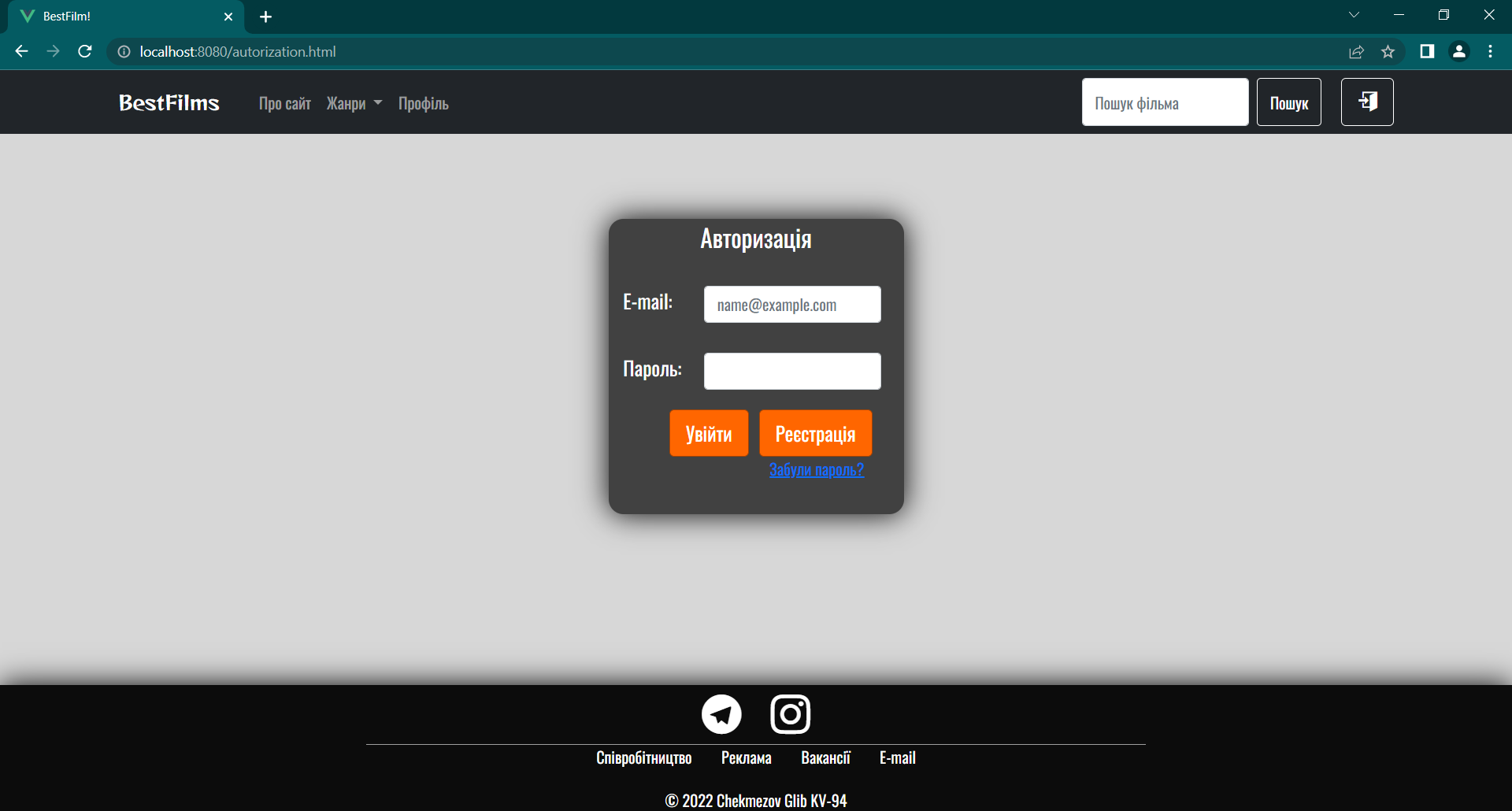
  Клієнт взаємодіє з сервером за допомогою бібліотеки fetchAPI.

