

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи ступеня  
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Бутенко Богдана Віталійовича*  
(ПІБ)

академічної групи *122-21ск-1*  
(шифр)

спеціальності *122 Комп'ютерні науки*  
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*  
(назва освітньої програми)

на тему: *Розробка веб-сайту бібліотеки фільмів та серіалів з використанням HTML, CSS, JavaScript*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Спірінцев В.В.</i>			
розділів:				
спеціальний	<i>доц. Спірінцев В.В.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро  
2024



## РЕФЕРАТ

Пояснювальна записка: 93 с., 25 рис., 2 табл., 30 джерел.

Об'єкт розробки: веб-орієнтований веб-сайт бібліотека фільмів та серіалів з використанням HTML, CSS, JavaScript.

Мета кваліфікаційної роботи: розробка веб-сайт бібліотека фільмів та серіалів з використанням HTML, CSS, JavaScript, який би допоміг користувачам відстежувати фільми чи серіали, які були переглянуті ними, та на яких сезонах був зупинений огляд серіалу.

У вступі проводиться аналіз та досліджується поточний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, робиться обґрунтування актуальності теми і формується постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформу для розробки, виконано проектування і розробку веб-орієнтованої інформаційної системи, описана робота системи, алгоритм і структура його функціонування, а також виклик та завантаження додатку, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню додатку та розраховано час на його створення.

Практичне значення полягає у створенні веб-сайту, що забезпечує: можливість переглядати та відстежувати каталог переглянутих фільмів, тобто полегшує запам'ятовування переглянутих фільмів та серіалів, а також дозволяє зберігати власні враження від переглянутих фільмів.

Актуальність розробки веб-орієнтованого додатку для користувача, який часто передивляється фільми чи серіали є вагомою, оскільки розробка веб-сайт може допомогти користувачу відстежувати та пам'ятати на яких частинах чи сезонах був зупинений перегляд того, або іншого відеоматеріалу.

Список ключових слів: ВЕБ САЙТ, HTML, CSS, JAVASCRIPT, ФІЛЬМИ, СЕРІАЛИ, ФІЛЬТРИ, ЛОКАЛІЗАЦІЯ.

## ABSTRACT

Explanatory note: 93 p., 25 figures, 0 appendix, 2 tables, 30 sources.

Object of development: a web-oriented website library of movies and TV series using HTML, CSS, JavaScript.

The purpose of the qualification work: to develop a web-based movie and TV series library using HTML, CSS, JavaScript, which would help users keep track of the movies or TV series they have watched and which seasons they have stopped watching.

The introduction analyzes and explores the current state of the problem, specifies the purpose of the qualification work and its scope, justifies the relevance of the topic, and formulates the task statement.

The first section analyzes the subject area, determines the relevance of the task and the purpose of the development, formulates the task statement, and specifies the requirements for software implementation, technologies and software tools.

The second section analyzes existing solutions, selects a development platform, designs and develops a web-based information system, describes the system's operation, algorithm and structure of its functioning, as well as the application call and download, defines input and output data, and characterizes the composition of technical means parameters.

In the economic section, the labor intensity of the developed information system is determined, the cost of creating the application is calculated, and the time for its creation is estimated.

The practical significance is to create a website that provides: the ability to view and track the catalog of watched movies, i.e. facilitates the memorization of watched movies and TV series, and allows you to save your own impressions of watched movies.

The relevance of developing a web-based application for a user who often re-watches movies or TV series is significant, since the development of a website can help the user to track and remember which parts or seasons they stopped watching a particular video.

List of keywords: WEBSITE, HTML, CSS, JAVASCRIPT, MOVIES, TV SHOWS, FILTERS, LOCALIZATION.

## ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ....	10
1.1. Загальні відомості з предметної галузі.....	10
1.1.1. Особливості електронної комерції у сучасних умовах України.....	11
1.1.2. Аналіз готових існуючих рішень.....	12
1.2. Призначення розробки та галузь застосування.....	14
1.3. Підстава для розробки.....	16
1.4. Постановка завдання.....	16
1.5. Вимоги до програми або програмного виробу.....	17
1.5.1. Вимоги до функціональних характеристик.....	17
1.5.2. Вимоги до інформаційної безпеки.....	18
1.5.3. Вимоги до складу та параметрів технічних засобів.....	19
1.5.4. Вимоги до інформаційної та програмної сумісності.....	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	21
2.1. Функціональне призначення системи.....	21
2.2. Опис застосованих математичних методів.....	22
2.3. Опис використаних технологій та мови програмування.....	23
2.3.1. Огляд використаної мови програмування.....	25
2.3.2. Огляд використаних технологій.....	27
2.4. Опис структури функціонування системи та алгоритмів.....	28
2.4.1. Структура і алгоритми системи.....	28
2.4.2. Структура бази даних.....	40

