

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-застосунку для перегляду інформації щодо фільмів
засобами TMDb API, HTML5, Bootstrap та JavaScript ES2018

Виконав: студент IV курсу, групи СН-41
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Базан Т.В.

(прізвище та ініціали)

Керівник

(підпис)

Млинко Б.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Бойко І.В.

(прізвище та ініціали)

Тернопіль
2021

АНОТАЦІЯ

Розробка веб-застосунку для перегляду інформації щодо фільмів засобами TMDb API, HTML5, Bootstrap та JavaScript ES2018 // Кваліфікаційна робота освітнього рівня «Бакалавр» // Базан Тетяна Володимирівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2021 // С. – 47, рис. – 17, табл. – 1, додат. –1, бібліогр. –30 .

Ключові слова: веб-застосунок, SPA, TMDb, API, JavaScript, HTML5, Bootstrap.

В процесі роботи над дипломною роботою було проведено огляд предметної області, описано призначення веб-застосунку: підвищення ступеня поінформованості споживачів щодо популярних фільмів протягом тижня, оглянуто та описано теоретичний матеріал про HTML5, JavaScript ES2018, Bootstrap, TMDb API, зроблено проектування застосунку, розроблено шаблон сторінки, описано розробку веб-застосунку, наведено функції для заповнення контейнера сторінки залежно від вмісту хешу та їх коди, зроблено розгортання сайту на хостингу Heroku, описано проведення валідації та тестування на кросбраузерність і швидкість роботи розробленого застосунку, успішно пройдено валідацію HTML-структуру, виявлено, що застосунок не відображається коректно у старіших версіях браузерів, наведено результат перевірки швидкості роботи застосунку, на мобільних пристроях вона є надто низька, виявлено, що відбувається сильний зсув макету.

ANNOTATION

Web application development for film information browsing using TMDb API, HTML5, Bootstrap and JavaScript ES2018 // Diploma work degree "Bachelor" // Tetyana Bazan // Ternopil Ivan Puluj National Technical University, Department of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2021 // Pages – 47, figures – 17, tables – 1, applications – 1, references – 30.

Tags: web application, SPA, TMDb, API, JavaScript, HTML5, Bootstrap.

In the process of work on the thesis was described the purpose of the web application: raising consumer awareness of popular movies during the week, reviewed and described the theoretical material about HTML5, JavaScript ES2018, Bootstrap, TMDb API, designed the application, developed a page template, described development web-application, the functions for filling the page container depending on the content of the hash and their codes were given, the site was deployed on Heroku hosting, validation and testing for cross-browser and speed of the developed application was described, HTML-structure was successfully validated, the application is not displayed correctly in older versions of browsers, the result of checking the speed of the application is given, on mobile devices it is too low, it is found that there is a strong shift in the layout.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕСМА (англ. European Computer Manufacturer's Association) – Європейська асоціація виробників комп'ютерів.

API (англ. application programming interface) – прикладний програмний інтерфейс.

CSS (англ. cascading style sheets) – каскадні таблиці стилів.

DOM (англ. document object model) – об'єктна модель документа.

HTML (англ. hyper text markup language) – мова розмітки гіпертекстових документів.

SPA (англ. single page application) – застосунок на одній сторінці.

JSON (англ. JavaScript object notation) – запис об'єктів JavaScript.

TMDb (англ. The Movie DataBase) – інтернет база даних.

XML (англ. Extensible Markup Language) – розширювана мова розмітки.

ПК – персональний комп'ютер.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ОДНОСТОРИНКОВОГО ВЕБ-ЗАСТОСУНКУ ДЛЯ ПЕРЕГЛЯДУ ІНФОРМАЦІЇ ПРО ФІЛЬМИ.....	9
1.1 Аналіз відомих рішень.....	9
1.2 Огляд засобів реалізації програмного забезпечення та тестування ..	10
1.2.1 Мова розмітки HTML.....	10
1.2.2 Мова програмування JavaScript.....	12
1.2.3 Стандарт ECMAScript 2018	13
1.2.4 Стилзація за допомогою CSS та Bootstrap	14
1.2.5 The Movie Database API.....	16
1.2.6 Валідація HTML.....	16
1.2.7 Кросбраузерне тестування	17
1.2.8 Швидкість роботи застосунку	18
1.3 Постановка задачі на розробку веб-застосунку для перегляду інформації щодо фільмів	19
РОЗДІЛ 2. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПЕРЕГЛЯДУ ІНФОРМАЦІЇ ПРО ФІЛЬМИ.....	21
2.1 Проектування веб-застосунку.....	21
2.2 Створення шаблону сторінки.....	22
2.3 Популярні фільми	23
2.4 Конкретний фільм	25
2.5 Пошук фільму.....	30
2.6 Додавання контролера для заповнення вмісту контейнера.....	32
2.7 Перенесення застосунку на хостинг та розгортання.....	32
2.8 Валідація та тестування.....	35
2.8.1 Валідація HTML структури	35
2.8.2 Перевірка на кросбраузерність.....	36

2.8.3 Тестування швидкості роботи	36
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	39
3.1 Соціальні та психологічні фактори ризику	39
3.2 Долікарська допомога при ураженні електричним струмом.....	40
3.3 Санітарно-гігієнічні вимоги до умов праці	41
3.3.1 Організація праці на робочому місці	41
3.3.2 Вимоги виробничої санітарії до робочого місця	42
3.3.3 Вибір оптимального режиму роботи і відпочинку.....	42
ВИСНОВКИ.....	44
ПЕРЕЛІК ДЖЕРЕЛ	45
ДОДАТКИ	

ВСТУП

Актуальність дослідження. У наш час спостерігається тенденція щодо розширення ринку надання різноманітних розважальних послуг. До цих послуг, звичайно ж, потрібно віднести й перегляд фільмів. Саме тому в даній роботі було розроблено застосунок для перегляду інформації про фільми.

Мета і задачі дослідження. Підвищення ступеня поінформованості споживачів щодо популярних фільмів протягом тижня.

Для досягнення поставленої мети було сформовано ряд наступних завдань:

- Зробити огляд предметної області.
- Виконати проектування односторінкового веб-застосунку для перегляду інформації про популярні фільми.
- Здійснити розроблення даного веб-застосунку.
- Провести валідацію та тестування даного застосунку.
- Розробити план заходів щодо перенесення розробленого застосунку на хостинг та його розгортання.

РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ОДНОСТОРІНКОВОГО ВЕБ-ЗАСТОСУНКУ ДЛЯ ПЕРЕГЛЯДУ ІНФОРМАЦІЇ ПРО ФІЛЬМИ

1.1 Аналіз відомих рішень

В останні роки зростає популярність фільмів як способу проведення дозвілля. Переглядаючи кінострічки людина з легкістю може поринути в іншу реальність. Улюблені фільми збагачують людське світобачення, змінюють настрій, допомагають навчитися проявляти щирі почуття в реальному житті.

В Україні існує багато сайтів про фільми. Частина з них надає не лише інформацію про фільми, а й можливість їх безкоштовного перегляду онлайн або скачування торентів. Проте такі сайти не є легальними так як порушують авторське право. Відомими прикладами таких сайтів є: moviestape.net, uakino.club, kinoukr.com.

Іншим типом сайтів в обраній предметній області можна виділити медіасервіси, наприклад з найпопулярніших Megogo, Sweet.tv. Ці сайти безкоштовно надають інформацію про фільми та можливість перегляду при передплаті. Також наведені сервіси є провайдерами телебачення. Вони надають багато послуг та можливостей. Це ускладнює їх інтерфейс та функціональність, відсутність сортування за популярністю. На деяких з цих сервісів є спливаючі повідомлення, які пропонують підписку на сервіс. Сторінки таких сайтів мають великий обсяг, оскільки на них є багато медіа даних. При повільному з'єднанні це сповільнює роботу застосунку.

Розглянувши недоліки існуючих рішень було зроблено висновок, що застосунок повинен мати малий обсяг медіа даних, щоб швидкість роботи завжди була високою, простий інтерфейс та відсутність спливаючої реклами.

В таблиці 1.1 наведено порівняльну характеристику додатку, який буде розроблено з відомими рішеннями.

Таблиця 1.1 – Порівняльна таблиця

Характеристика	Відомі рішення	Застосунок, що буде розроблено
1	2	3
Малий розмір сторінки	–	+
Простий інтерфейс	–	+
Реклама	+	–
Велика кількість функцій	+	–

Отже, метою даної кваліфікаційної роботи є розробка односторінкового веб-застосунку, який повинен надавати інформацію про популярні фільми протягом останнього тижня, можливість пошуку по назві та перегляду інформації про конкретний фільм.

1.2 Огляд засобів реалізації програмного забезпечення та тестування

На сьогоднішній день ринок надає широкий вибір технологій реалізації програмного забезпечення. В якості засобів розробки веб-застосунку для перегляду інформації щодо фільмів розглядаються наступні технології: HTML, JavaScript EcmaScript2018, Bootstrap, TMDb API.

1.2.1 Мова розмітки HTML

HTML стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються з її допомогою (або XHTML), саме тому вона буде використаною при розробці застосунку.

HTML-документ оброблюється браузером та відтворюється на екрані у звичному для людини вигляді. Зазвичай веб-сайт включає кілька різних HTML-сторінок. Наприклад: домашня сторінка, звичайна сторінка, сторінка пошуку матимуть окремі HTML-документи.

Кожна HTML-сторінка складається з набору тегів, які можна назвати будівельними блоками веб-сторінок. Вони створюють ієрархію, яка структурує контент за розділами, параграфами, заголовкам і іншим блокам контенту[5].

Більшість елементів HTML мають початковий тег, що містить його ім'я та атрибути та закриваючий тег, що містить косу риску вперед та ім'я.

HTML-теги мають два основних види: блочні та вбудовані.

Елементи рівня блоку займають весь вільний простір і завжди запускають новий рядок в документі, наприклад контейнери, заголовки і параграфи.

Вбудовані елементи займають стільки місця, скільки їм потрібно, і не запускають новий рядок на сторінці. Вони зазвичай служать для форматування внутрішнього вмісту елементів рівня блоку, наприклад посилання та підкреслені рядки.

Як і більшість речей, HTML має як сильні, так і слабкі сторони.

Переваги:

- Широко використовується (велика кількість ресурсів і велике співтовариство).
- Виконується спочатку в кожному веб-браузері.
- У відкритому доступі і абсолютно безкоштовна.
- Чиста і послідовна розмітка.
- Офіційні веб-стандарты підтримуються консорціумом World Wide Web (W3C).
- Легко інтегрується з базовими мовами, такими як PHP і Node.js.