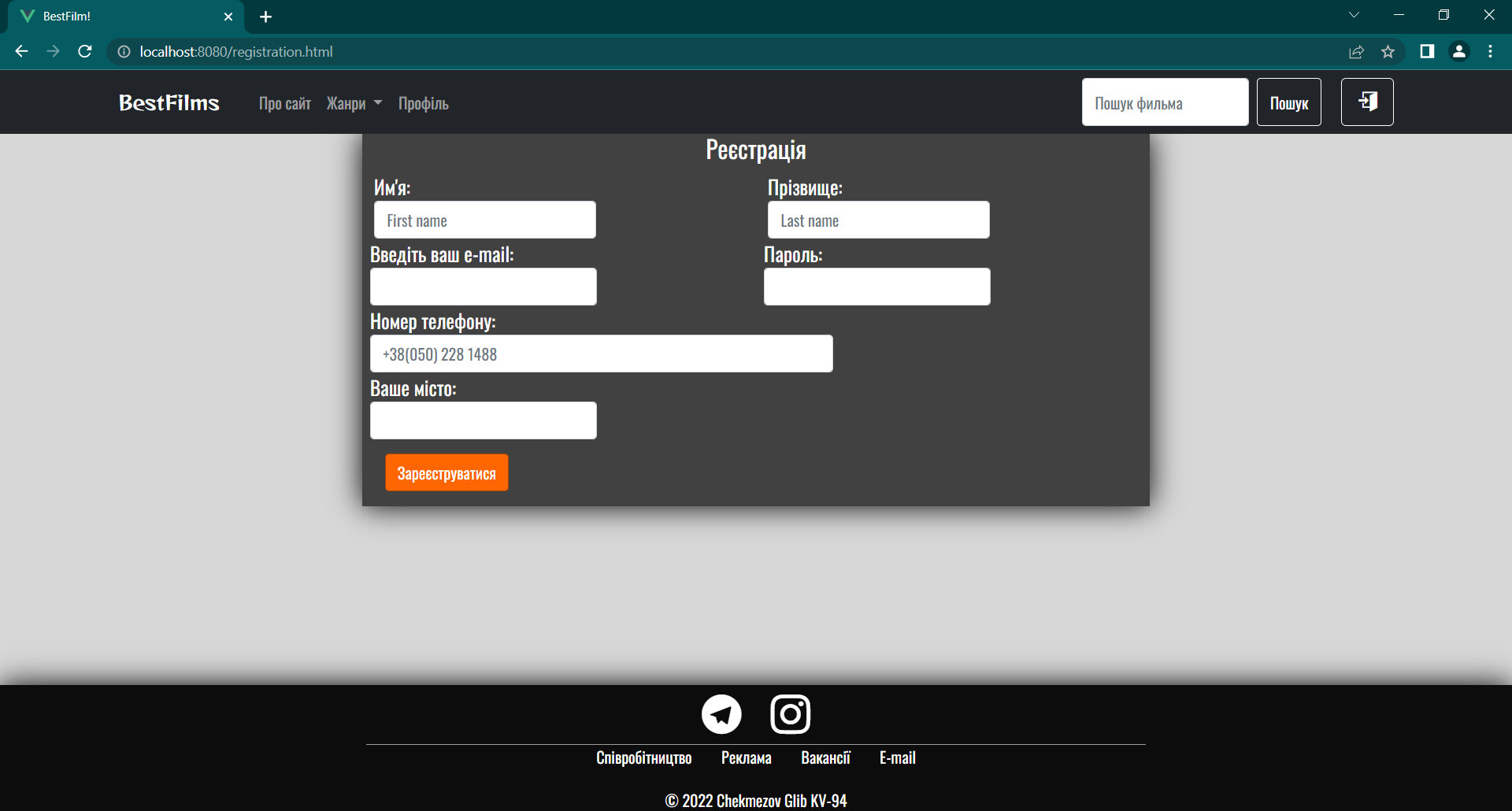
Сервер реалізовано за допомогою NodeJS + Express. Як сховище данних використовується SQLite3. Для більш простої взаємодії з базою данних використано ORM Sequelize.