2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	42
2.6. Опис розробленої системи.....	42
2.6.1. Використані технічні засоби.....	42
2.6.2. Використані програмні засоби.....	43
2.6.3. Виклик та завантаження програми.....	44
2.6.4. Опис інтерфейсу користувача.....	45
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	52
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту....	52
3.2. Витрати на створення програмного забезпечення.....	55
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТОК А. Код програми.....	62
ДОДАТОК Б. Відгук керівника економічного розділу.....	92
ДОДАТОК В. Перелік файлів на диску.....	93

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

ПЗ – Програмне забезпечення

ІС – Інформаційна система

ОС – Операційна система

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

JS – JavaScript

ПК – Персональний комп'ютер

## ВСТУП

У сучасному світі цифровізація культурних та розважальних сервісів набуває все більшої актуальності. Особливо помітний прогрес у сфері медіа послуг, де веб-сайти та онлайн-платформи для перегляду фільмів та серіалів стають невід'ємною частиною повсякденного життя мільйонів користувачів. Розвиток таких платформ відіграє ключову роль у розширенні доступу до культурних продуктів та сприяє зростанню культурної обізнаності населення.

Створення веб-сайту бібліотеки фільмів та серіалів, що використовує сучасні веб-технології, такі як HTML, CSS і JavaScript, а також бібліотеку React, забезпечує нові можливості для зручного та інтерактивного доступу до великої колекції відеоконтенту. Це не лише сприяє збереженню культурної спадщини, але й відкриває нові горизонти для розвитку індустрії розваг.

Актуальність розробки веб-орієнтованої платформи для бібліотеки фільмів і серіалів полягає в наступному:

- Постійне зростання попиту на онлайн-доступ до кіно та серіалів.
- Необхідність інтеграції сучасних технологій для покращення користувацького досвіду.
- Можливість масштабування сервісу для забезпечення глобального доступу.

Об'єктом дослідження в даній роботі є веб-сайт бібліотеки фільмів і серіалів, розроблений з використанням React. Мета кваліфікаційної роботи полягає у створенні веб-орієнтованого додатку, що покращить доступ та взаємодію користувачів із багатим каталогом аудіовізуального контенту.

Для досягнення цієї мети були визначені наступні завдання:

- Аналіз ринку онлайн-платформ для перегляду фільмів та серіалів.
- Розробка вимог до програмного забезпечення, включаючи функціональність, інформаційну безпеку, інтерфейс користувача та забезпечення технічної сумісності.



- Проектування і розробка веб-сайту з урахуванням сучасних веб-стандартів та технологій.
- Розрахунок ресурсів та витрат на створення та підтримку сайту.

Мета кваліфікаційної роботи полягає в розробці веб-сайту для бібліотеки фільмів та серіалів, який використовує передові технології веб-розробки, зокрема HTML, CSS, JavaScript, і бібліотеку React. Цей сайт має забезпечити користувачам зручний та інтуїтивно зрозумілий інтерфейс для ефективного пошуку та перегляду контенту.

Практичне значення роботи полягає у створенні надійного та зручного веб-сервісу, який може значно розширити можливості доступу до культурного контенту та стимулювати культурне зростання серед широкої аудиторії.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

#### 1.1. Загальні відомості з предметної галузі

Веб-технології стали невід'ємною частиною сучасного світу, забезпечуючи користувачам доступ до інформації, розваг та комунікацій через інтернет. Однією з найпоширеніших галузей застосування веб-технологій є створення бібліотек фільмів та серіалів, що дозволяють користувачам зручно переглядати інформацію про відеоматеріали, читати рецензії, оцінки.

Сучасні веб-технології включають HTML (HyperText Markup Language), CSS (Cascading Style Sheets) та JavaScript. HTML дозволяє створювати структуру веб-сторінок, CSS забезпечує їх стилізацію, а JavaScript додає інтерактивність і динамічний контент. З розвитком технологій HTML5 та CSS3 можливості веб-розробки значно розширилися. HTML5 підтримує мультимедійні елементи, такі як відео та аудіо, що дозволяє створювати багатофункціональні веб-сайти. CSS3 додав нові можливості для анімації та оформлення, роблячи веб-сторінки ще більш привабливими.