Недоліки:

- В основному використовується для статичних веб-сторінок. Для динамічної функціональності може знадобитися використовувати JavaScript або іншу мову програмування, наприклад PHP.
- Це не дозволяє користувачеві реалізувати логіку. В результаті всі веб-сторінки потрібно створювати окремо, навіть якщо вони використовують одні і ті ж елементи, наприклад заголовки і колонтитули.
- Деякі браузери приймають нові функції повільно.

- Іноді поведінку браузера важко передбачити (наприклад, старі браузери не завжди відтворюють нові теги).

1.2.2 Мова програмування JavaScript

JavaScript – це мова програмування, яка використовується в сторінках HTML для збільшення можливостей. Вона була розроблений фірмою Netscape на базі мови Sun's Java корпорації Sun. JavaScript є надбудовою стандарту HTML і значно розширює можливості html-документа, створеного з використанням цієї технології. JavaScript інтегрується в файл HTML у вигляді декількох рядків коду.

JavaScript в основному використовується для веб-програм та веб-браузерів. Але JavaScript також використовується поза веб-мережею в програмному забезпеченні, на серверах та вбудованих апаратних елементах управління. Ось кілька основних речей, для яких використовується JavaScript[8]:

А) Додавання інтерактивної поведінки до веб-сторінок

JavaScript дозволяє користувачам взаємодіяти з веб-сторінками, наприклад дає змогу:

- показувати або приховувати додаткову інформацію при натисненні кнопки;
- змінювати колір кнопки, коли курсор наводиться на неї;
- переглядати зображення за допомогою каруселі (стрічки зображень) на домашній сторінці;
- збільшувати або зменшувати зображення;
- відображати таймера або зворотній відлік на веб-сайті;
- та інше.

Б) Створення веб- і мобільних додатків

Розробники можуть використовувати різні фреймворки JavaScript для розробки та створення веб- і мобільних додатків. Фреймворки JavaScript - це

колекції бібліотек коду JavaScript, які надають розробникам заздалегідь написаний код, який використовується для рутинних функцій програмування та завдань.

Популярні інтерфейсні фреймворки JavaScript включають React, React Native, Angular та Vue. Багато компаній використовують Node.js, середовище виконання JavaScript, побудоване на механізмі JavaScript V8 від Google Chrome. Кілька відомих прикладів включають PayPal, LinkedIn, Netflix та Uber!

В) Створення веб-серверів та розробка серверних додатків

Крім веб-сайтів та додатків, розробники також можуть використовувати JavaScript для створення простих веб-серверів та розвитку внутрішньої інфраструктури за допомогою Node.js.

Г) Розробка ігор

Також можливо використовувати JavaScript для створення браузерних ігор. Це чудовий спосіб для початківців розробників відпрацювати свої навички JavaScript.

1.2.3 Стандарт ECMAScript 2018

ECMAScript є стандартом для мов сценаріїв, таких як JavaScript, JScript тощо. JavaScript - це мова, заснована на ECMAScript і вважається однією з найпопулярніших реалізацій ECMAScript [7].

Дев'яте видання стандарту ECMAScript, офіційно відоме як ECMAScript 2018 (або скорочено ES2018), вийшло в червні 2018 року.

Нові можливості ES2018:

- Властивості Rest/Spread.
- Асинхронна ітерація.
- Promise.prototype.finally.
- Нові функції RegExp.

1.2.4 Стилзація за допомогою CSS та Bootstrap

CSS - це мова для опису презентації веб-сторінок, включаючи кольори, макет та шрифти [18]. Це дозволяє адаптувати презентацію до різних типів пристроїв, таких як великі екрани, маленькі екрани або принтери. CSS не залежить від HTML і може використовуватися з будь-якою мовою розмітки на основі XML. Відокремлення HTML від CSS полегшує ведення веб-сайтів, обмін таблицями стилів між сторінками та адаптацію сторінок до різних середовищ. Це називається відокремленням структури (або змісту) від презентації [6].

Є ряд переваг CSS, серед яких:

- Швидша швидкість сторінки. CSS дозволяє використовувати одне правило CSS і застосовувати його до всіх випадків певного тегу в документі HTML, що зменшує кількість коду і пришвидшує роботу сторінки.
- Покращений досвід користування. CSS не тільки полегшує огляд веб-сторінок, але і дозволяє зручне форматування. Коли кнопки та текст знаходяться в логічних місцях і добре організовані, покращується взаємодія з користувачами.
- Швидший час розробки. За допомогою CSS можливо застосовувати певні правила форматування та стилі до кількох сторінок з одним рядком коду. Одну каскадну таблицю стилів можна відтворити на декількох сторінках веб-сайту.
- Простота форматування змін. Якщо потрібно змінити формат певного набору сторінок, то не потрібно виправляти кожну окрему сторінку, а можна просто відредагувати відповідну таблицю стилів CSS, і зміни будуть застосовані до всіх сторінок, які використовують цю таблицю стилів.

Bootstrap – це повний frontend каркас, який допомагає розробляти веб-додатки без необхідності хвилюватися про їх зовнішній інтерфейс. Фреймворк дозволяє розробникам вибирати з діапазону популярних повторно використовуваних компонентів HTML, які заздалегідь розроблені та стилізовані Bootstrap [10].

Переваги Bootstrap:

- Адаптивна сітка. Бібліотека постачається із власною попередньо визначеною системою сіток.
- Чутливі зображення. Bootstrap має свій код для автоматичної зміни розміру зображень на основі поточного розміру екрана.
- Компоненти. В бібліотеці наявний цілий набір компонентів, які можна легко прикріпити до веб-сторінки, наприклад: навігаційні панелі, випадаючі меню, індикатори прогресу.
- JavaScript. Bootstrap також дозволяє розробникам скористатися перевагами понад десятка користувацьких плагінів JQuery. Ця бібліотека надає вам ще більше місця для гри з інтерактивністю, пропонуючи прості рішення для модальних спливаючих вікон, переходів, каруселей зображень.
- Документація. Кожен фрагмент коду детально описується та пояснюється на офіційному веб-сайті. Пояснення також включають зразки коду для базової реалізації, що спрощує процес.
- Зовнішні шаблони. По мірі зростання популярності люди почали створювати шаблони на основі Bootstrap, щоб ще більше пришвидшити процес веб-розробки. Існує багато веб-сайтів, присвячених спільному користуванню та купівлі власних шаблонів на основі Bootstrap.

Основними недоліками використання бібліотеки є:

- Заплутаний синтаксис. При першому ознайомленні Bootstrap, деякі частини його синтаксису можуть заплутати. Наприклад, використовуючи систему сіток, щоб створити стовпець, який займає третину екрану, до нього потрібно додати клас `.col-md-4`. (Bootstrap використовує 12-стовпчасту систему, а 4 - це третина з 12).
- Надто великі файли. Файли Bootstrap можуть бути великими, щоб врахувати чисту функціональність, запропоновану його фреймворком. Це може призвести до збільшення часу завантаження веб-сайтів, особливо в повільних мережах.

1.2.5 The Movie Database API

TMDb - це онлайн-база даних, що стосується фільмів, телевізійних програм в Інтернеті - включаючи акторський склад, виробничу групу та особисті біографії, опис сюжетів, рейтинги. Кожна частина даних була додана співтовариством починаючи з 2008 року[21].

Переваги TMDb:

- Щороку з 2008 року кількість внесків до бази даних зростала. TMDb має понад 400 000 розробників та компаній, що використовують платформу.
- Поряд із великими метаданими про фільми, телешоу та людей, також можна знайти одну з найкращих добірок плакатів із високою роздільною здатністю та фанартів. В середньому щодня додається понад 1000 зображень.
- Інтернаціональність. Проект офіційно підтримує 39 мов, та щодня використовується у понад 180 країнах.

TMDb-API – це веб-сервіс для отримання фільмів, серійної та акторської інформації. Результати API є JSON і включають такі елементи, як специфікації фільмів, зображення, плакати, рейтинги тощо.

1.2.6 Валідація HTML

Досить часто при написанні коду допускаються помилки. Неправильний або нестандартний код може спричинити несподівані результати у тому, як відображається або функціонує сторінка у браузері. Щоб запобігти цьому буде перевірено HTML-код на відповідність офіційним вимогам та стандартам, встановленим Консорціумом World Wide Web для веб-сторінок HTML / XHTML[11].

Консорціум World Wide Web надає простий онлайн-інструмент (<https://validator.w3.org/>), який автоматично перевіряє HTML-код та вказує на

будь-які проблеми/помилки, які може мати код, наприклад, відсутність закриваючих тегів або відсутність лапок навколо атрибутів .

Валідація веб-сторінки - це процес забезпечення її відповідності нормам або стандартам, визначеним Консорціумом всесвітньої паву, який є повноваженнями підтримувати стандарти HTML.

Існує кілька конкретних причин перевірки веб-сторінки, деякі з них:

- Це допомагає створювати веб-сторінки, сумісні між браузерами та платформами. Він також може бути сумісним із майбутньою версією веб-браузерів та веб-стандартами.
- Веб-сторінки, що відповідають стандартам, покращують видимість сторінки для пошукової машини, в результаті чого вона з більшою ймовірністю відобразатиметься в результатах пошуку.
- Це зменшить несподівані помилки та зробить веб-сторінки більш доступними для відвідувачів.

1.2.7 Кросбраузерне тестування

Кросбраузерність – це умова, яка вимагає щоб сайт однаково і коректно відображався у всіх сучасних браузерах і їх попередніх версіях[12].

Кросбраузерне тестування - це тип нефункціонального тестування, що дозволяє перевірити, чи працює веб-сайт належним чином при доступі через:

- Різні комбінації браузер-ОС, тобто у таких популярних браузерах, як Firefox, Chrome, Edge, Safari - у будь-якій з популярних операційних систем, таких як Windows, macOS, iOS та Android.
- Різні пристрої, тобто користувачі можуть переглядати та взаємодіяти з вашим веб-сайтом на популярних пристроях - смартфонах, планшетах, настільних ПК та ноутбуках тощо.
- Допоміжні інструменти, тобто веб-сайт сумісний з допоміжними технологіями, такими як зчитувачі екрану, для осіб, які мають різні можливості.

1.2.8 Швидкість роботи застосунку

Коли люди використовують веб-програми, вони хочуть, щоб вони забезпечували швидку та ефективну роботу. Як результат, швидкість стала однією з найкращих метрик, яку люди використовують для оцінки якості дизайну веб-додатків. Тому, після розробки веб-застосунку буде проведено тестування швидкості-роботи на мобільних пристроях та на ПК.

Дослідження споживачів показує, що реакція на стрес на затримку мобільної швидкості подібна до реакції на перегляд фільму жахів або розв'язання математичної задачі та більша, ніж очікування на касі в роздрібному магазині.