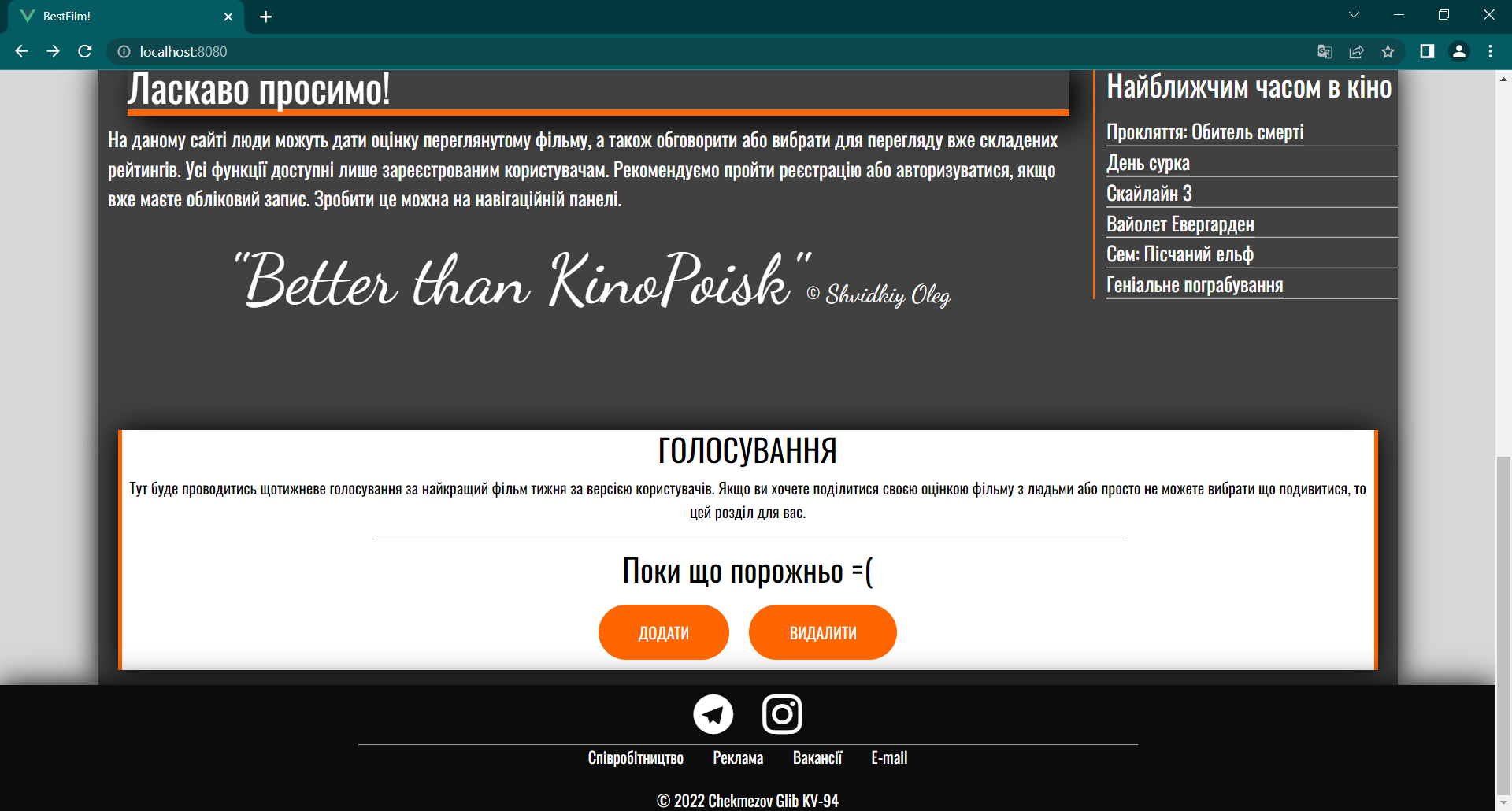
**Скріншоти**

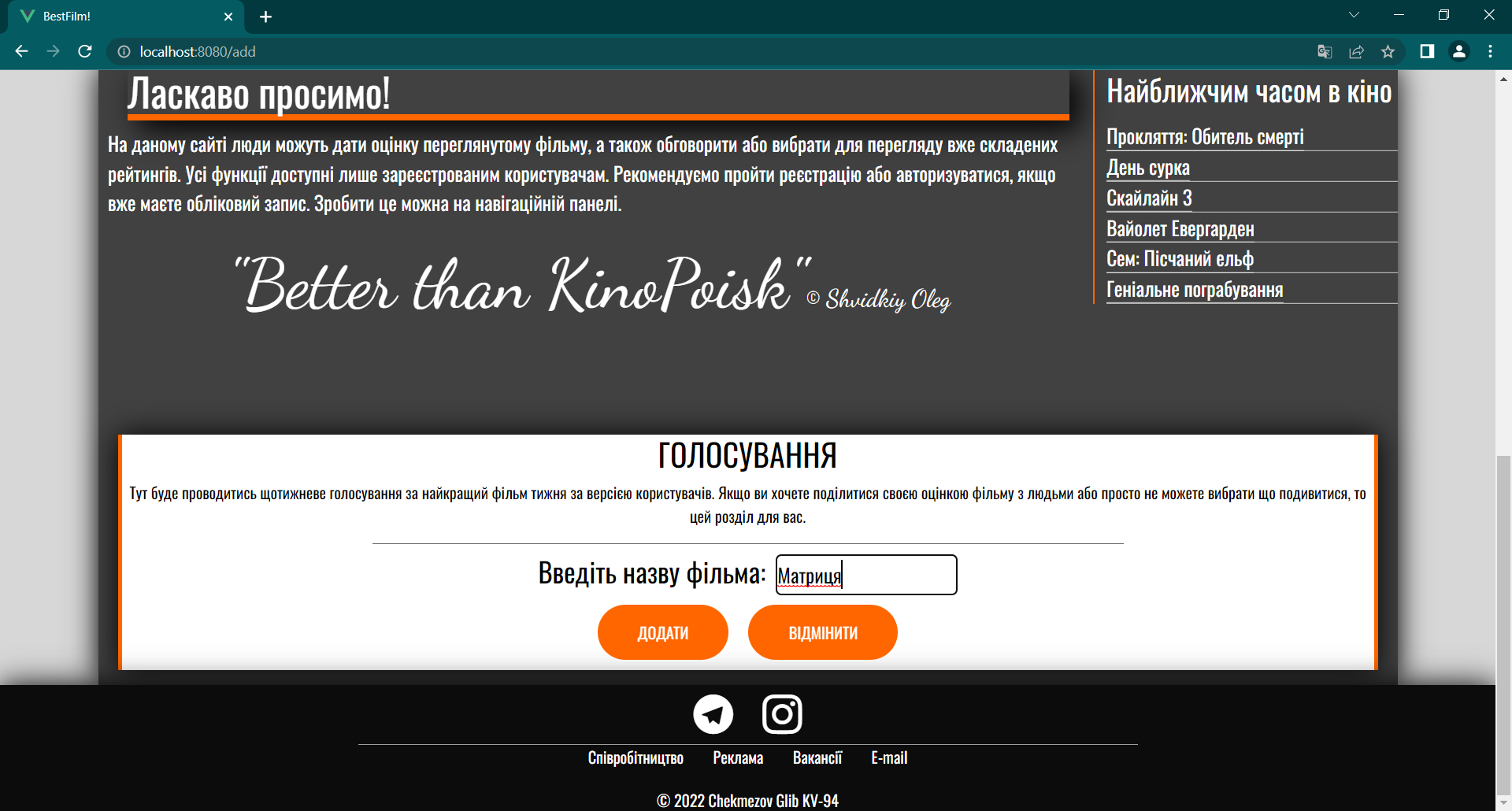


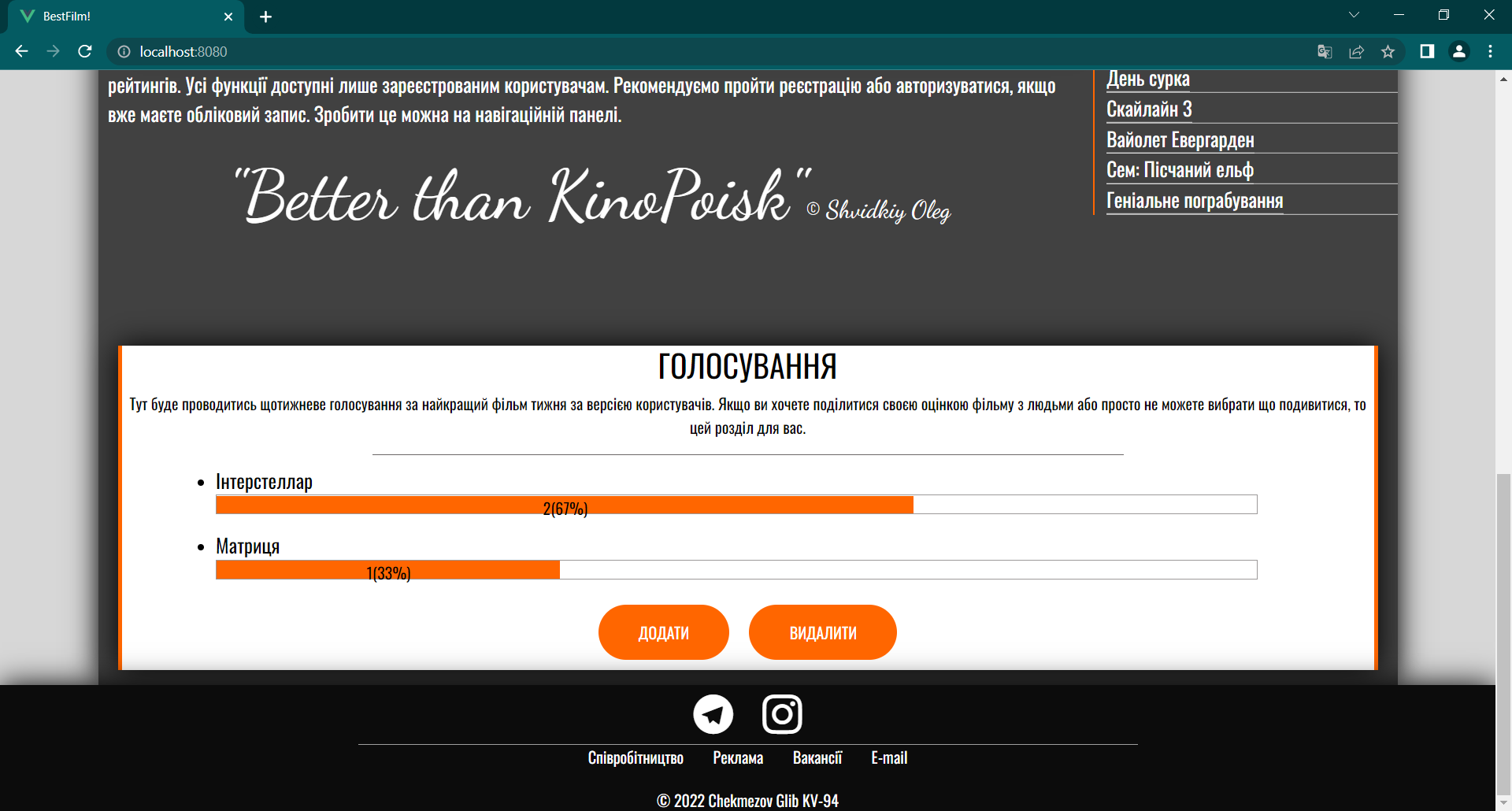












**Код**

**Home.vue**

**<template>**

**<div>**

**<FilmList**

***v-if*="$store.state.ItemModelList.items.length"**

**:itemList="$store.state.ItemModelList.items"**

**/>**

**<h2 *v-else*>Поки що порожньо =(</h2>**

**<div class="poll\_\_buttons-wrapper">**

**<router-link**

**to="/add"**

**class="btn butn-white btn-animated"**

**@click="setPreventDefault($event)"**

**>Додати</router-link**

**>**

**<router-link**

**class="btn butn-white btn-animated"**

**@click="setPreventDefault($event)"**

**to="/del"**

**>Видалити</router-link**

**>**

**</div>**

**</div>**

**</template>**

**<script>**

***import* FilmList *from* "../components/filmList.vue";**

***export* *default* {**

**name: "Home",**

**components: {**

**FilmList,**

**},**

**date() {**

***return* {};**

**},**

**mounted() {},**

**methods: {**

**setPreventDefault(e) {**

**e.preventDefault();**

**},**

**del(e) {**

**e.preventDefault();**

**},**

**},**

**};**

**</script>**

**add.vue**

**<template>**

**<div>**

**<div class="text-box">**

**<h3>Введіть назву фільма:</h3>**

**<input type="text" v-model="title" />**

**</div>**

**<div class="text-box">**

**<router-link to="/" class="btn butn-white btn- animate" @click="add()"**

**>Додати</router-link**

**>**

**<router-link to="/" class="btn butn-white btn- animate"**

**>Відмінити</router-link**

**>**

**</div>**

**</div>**

**</template>**

**<script >**

***export* *default* {**

**data() {**

***return* {**

**title: null,**

**};**

**},**

**methods: {**

**add() {**

***if* (this.title) this.