Веб-технології є важливою складовою галузі 12 Інформаційні технології. Ця галузь охоплює різноманітні аспекти, включаючи розробку програмного забезпечення, системне програмування, аналіз даних, штучний інтелект, кібербезпеку та багато іншого. Веб-розробка, зокрема, є однією з ключових напрямків у підготовці фахівців з комп'ютерних наук, оскільки включає вивчення різноманітних мов програмування, фреймворків та бібліотек, які використовуються для створення інтерактивних та зручних веб-додатків.

Розвиток технологій продовжує змінювати спосіб організації та доступу до інформації. Зростання швидкості інтернет-з'єднань, впровадження нових стандартів веб-технологій та розвиток штучного інтелекту відкривають нові можливості для створення інтерактивних та зручних веб-сайтів. Сучасні дослідження в галузі комп'ютерних наук спрямовані на оптимізацію алгоритмів,

покращення безпеки веб-додатків та інтеграцію нових технологій для підвищення ефективності та зручності користувачів.

Таким чином, веб-технології продовжують еволюціонувати, забезпечуючи нові можливості для створення інтерактивних та зручних веб-сайтів, які відповідають сучасним вимогам користувачів. Веб-сайти для бібліотек фільмів і серіалів залишаються однією з найбільш популярних галузей застосування веб-технологій, надаючи користувачам зручний доступ до інформації про відеоматеріали.

### **1.1.1. Особливості використання веб-технологій у розважальній індустрії сучасної України**

З останніми технологічними змінами, спостерігається збільшення використання веб-технологій для доступу до розваг, таких як фільми та серіали. Це відображає глобальний тренд в цифровізації розважальної індустрії, де користувачі віддають перевагу доступу до контенту в будь-який час та з будь-якого місця. Із збільшенням використання мобільних пристроїв та поліпшенням швидкості інтернету, більше українців обирають онлайн-бібліотеки фільмів та серіалів як основне джерело розваг.

Веб-сайти, такі як розроблений в рамках цієї роботи, дозволяють користувачам фільтрувати фільми та серіали за жанрами, переглядати інформаційні картки та детальні описи, що значно покращує їхній досвід користування та задоволення від перегляду. Це також створює конкуренцію серед власників таких сайтів, змушуючи їх постійно оновлювати контент, поліпшувати інтерфейси та забезпечувати високу швидкість завантаження та відтворення.

З урахуванням війни, сектор онлайн-розваг також стикається з викликами, наприклад, з обмеженнями в електропостачанні, що може вплинути на доступність сервісів. Проте, веб-технології надають унікальні можливості для адаптації до цих умов: від використання енергоефективного програмного

забезпечення до оптимізації контенту для перегляду за обмежених умов.

Шляхи вирішення проблем веб-технологій у розважальній індустрії України включають:

- Забезпечення стабільності роботи веб-серверів під час перебоїв з електрикою, використовуючи автономні джерела живлення та хмарні рішення.
- Оптимізація веб-сайтів для швидкого завантаження навіть при низькій швидкості інтернету, адаптація контенту для мобільних пристроїв.
- Підвищення безпеки даних користувачів, використовуючи сучасні методи шифрування та захисту від кібератак.
- Співпраця з локальними інтернет-провайдерами для забезпечення надійної інтернет-підключення, особливо в регіонах, близьких до зони конфлікту.

### **1.1.2. Аналіз готових існуючих рішень**

Для аналізу були обрані популярні онлайн-платформи для перегляду фільмів та серіалів, такі як Netflix та Hulu. Основні особливості цих сайтів включають привабливий і сучасний дизайн, зручний користувацький інтерфейс, і високу функціональність, що дозволяє легко орієнтуватися та знаходити необхідний контент. На цих платформах реалізовані такі компоненти, як інформаційні банери з актуальними новинками, розділи з популярними фільмами та серіалами, а також наявність докладних карток кожного фільму чи серіалу з описом, рейтингом та іншою корисною інформацією. Зображення дизайну головних сторінок цих сайтів показані на рис.1.1 – рис.1.2.