Коли сайт починає завантажуватися, користувачі чекають появи вмісту через певний проміжок часу. Поки цього не відбувається, про користувацький досвід не може бути й мови. Ця відсутність досвіду швидкоплинна на швидких сполученнях. Однак при повільному з'єднанні користувачі змушені чекати. Користувачі можуть відчувати більше проблем, оскільки ресурси сторінки повільно стікають.

Сайти часто перевантажені тим, що ми можемо вважати невеликою кількістю неоптимізованого коду. Це створює низьку ефективність, що призводить до невідповідності. Знаючи, що ми знаємо про поведінку людей, користувачі будуть терпіти лише низькопродуктивні програми обмежений час, перш ніж відмовитися від них.

Web Vitals - це ініціатива Google щодо надання уніфікованих вказівок щодо якісних сигналів, необхідних для забезпечення чудового досвіду роботи в Інтернеті[15].

За ці роки Google запропонував ряд інструментів для вимірювання та звітування про ефективність.

Core Web Vitals – це підмножина важливих веб-даних, які застосовуються до всіх веб-сторінок, вони повинні вимірюватися всіма власниками сайтів і відображатимуться в усіх інструментах Google.

Показники, що складають Core Web Vitals, з часом змінюються. На 2020 рік вони складаються з таких трьох аспектів взаємодії з користувачем:

Largest Contentful Paint [29]. Найбільша візуалізація контенту показує, коли з'являється найбільший текст чи зображення. Щоб забезпечити хороший досвід роботи, LCP повинен відбуватися протягом 2,5 секунд після першого початку завантаження сторінки.

First Contentful Paint [27]. Перша візуалізація вмісту показує, коли з'являється текст чи зображення.

Cumulative Layout Shift [24]. Сукупне зміщення макета вимірює рух видимих елементів у межах області перегляду. Вимірює візуальну стабільність. Щоб забезпечити хороший досвід роботи, сторінки потрібно підтримувати цей показник менше 0,1.

Також в даній дипломній роботі були досліджені наступні показники для розробленого застосунку:

- Time to Interactive [26]. Час до повного завантаження – це період часу, через який сторінка стане повністю інтерактивною.
- Speed Index [28]. Індекс швидкості показує, через скільки часу відображається вміст сторінки.
- Total Blocking Time [25]. Загальна тривалість усіх періодів часу в мілісекундах між першою візуалізацією вмісту та часом до повного завантаження, коли час виконання завдання перевищує 50 мс.

1.3 Постановка задачі на розробку веб-застосунку для перегляду інформації щодо фільмів

Метою даної кваліфікаційної роботи є розробка односторінкового веб-застосунку, який повинен надавати інформацію про популярні фільми протягом останнього тижня, можливість пошуку по назві та перегляду інформації про конкретний фільм.

Застосунок повинен бути динамічним, а саме вміст сторінки повинен змінюватись залежно від виконуваного завдання, наповнення сторінки має завантажуватись при потребі з віддаленої бази даних через API. Для забезпечення цього буде використано мову програмування JavaScript.

Для розробки буде використано стандарт ES2018 завдяки його новій можливості `Promise.prototype.finally`, яка буде використана при написанні коду.

Для стилізації буде використовуватися бібліотека Bootstrap, оскільки вона є простою у використанні, має необхідні компоненти та зрозумілу документацію.

Як джерело даних буде використано The Movie Database API, оскільки TMDb містить актуальну інформацію про всі нові і популярні фільми.

РОЗДІЛ 2. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПЕРЕГЛЯДУ ІНФОРМАЦІЇ ПРО ФІЛЬМИ

2.1 Проектування веб-застосунку

Завданням даної дипломної роботи є розробка односторінкового застосунку (SPA) для перегляду інформації про фільми.

SPA - це веб-застосунок на одній сторінці, який завантажує всі файли JavaScript (модулі, віджети, елементи управління тощо) та файли CSS для забезпечення роботи та завантаження самої сторінки.

Переваги такого підходу:

- SPA-програми працюють бездоганно як на стаціонарних, так і на мобільних;
- широкий користувальницький інтерфейс, так званий досвід користувача;
- SPA значно зменшує так зване ходіння по колу, тобто завантаження одного і того ж вмісту знову і знову.

Застосунок має 3 основні функції:

- виведення фільмів, що були популярні протягом останнього тижня;
- виведення інформації про конкретний фільм;
- пошук фільму за назвою.

Стартовою сторінкою є сторінка на якій виведено 20 найпопулярніших фільмів протягом останнього тижня. В шапці сторінки можна здійснити пошук фільму за назвою та перехід на головну сторінку. Для перегляду інформації про конкретний фільм потрібно натиснути на його назву. Оскільки застосунок складається з однієї зміни вмісту контейнера здійснюється за допомогою відслідковування зміни хешу.

2.2 Створення шаблону сторінки

Сторінка складається з 3 частин: навігаційної панелі, контейнера та футера. На рисунку 2.1 показаний шаблон сторінки.



Рисунок 2.1 – Шаблон сторінки

В шапці сторінки розміщено посилання на стартову сторінку та форму для пошуку конкретного фільму. В лістингу 2.1 наведено відповідний код.

У контейнері знаходиться основний вміст сторінки, html-код можна переглянути у лістингу 2.2. Він заповнюється залежно за допомогою контролера, це описано у розділі 2.6 даної пояснювальної записки.

Лістинг 2.1 – Навігаційна панель

```
<nav class="navbar navbar-dark bg-primary">
  <a class="navbar-brand" href="#start" style="margin: px;">Популярні за
тиждень</a>
  <form class="form-inline my-2 my-lg-0">
    <input class="form-control mr-sm-2" type="search" id ="search"
placeholder="Введіть назву фільму" >
    <button class="btn btn-primary" type="submit"onclick="ButtonClick()">
Шукати</button>
  </form>
</nav>
```

Лістинг 2.2 – Контейнер

```
<div class="container" style="background-color: rgba(176, 199, 221,
0.788);"> <div id = 'result' class="row"></div></div>
```

У останній частині сторінки наведено інформацію про розробника.
Відповідний код у лістингу 2.3.

Лістинг 2.3 – Футер сторінки

```
<footer class="bg-light text-center text-lg-start">
  <hr style="margin:0px">
  <div class="text-center p-3" style="background-color: rgba(57,
109, 158, 0.521);">
    © 2020 Bazan Tetyana, SN-41, TNTU
  </div>
</footer>
```

2.3 Популярні фільми

На головній сторінці розробленого застосунку виводиться інформація про 20 найпопулярніших фільмів за останній тиждень. Для її отримання здійснюється запит за допомогою функції `fetch` за посиланням:

`https://api.themoviedb.org/3/trending/movie/week?api_key={key}&language=uk&include_image_language=uk,null`, де `{key}` – це ключ отриманий при реєстрації на сайті The Movie Database. Отримана відповідь декодується в формат JSON. За допомогою документації TMDb API можна дізнатись склад даної відповіді. На рисунку 2.2 наведено схему відповіді API на даний запит.

Field	Type	Optional
object		
page	integer	optional
object (Movie List Result Object)		optional
poster_path	string or null	optional
adult	boolean	optional
overview	string	optional
release_date	string	optional
genre_ids	array[integer]	optional
id	integer	optional
original_title	string	optional
original_language	string	optional
title	string	optional
backdrop_path	string or null	optional
popularity	number	optional
vote_count	integer	optional
video	boolean	optional
vote_average	number	optional
total_pages	integer	optional
total_results	integer	optional

Рисунок 2.2 – Схема відповіді TMDb API на запит про популярні фільми

Для виведення на головні сторінці інформації про найпопулярніші фільми необхідні назва, зображення та ідентифікатор фільму. Ця інформація зберігається у масиві `results`. За допомогою метод `map()` отримуються `title`, `id` кожного елемент масиву `results`, вони зберігаються як текст `title`, елементу `<a>`, та як посилання на сторінку відповідно. Також `poster_path` зберігається як посилання на зображення `img`. Для кожного елементу створюється картка `<div>` з класом `card`. Після цього до карток додаються нащадки `img` та `title`. Відповідний код наведено в лістингу 2.4.

Лістинг 2.4 – Обробка інформації про найпопулярніші фільми

```

root.className = "row row-cols-md-5";
root.innerHTML = "";
res.results.map(el => {
  const card = document.createElement('div');
  card.className = "card border border-light shadow-0 mb-5";

  const title = document.createElement('a');
  title.href = "#" + el.id;
  title.innerText = el.title;
  title.className = "btn btn-light card-body";
  const br = document.createElement('br');

  const img = document.createElement('img');
  img.src = 'http://image.tmbd.org/t/p/w200' + el.poster_path;
  img.className = "card-img-top";

  card.appendChild(img);
  card.appendChild(br);
  card.appendChild(title);
  root.appendChild(card);
})

```

Кожна картка додається як нащадок `root`, елемент `<div>` з `id = "result"`.

Розроблену сторінку можна переглянути на рисунку 2.3.

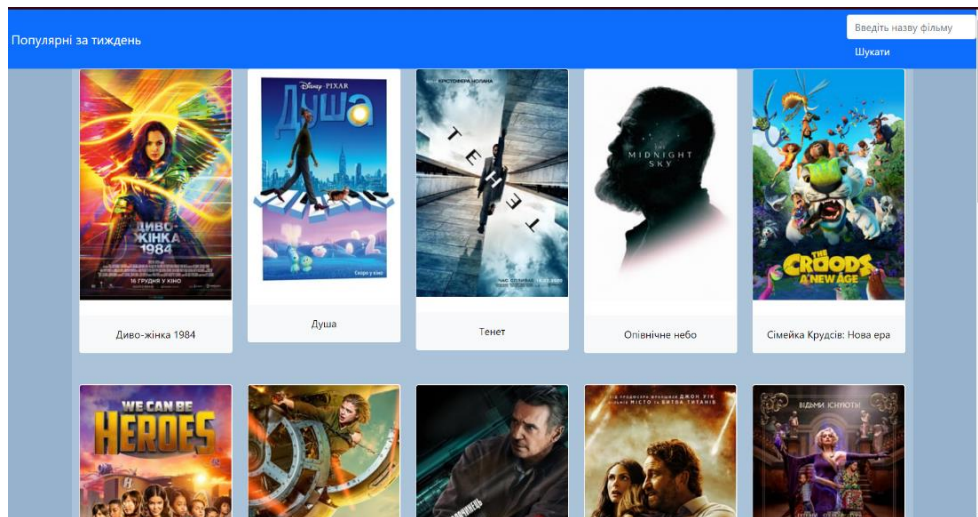


Рисунок 2.3 – Сторінка з популярними фільмами

Як видно з рисунку інформація про кожен фільм виводиться в окремих картках. Назва фільму є посиланням на інформацію про нього.