$store.dispatch("ADD\_ITEM", this.title);**

**},**

**},**

**};**

**</script>**

**delete.vue**

**<template>**

**<div>**

**<h3>Список варіантів:</h3>**

**<ul class="text-box" *v-for*="(item, i) in itemList" v-bind:key="item.id">**

**<li**

**v-on:click="item.chosed = !item.chosed"**

**v-bind:class="{ chosed: item.chosed }"**

**>**

**{{ i + 1 }}.{{ item.title }}**

**</li>**

**</ul>**

**<div class="text-box">**

**<router-link to="/" @click="del()" class="btn butn-white btn-animate"**

**>Видалити</router-link**

**>**

**<router-link to="/" class="btn butn-white btn-animate"**

**>Відмінити</router-link**

**>**

**</div>**

**</div>**

**</template>**

**<script >**

***export* *default* {**

**data() {**

***return* {**

**itemList: this.$store.state.ItemModelList.items,**

**};**

**},**

**methods: {**

**del() {**

**this.$store.dispatch("DELETE\_ITEM");**

**},**

**},**

**};**

**</script>**

**main.js**

***import* { createApp } *from* "vue";**

***import* App *from* "./App.vue";**

***import* router *from* "./router";**

***import* store *from* "./store";**

**const app = createApp(App);**

**app.use(router);**

**app.use(store);**

**app.mount("#app");**

**router.js**

***import* { createRouter, createWebHistory } *from* "vue-router";**

***import* home *from* "./views/home";**

***import* add *from* "./views/add";**

***import* del *from* "./views/delete";**

**const routes = [**

**{**

**path: "/",**

**name: "Home",**

**component: home,**

**},**

**{**

**path: "/add",**

**name: "add",**

**component: add,**

**},**

**{**

**path: "/del",**

**name: "del",**

**component: del,**

**props: true,**

**},**

**];**

**const router = createRouter({**

**history: createWebHistory(),**

**routes,**

**});**

***export* *default* router;**

**store.js**

***import* { createStore } *from* "vuex";**

***import* itemModelList *from* "@/model/itemModelList";**