Рис. 1.1. Картки в Netflix та зображення головного меню на різних девайсах

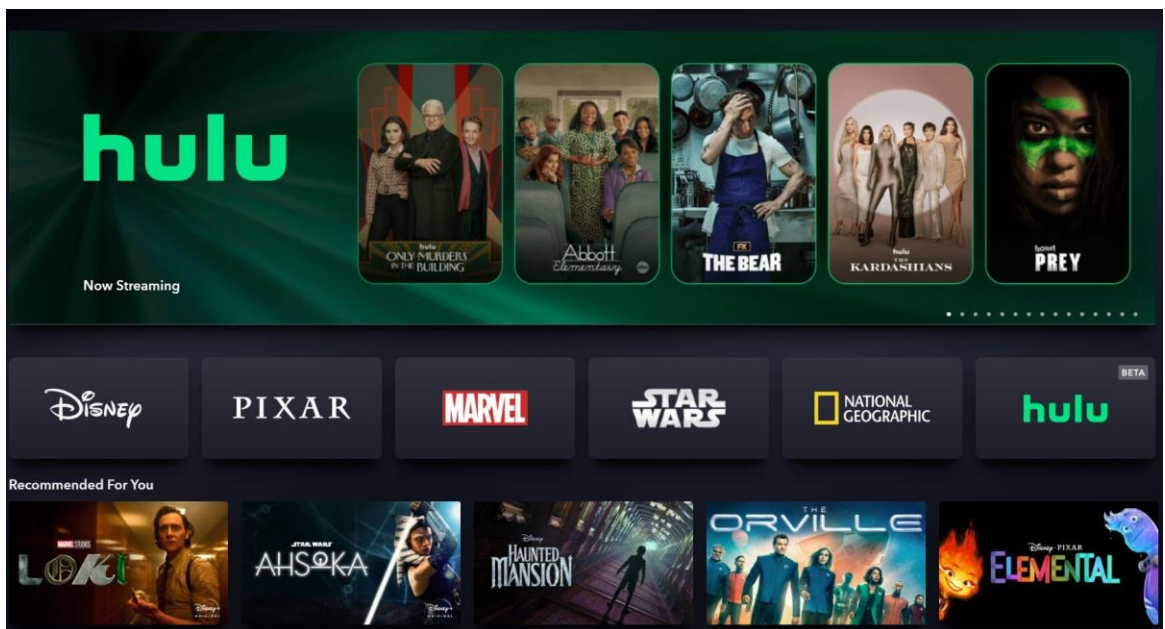


Рис. 1.2. Головне меню на Hulu

Основа сайту складається з карток фільмів та серіалів, також на сайтах реалізовані пошуки відеоматеріалу – це з основного функціоналу. Дуже зручно, що сайти пропонують рекомендовані матеріали та ті що вже розкидані по жанрам.

Отже, аналіз цих технічних рішень показав, що для обох існуючих систем притаманний такий набір компонентів: головна сторінка, секція з

відеоматеріалом та пошук.

Однак, аналізуючи цей набір даних був виявлений недолік, який полягає в тому, що в існуючих системах відсутній простий та доступний спосіб фільтрувати по декількома жанрам, а також відсутність можливості залишати свої відгуки, хоч функціонал реалізований таким способом, що сайт запам'ятовує де зупинився користувач. Тому, у межах цієї кваліфікаційної роботи, прийнято рішення про розробку інструментарію для користувача, який надав би змогу фільтрувати свою бібліотеку фільмів та залишати відгуки по своїм переглянутим фільмам.

На базі цього аналізу, проект кваліфікаційної роботи веб-сайту бібліотеки фільмів також включає наступні ключові компоненти:

- Головна сторінка з вітаючим вікном та ротацією випадково вибраних фільмів з бібліотеки.
- Навігаційні елементи у вигляді хедера та футера, що включають розділи з різними категоріями фільмів і серіалів, фільтри за жанрами та пошукову систему.
- Каталог із карточками фільмів, які відображають основну інформацію та зображення обкладинок.
- Механізм авторизації користувачів, що дозволяє зберігати персоналізовані налаштування та історію переглядів.
- Можливість додавання відгуків та рейтингів для кожного фільму чи серіалу.

## **1.2. Призначення розробки та галузь застосування**

У межах цієї кваліфікаційної роботи розглядається веб-сайт онлайн-бібліотеки фільмів та серіалів. Ця система застосовується у галузі онлайн-розваг та медіа-сервісів.

Призначенням веб-сайту є створення зручного і доступного способу для

користувачів відстежувати свої фільми та серіали, а також залишати свій відгук до відеоматеріалу. Веб-сайт бібліотеки фільмів та серіалів пропонує ряд переваг для користувачів.