2.4 Конкретний фільм

Однією з функцій застосунку є вивід інформації про конкретний фільм. Для цього при натисненні на назву фільму відбувається перехід на сторінку з конкретним фільмом, вона має вигляд як на рисунку 2.4.

Контейнер розділено на 2 колони: у першій розміщено назву фільму та його афішу, а в другій його опис та посилання на подібні фільми.

Для отримання інформації про конкретний фільм використовується таке посилання: https://api.themoviedb.org/3/movie/'+id+'?api_key={key}&language=uk&include_image_language=uk,null. Ідентифікатор фільму передається при виклику функції (див. розділ 2.6).

Після отримання відповіді та її декодування у формат JSON потрібно отримати дані з масиву. Для того, щоб знати, які дані де зберігаються в отриманому масиві було використано схему на рисунку 2.5.

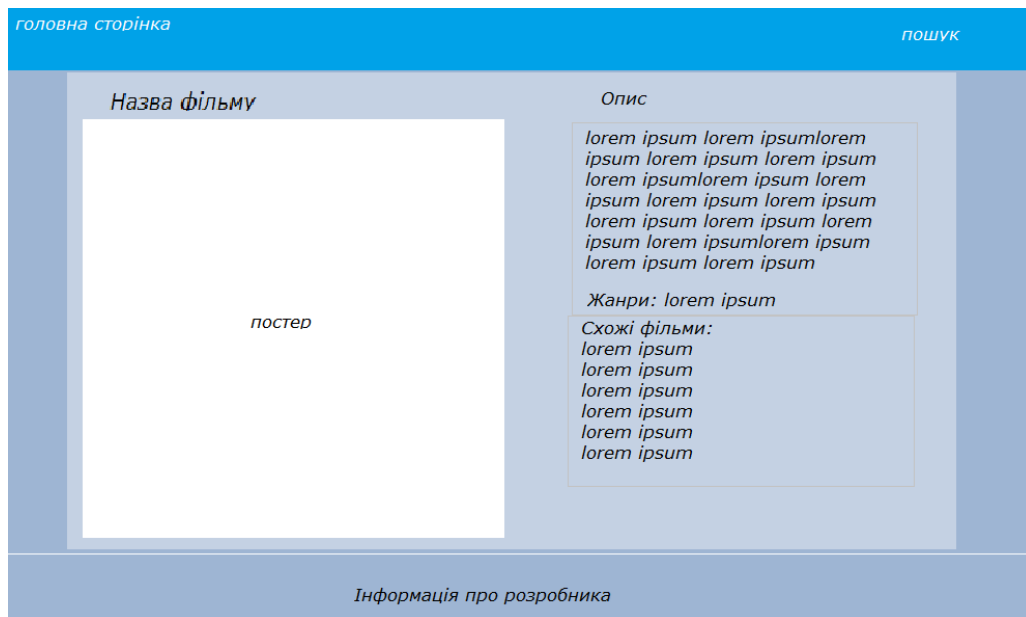


Рисунок 2.4 – Макет сторінки з інформацією про фільм

Schema	Example		expand all
object			
adult	boolean	optional	
backdrop_path	string or null	optional	
belongs_to_collection	null or object	optional	
budget	integer	optional	
▶ genres	array[object]	optional	
homepage	string or null	optional	
id	integer	optional	
imdb_id	string or null 3 validations	optional	
original_language	string	optional	
original_title	string	optional	
overview	string or null	optional	
popularity	number	optional	
poster_path	string or null	optional	
▶ production_companies	array[object]	optional	
▶ production_countries	array[object]	optional	
release_date	string 1 validations	optional	
revenue	integer	optional	
runtime	integer or null	optional	
▶ spoken_languages	array[object]	optional	
status	string 1 validations	optional	
tagline	string or null	optional	
title	string	optional	
video	boolean	optional	
vote_average	number	optional	
vote_count	integer	optional	

Рисунок 2.5 – Схема відповіді на запит про конкретний фільм

З цих дані необхідні назва, шлях до афіші, жанри та опис фільму. У лістингу 2.5 наведено код, який опрацьовує отримані дані та поміщує їх у відповідні елементи DOM-дерева. Спочатку відбувається створення двох

колонок `imgTitle` та `opys`. Після цього створюються елементи `title`, `img`, `dt`, `about`, `dt2`, `genres`. Їм присвоюються необхідні класи, далі відбувається заповнення їх інформацією з отриманого масиву.

Лістинг 2.5 – Обробка інформації про фільм

```
const imgTitle = document.createElement('div')
imgTitle.className = "col card"
const opys = document.createElement('dl')
opys.className = "col"

const title = document.createElement('h1')
title.innerText = res.title
const img = document.createElement('img')
img.src = 'http://image.tmbd.org/t/p/w500' + res.poster_path
img.class = "card-img-top"
imgTitle.appendChild(title)
imgTitle.appendChild(img)

const dt = document.createElement('dt')
dt.innerText = "Опис:  "

const about = document.createElement('dd')
about.innerText = res.overview
about.className = "text-muted"

const dt2 = document.createElement('dt')
dt2.innerText = "Жанри:  "
let a = ""
const genres = document.createElement('dd')
res.genres.map (g =>
{
  a+=g.name + "  "
})
genres.innerText = a;
genres.className = "text-muted"

let card = document.createElement('dl')
GetSimilary(card, id)

root.appendChild(imgTitle)
root.appendChild(opys)
opys.appendChild(dt)
opys.appendChild(about)
opys.appendChild(dt2)
opys.appendChild(genres)
opys.appendChild(card)
```

Для отримання даних про схожі фільми використовується функція `GetSimilary`. Її код можна переглянути в лістингу 2.6.

Лістинг 2.6 – Отримання інформації про схожі фільми

```
function GetSimilary(card, id)
{
  const dt = document.createElement('dt')
  dt.innerText = "Схожі фільми:"
  card.appendChild(dt)

  const dd = document.createElement('dd')
  dd.className = "text-muted"

  url = " https://api.themoviedb.org/3/movie/"+id+"/similar?api
_key=3a66dca4ab982982a2e4d01db94410b4&language=uk&page=1"
  fetch(url)
  .then(response=>response.json())
  .then(res => {
    if(res.total_results === 0)
    {
      dd.innerText = "Схожих фільмів не знайдено(("
    }
    else
    {
      res.results.map(r=>{
        const title = document.createElement('a')
        const br = document.createElement('br')
        title.href = "#"+r.id;
        title.innerText = r.title
        dd.appendChild(title)
        dd.appendChild(br)
      })
    }
  })
  card.appendChild(dd)
}
```

Аналогічно як і з попередніми функціями для того щоб знати вміст відповіді використовується схема відповіді, що на рисунку 2.6.

page	integer	optional
▼ results	array(object) (Movie List Result Object)	optional
poster_path	string or null	optional
adult	boolean	optional
overview	string	optional
release_date	string	optional
genre_ids	array(integer)	optional
id	integer	optional
original_title	string	optional
original_language	string	optional
title	string	optional
backdrop_path	string or null	optional
popularity	number	optional
vote_count	integer	optional
video	boolean	optional
vote_average	number	optional
total_pages	integer	optional
total_results	integer	optional

Рисунок 2.6 – Схема відповіді на запит про подібні фільми

На рисунку 2.7 можна переглянути сторінку з інформацію про фільм «Відьми».

Популярні за тиждень

Введіть назву фільму

Шукати

Відьми

ВІДЬМИ ІСНУЮТЬ!

ЕННІ ГЕТЕВЕЙ ОКТАВІЯ СПЕНСЕР ГІБЕРТІ ГІЛЧІ

ВІДЬМИ

Опис:
Фільм розповідає загадкову гумористичну та зворушливу казку про юного сироту. Наприкінці 1967 року він їде жити до своєї люблячої бабусі у містечко Демополіс, що у штаті Алабама. Коли хлопчик та бабуся зустрічають кількох гламурних, але насправді злих відьом, вона мудро вирішує забрати юного героя на дорогий курорт біля моря. На жаль, вони прибувають туди десь тоді само, коли Верховна Відьма збирає під прикриттям подруг з усього світу, щоб втілити у життя свої підступні плани.

Жанри:
Фентезі Сімейний Пригоди Комедія Жахи

Схожі фільми:
[Відьми](#)
[Хроніки Нарнії: Лев, чаклунка та шафа](#)
[Мері і відьомська квітка](#)
[Сьомий син](#)
[Хроніки Нарнії: Поїми Каспіан](#)
[Нарівник країни Оз](#)
[Зоряний Пил](#)
[Гаррі Поттер і філософський камінь](#)
[The Muppets' Wizard of Oz](#)
[Кожен Аєць](#)
[Доктор Дулітл](#)
[Хранитель часу](#)
[Віллі Вонка та шоколадна фабрика](#)
[Громобій](#)
[Джеймс і персик-гігант](#)
[Іствікські відьми](#)
[Академія вампірів](#)

Рисунок 2.7 – Інформація про фільм «Відьми»

З рисунку видно що вивід інформації відбувається в двох колонках як і було зображено на макеті (рис.2.4).

2.5 Пошук фільму

Для пошуку необхідного фільму використовується форма з лістингу 2.1 та функція `SearchFilm()`, її код можна переглянути в лістингу 2.7.

Лістинг 2.7 – Пошук

```

root.innerHTML=""

const box = document.getElementById("search")
const movie = box.value;
const card = document.createElement('div')
if (movie !== "")
{
    url = "https://api.themoviedb.org/3/search/movie?api_key=3a6
6dca4ab982982a2e4d01db94410b4&language=uk&query="+movie+"&page=
1&include_adult=true"
    fetch(url)
    .then(response => response.json())
    .then(res => {
        if(res.total_results == 0)
        {
            card.innerHTML = "Нічого не знайдено(( Спробуйте вве
сти назву фільму англійською"
        }
        else
        {
            card.innerHTML = "Результати:"
            res.results.map(el => {
                const title = document.createElement('a')
                const br = document.createElement('br')
                title.href = "#"+el.id;
                title.innerHTML = el.title
                card.appendChild(br)
                card.appendChild(title)
                console.log(el.title)
            })
        }
        root.appendChild(card)
    })
}

```

Спочатку отримуються фільм введений користувачем. Якщо стрічка пошуку не є пустою, то за допомогою функції `fetch()` отримується вміст за посиланням: `https://api.themoviedb.org/3/search/movie?api_key={key}&`

language=uk&query="+movie +"&page=1&include_adult=true, де movie шуканий фільм. Схему відповіді можна переглянути на рисунку 2.8.

object		
page	integer	optional
▼ results	array[object] {Movie List Result Object}	optional
poster_path	string or null	optional
adult	boolean	optional
overview	string	optional
release_date	string	optional
genre_ids	array[integer]	optional
id	integer	optional
original_title	string	optional
original_language	string	optional
title	string	optional
backdrop_path	string or null	optional
popularity	number	optional
vote_count	integer	optional
video	boolean	optional
vote_average	number	optional
total_results	integer	optional
total_pages	integer	optional

Рисунок 2.8 – Схема відповіді на пошуковий запит

Якщо кількість результатів дорівнює 0, то виводиться текст про відсутність результатів. В іншому випадку виводяться посилання на знайдені фільми. На рисунку 2.9 можна переглянути результати пошуку фільму під назвою «Один дома».