**const store = createStore({**

**state() {**

***return* {**

**ItemModelList: new itemModelList(),**

**};**

**},**

**mutations: {**

**ADD\_ITEM(state, title) {**

**state.ItemModelList.addItem(title);**

**},**

**DELETE\_ITEM(state) {**

**state.ItemModelList.deleteItem();**

**},**

**VOTE(state, id) {**

**state.ItemModelList.vote(id);**

**},**

**DOWNLOAD(state) {**

**state.ItemModelList.downloadItemsFromServer();**

**},**

**},**

**actions: {**

**ADD\_ITEM({ commit }, title) {**

**commit("ADD\_ITEM", title);**

**},**

**DELETE\_ITEM({ commit }) {**

**commit("DELETE\_ITEM");**

**},**

**VOTE({ commit }, id) {**

**commit("VOTE", id);**

**},**

**DOWNLOAD({ commit }) {**

**commit("DOWNLOAD");**

**},**

**},**

**getters: {},**

**});**

***export* *default* store;**

**itemModelList.js**

***import* ItemModel *from* "./itemModel";**

***export* *default* class itemListModel {**

**constructor() {**

**this.items = [];**

**this.totalVotes = 0;**

**this.downloadItemsFromServer();**

**}**

**calculateTotalVotes() {**

**this.totalVotes = 0;**

**this.items.forEach((e) => {**

**this.totalVotes += e.chooseNumberVotes;**

**});**

**this.items.map((item) => {**

**item.calculatePercent(this.totalVotes);**

**});**

**this.sendItemsOnServer();**

**}**

**addItem(title) {**

**const item = new ItemModel(title);**

**this.items.push(item);**

**console.log(this.items);**

**this.sendItemsOnServer();**

**}**

**deleteItem() {**

**let i = 0;**

***while* (i != this.items.length) {**

***if* (this.items[i].chosed) {**

**this.items.splice(i, 1);**

**i--;**

**}**

**i++;**

**}**

**this.calculateTotalVotes();**

**}**

**async downloadItemsFromServer() {**

***await* fetch("http://localhost:3000/")**

**.then((res) => res.json())**

**.then((res) => {**