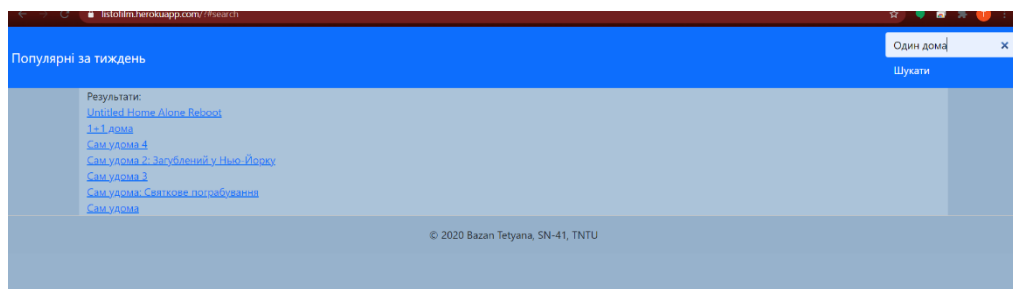


Рисунок 2.9 – Результати пошуку фільму «Один дома»

З рисунку видно, що було знайдено 9 фільмів і пошук працює правильно.

2.6 Додавання контролера для заповнення вмісту контейнера

Для того, щоб застосунок мав всього одну сторінку, а вміст контейнера заповнювався відносно хешу використовується функція `Control()`. За допомогою функції `getRouteInfo()` вона отримує вміст кешу і залежно від нього викликає необхідну функцію:

- `GetTrendFilms()` для виведення популярних фільмів за тиждень;
- `SearchFilm()` для пошуку фільму;
- `GetMovie()` для виведення даних про конкретний фільм.

Для кожної з функцій в якості параметру передається елемент `<div>` ідентифікатор якого `result`. Для пошуку конкретного фільму передається вміст хешу, який в даному випадку є ідентифікатором. Код функцій наведено в лістингу 2.8.

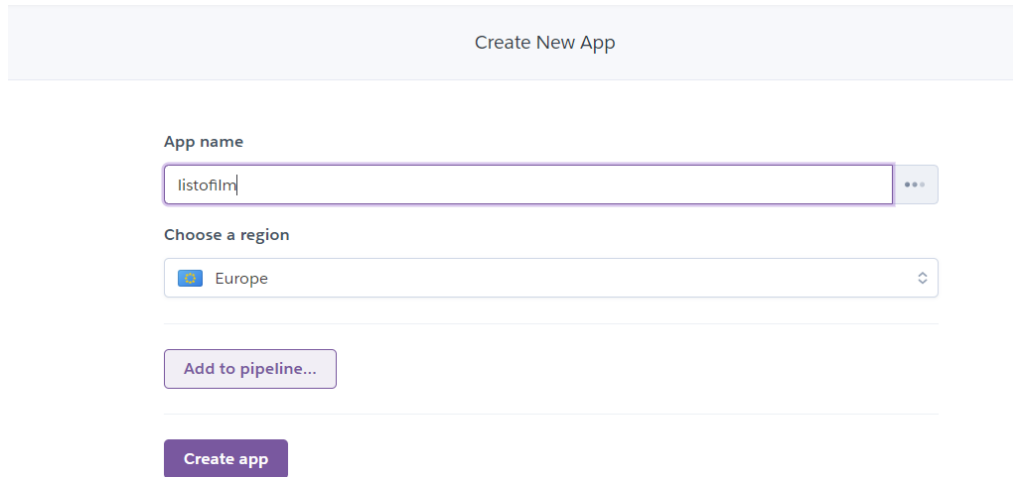
Лістинг 2.8 – Функція `Control()` та `getRouteInfo()`

```
function Control()
{
  const root = document.getElementById('result')
  const id = getRouteInfo()
  if (id=="start") GetTrendFilms(root)
  else if (id=="search") SearchFilm(root)
  else GetMovie(root, id)
}

function getRouteInfo()
{
  const hash = location.hash.slice(1);
  return hash
}
```

2.7 Перенесення застосунку на хостинг та розгортання

Для того щоб зробити застосунок для перегляду інформації про фільми доступним його було розміщено на хостингу Heroku. Для цього було створено новий застосунок `listofilm`, рисунок 2.10.



Create New App

App name
listofilm

Choose a region
Europe

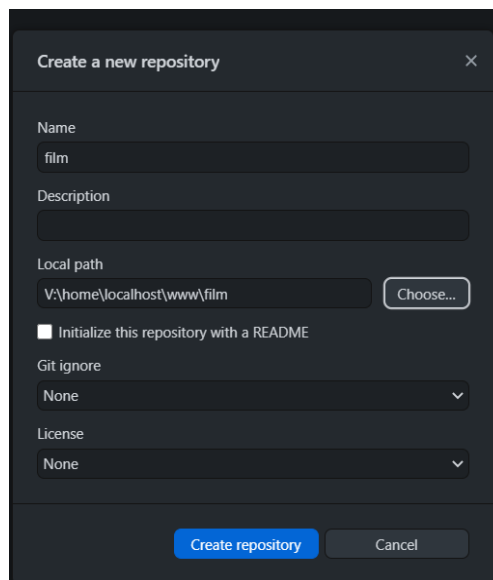
Add to pipeline...

Create app

Рисунок 2.10 – Створення нового застосунку на Heroku

Heroku надає 3 можливих методи розгортання: Heroku Git, GitHub та Container Registry. Для розгортання даного застосунку було обрано другий метод, використання GitHub репозиторію.

Для цього спочатку в папці, де зберігається код застосунку, було створено новий git-репозитарій, рисунок 2.11. Для цього було використано програму GitHub Desktop.



Create a new repository

Name
film

Description

Local path
V:\home\localhost\www\film Choose...

Initialize this repository with a README

Git ignore
None

License
None

Create repository Cancel

Рисунок 2.11 – Створення нового репозиторію

Після створення репозитарій публікується на GitHub, рисунок 2.12.

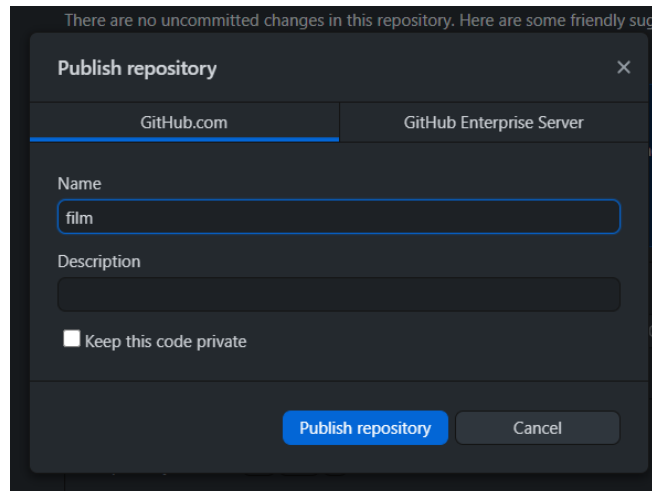


Рисунок 2.12 – Публікування репозиторію

Новий репозитарій розміщено за посиланням <https://github.com/Me1Z1n/film>.

Далі відбувається підключення репозиторію до створеного застосунку, рисунок 2.13.

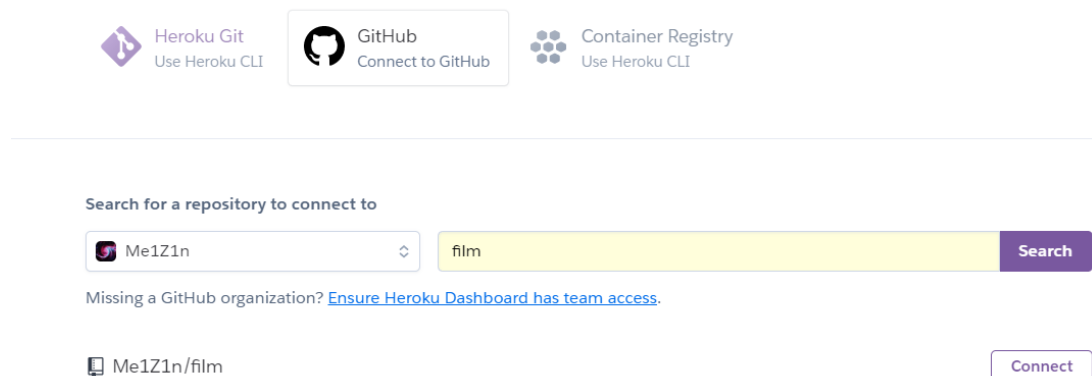


Рисунок 2.13 – Підключення до репозиторію

Після підключення відбувається увімкнення автоматичного розгортання вітки master, рисунок 2.14.

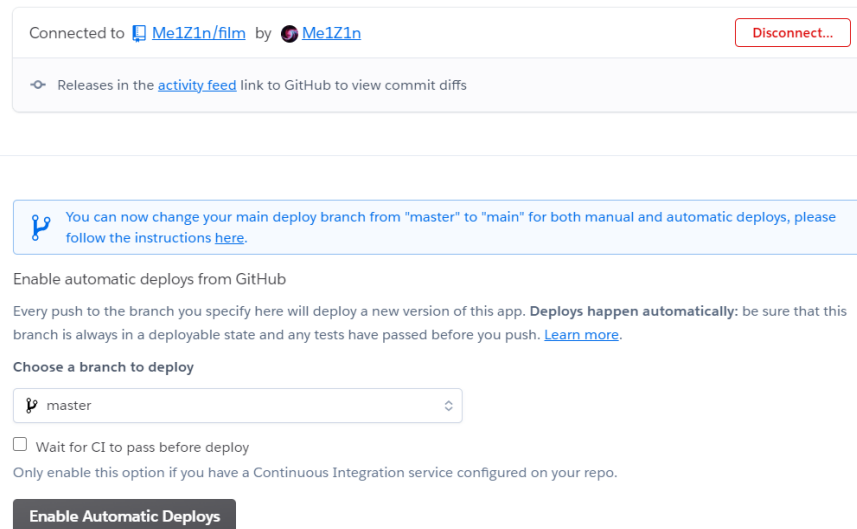


Рисунок 2.14 – Увімкнення автоматичного розгортання

Створений застосунок розміщено за посиланням <https://listofilm.herokuapp.com>.

2.8 Валідація та тестування

2.8.1 Валідація HTML структури

Валідацію HTML було здійснено використовуючи Nu HTML Checker, який розміщено за посиланням <https://validator.w3.org/>. Цей валідатор перевіряє відповідність стандартам розмітки веб-документів HTML, XHTML, SMIL, MathML тощо.

На рисунку 2.15 зображено результат валідації створеного сайту.

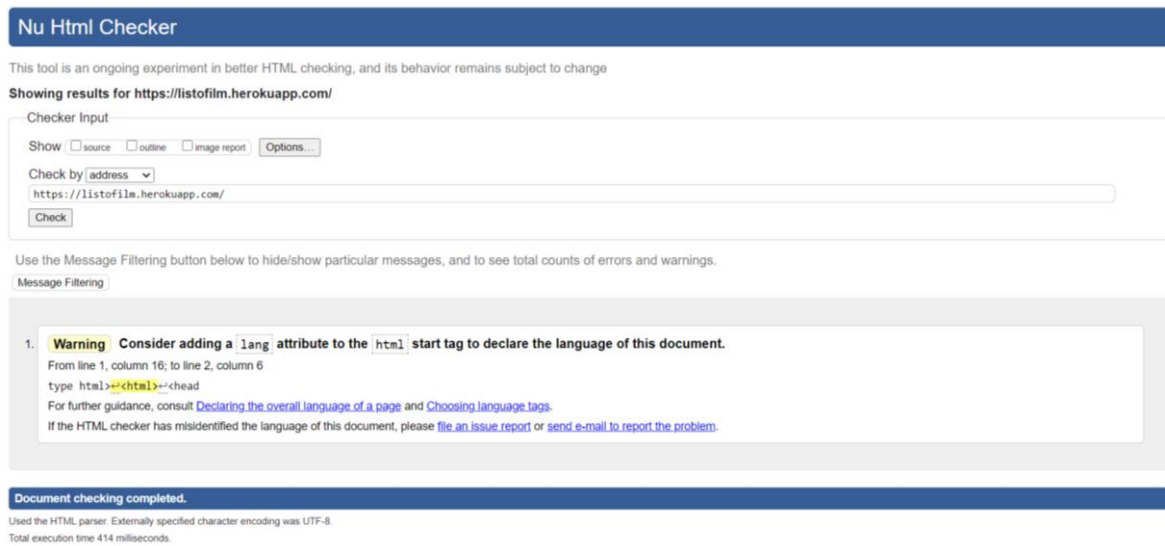


Рисунок 2.15 – Валідація HTML структури

В результаті було виявлено лише одне попередження про додавання атрибуту lang до тегу <html>.

2.8.2 Перевірка на кросбраузерність

Для перевірки використовувався застосунок BrowserStack, який розміщено за посиланням <https://www.browserstack.com/screenshots>.

Результати перевірки наведено у додатку А. Розроблений застосунок для перегляду інформації про фільми відображається коректно лише в останніх версіях браузерів. Перевірка не пройдена.

2.8.3 Тестування швидкості роботи

Для тестування швидкості роботи застосунку для перегляду інформації про фільми використовувався Google PageSpeed Insights, він розташований за посиланням <https://developers.google.com/speed/pagespeed/insights/>. На рисунку 2.16 показані результати для мобільних пристроїв, а на рисунку 2.17 – для комп'ютерів.

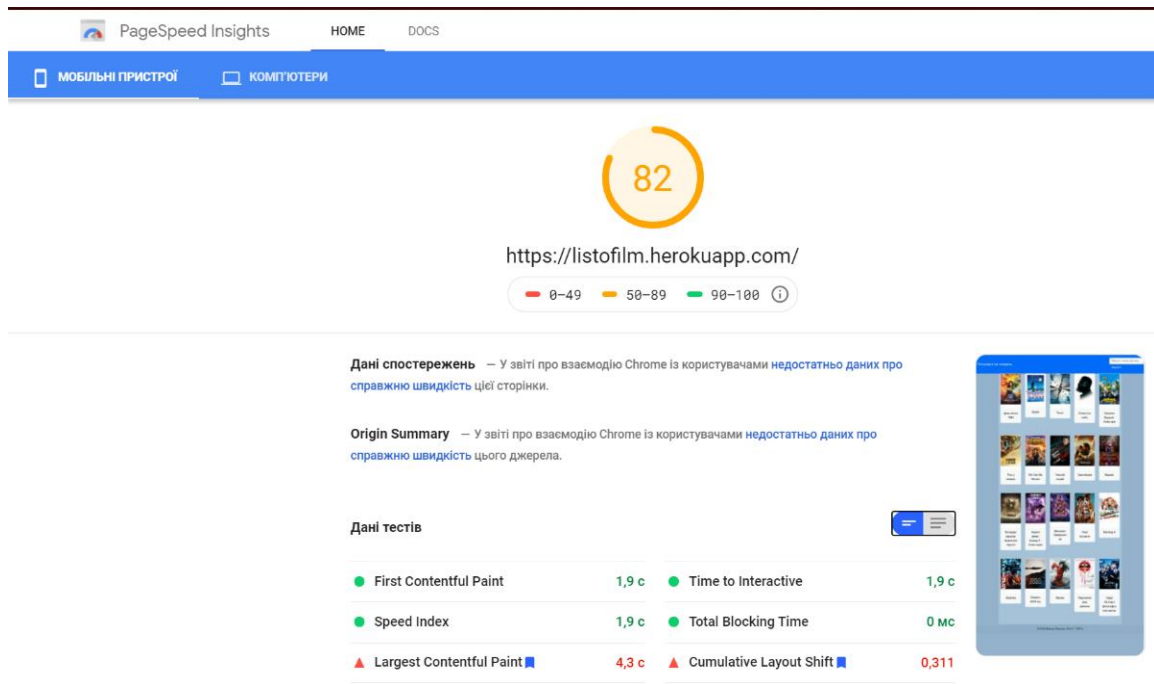


Рисунок 2.16 – Результати тесту для мобільних пристроїв

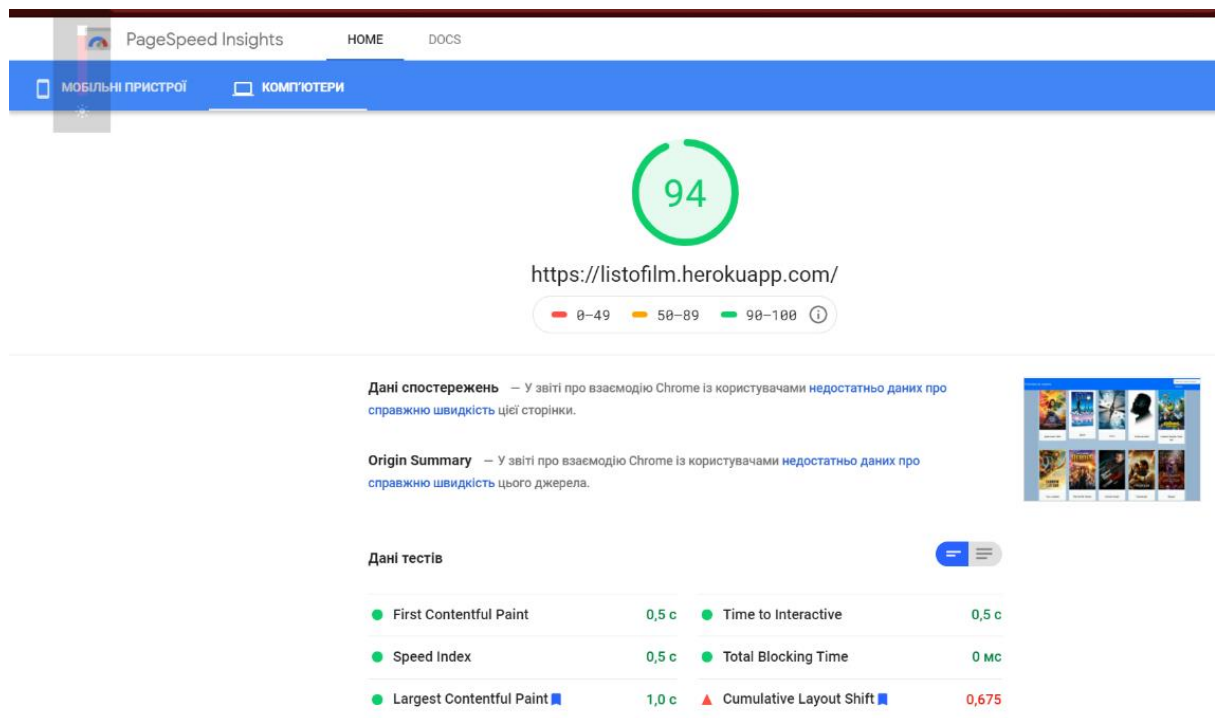


Рисунок 2.17 – Результати тесту для комп'ютерів

Аналізуючи результати тесту для мобільних пристроїв видно, що загальний результат 82 бали. Це означає, що застосунок потребує вдосконалення. Сукупне зміщення макету становить 0,3111, що є низьким

показником. Також час завантаження найбільшого елемента теж є надто довгим. Для усунення цих проблем потрібно додати перевірку розміру екрану, щоб не завантажувати великі зображення без потреби.

Результати тесту для комп'ютерів є добрими. Проте показник сукупного зміщення макету є надто низьким.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Соціальні та психологічні фактори ризику

При розробці веб-застосунків щодо перегляду інформації про фільми соціальні та психологічні фактори ризиків впливають на безпеку життєдіяльності людини, в даному випадку розробника.

Груповий або соціальний ризик являє собою залежність між частотою подій (аварій, катастроф, стихійних лих) та кількістю постраждалих в них людей, характеризує масштаби і тяжкість негативних наслідків надзвичайних ситуацій, а також різного роду явищ і перетворень, що знижують якість життя людей.

Джерела соціального ризику:

- урбанізація екологічно нестійких територій (поселення людей в зонах можливого затоплення, виникнення зсувів, селів, ландшафтних пожеж, виверження вулканів, підвищена сейсмічність регіону);
- промислові технології та об'єкти підвищеної небезпеки (аварії на АЕС, ТЕС, хімічних комбінатах; транспортні катастрофи; техногенне забруднення навколишнього середовища);
- соціальні та військові конфлікти (бойові дії; застосування зброї масового знищення);
- епідемії (поширення інфекцій);
- зниження якості життя (голод, убогість; погіршення медичного обслуговування; низька якість продуктів харчування; незадовільні житлово-побутові умови).