**res.items.forEach((item) => {**

**const newItem = new ItemModel(**

**item.title,**

**item.chooseNumberVotes,**

**item.percent,**

**item.chosed,**

**item.id**

**);**

**this.items.push(newItem);**

**});**

**});**

**this.calculateTotalVotes();**

**console.log(this.items);**

**}**

**sendItemsOnServer() {**

**fetch("http://localhost:3000/upload", {**

**method: "POST",**

**body: JSON.stringify({ items: this.items }),**

**headers: {**

**"Content-Type": "application/json",**

**},**

**});**

**}**

**vote(id) {**

**const index = this.items.findIndex((item) => item.id === id);**

**this.items[index].vote();**

**this.calculateTotalVotes();**

**}**

**}**

**itemModel.js**

***export* *default* class ItemModel {**

**constructor(**

**title,**

**chooseNumberVotes = 0,**

**percent = 0,**

**chosed = false,**

**id = Math.round(Math.random() \* 100000).toString()**

**) {**

**this.title = title;**

**this.id = id;**

**this.chooseNumberVotes = chooseNumberVotes;**

**this.percent = percent;**

**this.chosed = chosed;**

**}**

**calculatePercent(total) {**

***if* (total != 0)**

**this.percent = Math.round((this.chooseNumberVotes / total) \* 100);**

**console.log(this.percent);**

**}**

**vote() {**

**this.chooseNumberVotes++;**

**}**

**}**

**NodeJS+Express**

**index.js**

**const express = require("express");**

**const bodyParser = require("body-parser");**

**const db = require("./db/db");**

**const item = require("./itemModel.js");**

**const cors = require("cors");**

**const Items = db.Items;**

**const app = express();**

**app.use(bodyParser.json());**

**app.use(cors());**

**const PORT = 3000;**

**app.get("/", (req, res) => {**

**console.log("get");**

**sendTable().then((items) => {**

**res.send(items);**

**});**

**});**

**app.post("/upload", (req, res) => {**

**console.log("post");**

***//   console.log(req.body);***

**writeToDB(req.body);**

**res.sendStatus(201);**

**});**

**app.listen(PORT, () => {**

**console.log(`server has been started on port ${PORT}...`);**

**});**

**async function writeToDB(req) {**

**console.log(req);**

***await* Items.destroy({**

**where: {},**

**truncate: true,**

**});**

**req.items.forEach((item) => {**

**Items.create({**

**id: item["id"],**

**title: item["title"],**

**chooseNumberVotes: item["chooseNumberVotes"],**

**percent: item["percent"],**

**chosed: item["chosed"],**

**});**

**});**

**}**

**function sendTable() {**

***return* Items.findAll()**

**.then((res) => {**

**let responce = { items: [] };**

**res.forEach((item) => {**

**responce.items.push(item.dataValues);**

**});**

***return* responce;**

**})**

**.catch((err) => {**

**console.log(err);**

**});**

**}**

**tableModel.js**

**const Sequelize = require("sequelize");**

**module.exports = function (sequelize) {**

***return* sequelize.define(**

**"Items",**

**{**

**id: {**

**type: Sequelize.INTEGER,**

**primaryKey: true,**

**allowNull: false,**

**},**

**title: {**

**type: Sequelize.TEXT,**

**allowNull: false,**

**},**

**chooseNumberVotes: {**

**type: Sequelize.INTEGER,**

**allowNull: false,**

**},**

**percent: {**

**type: Sequelize.REAL,**

**allowNull: false,**

**},**

**chosed: {**

**type: Sequelize.INTEGER,**

**allowNull: false,**

**},**

**},**

**{**

**timestamps: false,**

**}**

**);**

**};**

**db.js**

**const Sequelize = require("sequelize");**

**const sequelize = new Sequelize({**

**dialect: "sqlite",**

**storage: "./db.sqlite3",**

**});**

**const Items = require("./tableModel")(sequelize);**

**module.exports = {**

**sequelize: sequelize,**

**Items: Items,**

**};**