Соціальні та психологічні фактори ризику:

- соціальні фактори, що впливають на життя та здоров'я людини, корупція і криміналізація суспільства, маніпуляція свідомістю;

- розрив у рівні забезпечення життя між різними прошарками населення, шкідливі звички, соціальні хвороби та їхня профілактика, алкоголізм та наркоманія;
- психологічна надійність людини та її роль у забезпеченні безпеки, захисні властивості людського організму;
- види поведінки людини та її психічна діяльність: психічні процеси, стани, властивості, поняття про психоемоційні напруження (стрес);
- види напруження, психотипи за реакцією людей на небезпеку, частота змін стресових станів у людей, що знаходяться в районі небезпечної ситуації;
- основи підвищення психофізіологічної стійкості кадрів до професійних небезпек;
- зростання злочинності як фактор небезпеки, види злочинних посягань на людину.

3.2 Долікарська допомога при ураженні електричним струмом

При розробці веб-застосунку постійно відбувається взаємодія з електричними приладами. Таким чином актуальним питанням при виконанні цього завдання є допомога при ураженні електричним струмом.

Дотик до струмопровідних частин (мережі під напругою) у більшості випадків зумовлює судомі м'язів - тобто людина не може самостійно відірватися від провідника. Необхідно швидко відключити ту частину електрообладнання, до якої доторкнулася людина.

Звільнення потерпілих в електроустановках напругою до 1000 В.

Для звільнення потерпілого використовують сухий одяг, палицю, дошку, шапку, сухі рукавиці, рукав одягу, діелектричні рукавиці.

Можна для звільнення потерпілого від джерела струму використати підстилку, потрібно діяти однією рукою, не дотикаючись до металевих частин другою.

Можна відкинути оголений провід сухою палицею від потерпілого, перерізати або перерубати його сокирою, яка знаходиться на пожежному щитку.

Внаслідок дії електроструму розрізняють три стани людського організму:

- I стан - потерпілий при свідомості: забезпечити повний спокій, проводити 2-3 годинне спостереження, викликати лікаря.
- II стан - потерпілий непритомний, але дихає, людину покласти горизонтально, розстебнути комір і пасок, дати нюхати нашатирний спирт чи оббризкати лице водою, викликати лікаря;
- III стан - потерпілий не дихає або дихає з перервами, уривчасто, як вмираючий. Робити штучне дихання і непрямий масаж серця.

3.3 Санітарно-гігієнічні вимоги до умов праці

При виконанні завдання розробки веб-застосунку щодо перегляду інформації про фільми умови праці відповідали санітарно-гігієнічним нормам[1][4].

3.3.1 Організація праці на робочому місці

Організація праці на робочому місці — це комплекс заходів, що забезпечують трудовий процес та ефективне використання знарядь виробництва і предметів праці.

Робоче місце — це зона, яка оснащена технічними засобами і в якій відбувається трудова діяльність працівника чи групи працівників.

Організація праці на робочому місці полягає у виборі робочої пози та системи робочих рухів, визначення розмірів робочої зони та розміщення у ній органів керування, інструментів, заготовок, матеріалів, пристроїв, а також у виборі оптимального режиму праці та відпочинку.

Розробка застосунку відбувається в робочій позі сидячи. Вона характеризується невеликою статичною стомлюваністю, більш спокійним положенням рук, можливістю виконання точної роботи, що найкраще підходить для роботи за ПК. Робоче місце для відповідає вимогам[2].

3.3.2 Вимоги виробничої санітарії до робочого місця

Площа робочого місця має бути такою, щоб, по-перше, дозволяла зручно, з найменшими витратами енергії, безпечно та продуктивно проводити трудовий процес, тобто щоб відповідала нормам технологічного проектування, і, по-друге, складала не менше 4,5 м² на одного працівника при площі приміщення не менше 15 м² [3].

Кожне робоче місце:

- обладнується необхідними засобами колективного захисту;
- укомплектовується необхідними засобами індивідуального захисту;
- має достатнє натуральне та штучне освітлення;
- має параметри мікроклімату відповідно до санітарних норм;
- має вентиляцію.

3.3.3 Вибір оптимального режиму роботи і відпочинку

Під оптимальним режимом праці і відпочинку розуміється таке чергування періодів праці і відпочинку, при якому досягається найбільша ефективність трудової діяльності людини і хороший стан її здоров'я.

Оптимальний режим праці і відпочинку досягається: паузами та перервами в роботі, зміною умов довкілля, відпочинком у спеціальних кабінетах-кімнатах, використанням психічного впливу музики.

Чергування праці і відпочинку встановлюють відповідно до ДСанПіН 3.3.2-007-98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

На початку, 10:00, завжди має місце стадія “впрацьовування” або наростаючої працездатності, коли відбувається відновлення робочих навиків. Тривалість цього періоду становить 0,5 год.

Швидкість і точність дій у цей період невеликі. Потім настає стадія високої стійкої працездатності тривалістю 3 год. Після цього настає стадія зменшення працездатності або стадія розвитку втоми, рухи уповільнюються і увага розсіюється, сприйняття притупляється. В цей час, 13:30, роблять обідню перерву.

Після обідньої перерви, 14:30, впрацьовування настає швидко — за 10-15 хв, бо робочі навики не втрачені. Працездатність у другій половині дня дещо нижча, ніж до обіду, і становить 80...90% дообіднього рівня. Через 3 год після обідньої перерви працездатність зменшується і в кінці робочого дня, 18:00, приблизно сягає рівня, який був на початку робочого дня.

Для зменшення стомлення встановлюють регламентовані перерви в роботі у періоди, що передують зменшенню працездатності. При розумовій праці ефективні довгі перерви на відпочинок і переключення на фізичну роботу. Загальна тривалість відпочинку встановлюється у відсотках до тривалості робочої заміни: при фізичній роботі вона має становити 4...20%, при роботі із нервовою напругою – 14...25%, а при розумовій праці – до 10...12%.

ВИСНОВКИ

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Зроблено огляд предметної області.
- Описано призначення веб-застосунку: підвищення ступеня поінформованості споживачів щодо популярних фільмів протягом тижня.
- Здійснено вибір засобів розробки: HTML5, JavaScript ES2018, Bootstrap, TMDb API.

В другому розділі роботи:

- Зроблено проектування застосунку.
- Розроблено шаблон сторінки.
- Описано розробку веб-застосунку.
- Наведено функції для заповнення контейнера сторінки залежно від вмісту хешу та їх коди.
- Зроблено розгортання застосунку на хостингу Heroku.
- Описано валідацію та тестування на кросбраузерність та швидкість роботи розробленого застосунку.
- Успішно пройдено валідацію HTML-структуру.
- Виявлено, що застосунок не відображається коректно у старіших версіях браузерів.
- Наведено результат перевірки швидкості роботи застосунку. На мобільних пристроях вона є кою.
- Виявлено, що відбувається сильний зсув макету.

У розділі «Безпека життєдіяльності, основи хорони праці» розглянуто соціальні та психологічні фактори ризику, порядок надання долікарської допомоги при ураженні електричним струмом. Висвітлено санітарно-гігієнічні вимоги до умов праці.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.
- 2 ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги.
- 3 ДСП 173-96 Державні санітарні правила планування і забудови населених пунктів.
- 4 ДСанПіН 3.3.2-007-98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
- 5 Anna. Что такое HTML? Основы языка разметки гипертекста [Електронний ресурс] / Anna // Hostinger – 2020. – Режим доступу до ресурсу: <https://www.hostinger.ru/rukovodstva/shto-takoje-html/>.
- 6 HTML & CSS [Електронний ресурс] // W3C. – 2016. – Режим доступу до ресурсу: <https://www.w3.org/standards/webdesign/htmlcss.html>.
- 7 Sreemaha. What is the difference between JavaScript and ECMAScript? [Електронний ресурс] / Sreemaha // Tutorials Point. – 2018. – Режим доступу до ресурсу: <https://www.tutorialspoint.com/What-is-the-difference-between-JavaScript-and-ECMAScript>.
- 8 What is JavaScript Used For? [Електронний ресурс] // Hack Reactor. – 2018. – Режим доступу до ресурсу: <https://www.hackreactor.com/blog/what-is-javascript-used-for>.
- 9 Kelhini F. New ES2018 Features Every JavaScript Developer Should Know [Електронний ресурс] / Faraz Kelhini // CSS-Tricks. – 2019. – Режим доступу до ресурсу: <https://css-tricks.com/new-es2018-features-every-javascript-developer-should-know/>.

10 Segal N. 10 Common Uses of Bootstrap [Електронний ресурс] / Nathan Segal // HTMLGOODIES. – 2016. – Режим доступу до ресурсу: <https://www.htmlgoodies.com/html5/10-common-uses-of-bootstrap/>.

11 How to Validate HTML Pages [Електронний ресурс] // Tutorial Republic. – 2017. – Режим доступу до ресурсу: <https://www.tutorialrepublic.com/html-tutorial/html-validation.php>.

12 Cross Browser Testing [Електронний ресурс] // BrowserStack. – 2019. – Режим доступу до ресурсу: <https://www.browserstack.com/cross-browser-testing>.

13 Walton P. Web Vitals [Електронний ресурс] / Philip Walton // web.dev. – 2020. – Режим доступу до ресурсу: <https://web.dev/vitals/>.

14 Why does speed matter? [Електронний ресурс] // web.dev. – 2019. – Режим доступу до ресурсу: <https://web.dev/why-speed-matters/>.

15 Web Application Speed: Why Performance Matters [Електронний ресурс] // Achievion. – 2017. – Режим доступу до ресурсу: <https://achievion.com/blog/web-application-speed-why-performance-matters.html>.

16 Мороз В. М. Технології створення та супроводження веб-сайтів / В. М. Мороз // Збірник тез доповідей VI Міжнародної науково-технічної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 16-17 листопада 2017 року. — Т. : ТНТУ, 2017. — Том 2. — С. 123–124.

17 Зеленко О.В., Валеева Л.Р., Климанов С.Г. Обзор современных. Web - технологий // Вестник Казанского технологического университета. 2015. т.18, в.2. С. 354-356.

18 MDN. Основи CSS – Вчимо веб-розробку [Електронний ресурс] / MDN – Режим доступу до ресурсу: https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/CSS_basics.

19 Шкабара, В. С. Розробка адаптивного сайту з використанням Bootstrap // Актуальні питання сучасної інформатики: збірник наукових праць студентів, магістрантів та викладачів / за ред. Т. А. Вакалюк. – Житомир: Вид-во ЖДУ ім. І. Франка, 2016. – Вип. 2. – 146 с. – С. 122-128.

20 Драбинко, В.П. Методика розробки односторінкових web-додатків // Наукове забезпечення технологічного прогресу ХХІ сторіччя: матеріали міжнародної наукової конференції (Т.2), 1 травня, 2020 рік. Чернівці, Україна: МЦНД. С.72-75.

21 The Movie Database API [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.themoviedb.org/3/trending/get-trending>.

22 Валідація веб-сторінок [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/da5206site/validation>.

23 Завгородній, В. В., Завгородня, Г. А. Сучасні технології розробки веб-сайтів. / В. В. Завгородній, Г. А. Завгородня. Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та праві міжнародна науково-практична конференція 19-20 квітня 2018 року, матеріали конференції. С. 34-36.

24 Walton P. Cumulative Layout Shift (CLS) [Електронний ресурс] / P. Walton, M. Mihajlija. – 2019. – Режим доступу до ресурсу: https://web.dev/cls/?utm_source=lighthouse&utm_medium=unknown.

25 Total Blocking Time [Електронний ресурс] – Режим доступу до ресурсу: https://web.dev/lighthouse-total-blocking-time/?utm_source=lighthouse&utm_medium=unknown.

26 Time to Interactive [Електронний ресурс] – Режим доступу до ресурсу: https://web.dev/interactive/?utm_source=lighthouse&utm_medium=unknown.

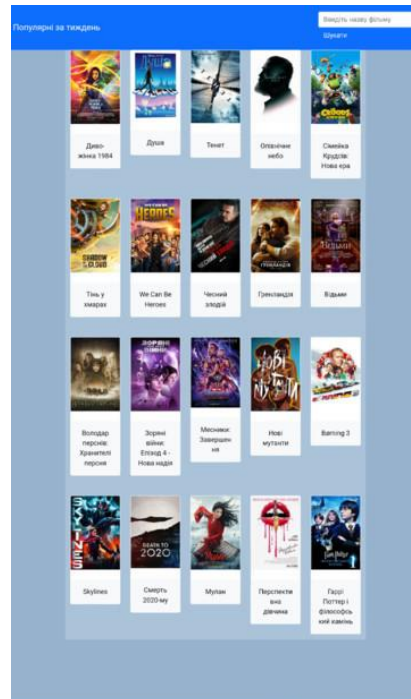
27 First Contentful Paint [Електронний ресурс] – Режим доступу до ресурсу: https://web.dev/first-contentful-paint/?utm_source=lighthouse&utm_medium=unknown.

28 Speed Index [Электронный ресурс] – Режим доступа до ресурсу: https://web.dev/speed-index/?utm_source=lighthouse&utm_medium=unknown.

29 Largest Contentful Paint [Электронный ресурс] – Режим доступа до ресурсу: https://web.dev/lighthouse-largest-contentful-paint/?utm_source=lighthouse&utm_medium=unknown.

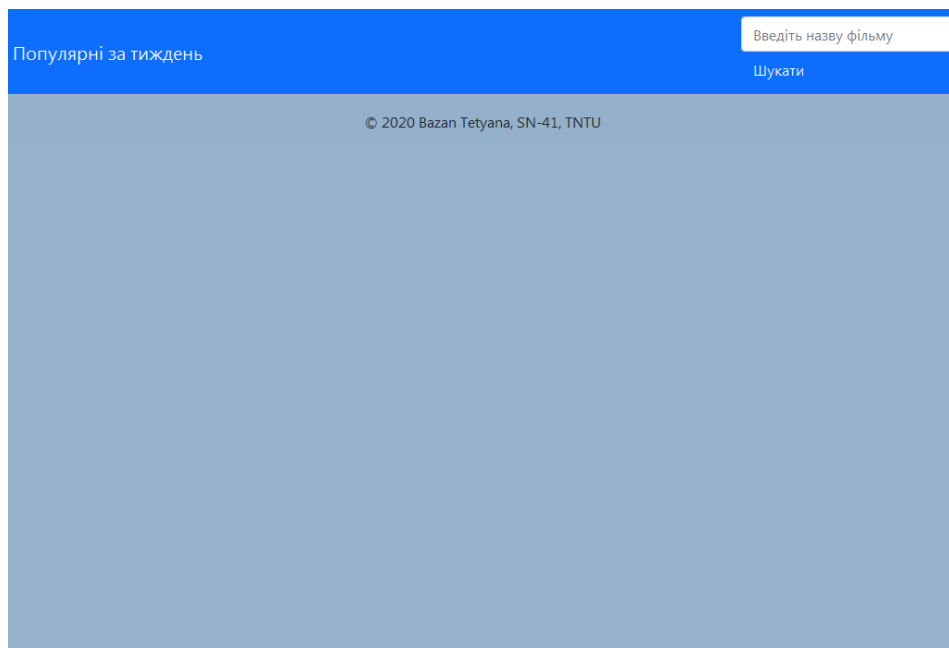
30 Lighthouse performance scoring [Электронный ресурс] – Режим доступа до ресурсу: <https://web.dev/performance-scoring/>.

ДОДАТКИ



BrowserStack

Samsung Galaxy S7



BrowserStack

Windows 7 Firefox v35

Популярні за тиждень

Введіть назву фільму

Шукати

Диво-жінка 1984	Душа	Тенет	Опівнічне небо	Смейка Крудсів: Нова ера
Тінь у хмарах	We Can Be Heroes	Чесний злодій	Гренландія	Відьми
Володар персня: Хранителі перся	Зоряні війни: Епізод 4 - Нова надія	Месники: Завершення	Нові мутанти	Varning 3
Skylines	Смерть 2020-му	Мулан	Перспективна дівчина	Гаррі Поттер і філософський камінь

© 2020 Bazan Tetyana, SN-41, TNTU

BrowserStack

Windows 8 Firefox v62

Введіть назву фільму

Шукати

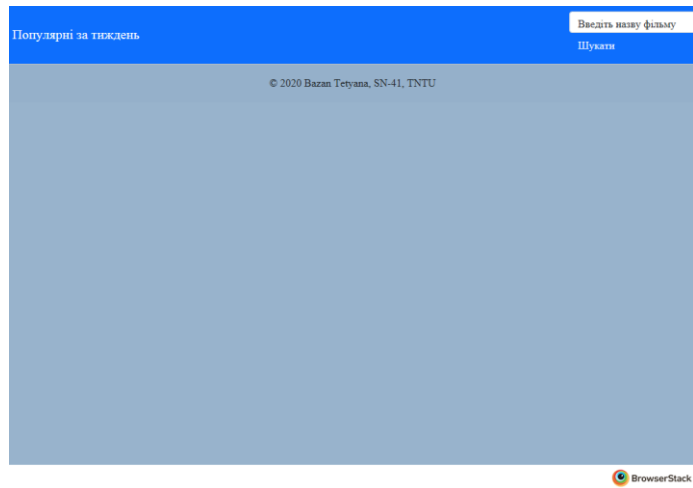
Популярні за тиждень

Диво-жінка 1984	Душа	Тенет	Олівічне небо	Сімейка Крудсів: Нова ера
Тінь у хмарах	We Can Be Heroes	Чесний злодій	Гренландія	Відьми
Володар пернів: Хранителі перся	Зоряні війни: Епізод 4 - Нова надія	Месники: Завершення	Нові мутанти	Ворнінг 3
Skylines	Смерть 2020-му	Мулан	Перспективна дівчина	Гаррі Поттер і філософський камінь

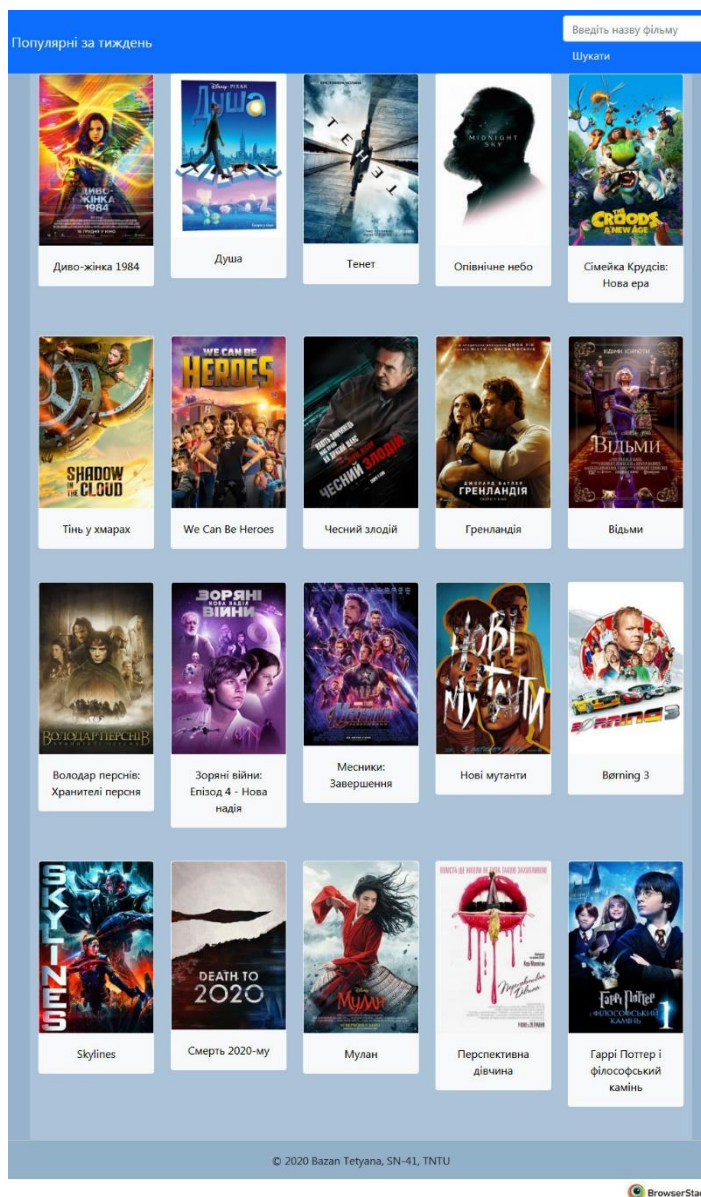
© 2020 Bazan Tetyana, SN-41, TNTU

BrowserStack

Windows 10 Chrome v66



Windows 7 Internet Explorer v11



Windows 7 Firefox v68

Популярні за тиждень

Введіть назву фільму

Шукати

Диво-жінка 1984	Душа	Тенет	Опівнічне небо	Сімейка Крудсів: Нова ера
Тінь у хмарах	We Can Be Heroes	Чесний злодій	Гренландія	Відьми
Володар перснів: Хранителі персня	Зоряні війни: Епізод 4 - Нова надія	Месники: Завершення	Нові мутанти	Vørring 3
Skylines	Смерть 2020-му	Мулан	Перспективна дівчина	Гаррі Поттер і філософський камінь

© 2020 Bazan Tetyana, SN-41, TNTU

BrowserStack

Windows 10 Firefox v57