Основні переваги включають:

- Доступність контенту 24/7, де користувачі можуть переглядати інформацію по переглянутим собою фільмам чи серіали.
- Простота інтерфейсу.
- Можливість додавати великий вибір фільмів і серіалів з можливістю фільтрації по різним жанрам, а також переглядати рейтинг залишений користувачем.

Розроблений веб-сайт призначений для:

- Забезпечення доступу до власної бібліотеки фільмів і серіалів в онлайн-режимі.
- Зручності та ефективності в перегляді інформації по медіаконтенту, де користувачі можуть легко знаходити та вибирати відеоматеріал за допомогою фільтрів.
- Можливість обирати найулюбленіші фільми чи серіали у своїй бібліотеці.

Веб-орієнтований додаток бібліотеки фільмів має виконувати такі функції як:

- Підтримка каталогу медіафайлів.
- Обробка запитів користувачів на перегляд фільмів і серіалів.
- Інтуїтивно зрозумілий інтерфейс для легкої навігації.

Аналіз існуючих рішень підкреслив важливість наступних аспектів для сучасного медіа-сервісу:

- Орієнтованість на клієнта, забезпечуючи легкий доступ до взаємодії зі своєю бібліотекою.
- Зберігання інформації, що включає деталі фільмів та відгук.
- Можливість локалізувати бібліотеку на зручну мову для

користувача, по стандарту на українську чи англійську.

Завдяки цим функціям, веб-сайт бібліотеки фільмів та серіалів стає ефективним інструментом для закриття потреб сучасних користувачів у відстеженні переглянутого контенту.

### **1.3. Підстава для розробки**

Відповідно до ОПП, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випусковою кафедрою, та затверджується наказом ректора. Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- ОПП за спеціальністю 122 «Комп'ютерні науки».
- Навчальний план та графік навчального процесу.
- Наказ ректора Національного технічного університету «Дніпровська політехніка» № 375-с від 23.04.2024 р.
- Завдання на дипломний проект на тему «Розробка веб-сайту бібліотеки фільмів та серіалів з використанням HTML, CSS, JavaScript».

### **1.4. Постановка завдання**

В даній кваліфікаційній роботі розглядається створення веб-орієнтованого додатку для бібліотеки фільмів та серіалів.

Мета цієї роботи: розробка веб-орієнтованого додатку, який би забезпечив зручний доступ до перегляду інформації про фільми та серіали через інтернет, використовуючи сучасні веб-технології HTML, CSS та JavaScript.

Інформаційна система створюється для надання зручного інтерфейсу для перегляду різноманітного медіаконтенту. Аудиторія веб-орієнтованого додатку становлять люди будь-якого віку, зацікавлені в фільмах та серіалах різних



жанрів.

Веб-орієнтований додаток повинен задовольняти наступним основним вимогам:

- Висока швидкість завантаження сторінок.
- Максимальна зручність роботи із системою для користувача.
- Легкість сприйняття інформації.
- Простота.
- Створення й розробка додатку включає наступні етапи:
- Аналіз вимог та обґрунтування необхідності: визначення функціональних та нефункціональних вимог до додатку, а також визначення його призначення та потенційної аудиторії.
- Проектування: складання технічного завдання, створення концепції та дизайну інтерфейсу додатку, розробка архітектури та вибір технологій.
- Розробка: програмування функціональності, розробка бази даних, тестування та відлагодження додатку.
- Підтримка та розвиток: надання технічної підтримки, оновлення та покращення функціональності додатку згідно з вимогами та потребами користувачів.

Головна сторінка має бути привабливою та функціональною, вона є першим, що постає перед користувачем. Основні елементи дизайну включають:

- Карусель з акційними пропозиціями та новими релізами.
- Секція з популярними фільмами та серіалами.
- Футер з інформацією про компанію та додатковими посиланнями.

## **1.5. Вимоги до програми або програмного виробу**

### **1.5.1. Вимоги до функціональних характеристик**

Для досягнення поставленої мети в розробці веб-сайту бібліотеки фільмів

та серіалів, повинні бути реалізовані наступні функціональні характеристики:

- Сучасні технології для розробки: використання HTML, CSS та JavaScript для створення динамічного та адаптивного веб-інтерфейсу.
- Можливість онлайн перегляду інформації по своїм переглянутим фільмам і серіалам: користувачі можуть легко переглядати і вибирати картки відеоматеріалів з різноманітних жанрів і категорій.
- Легкий пошук потрібних фільмів або серіалів: ефективні фільтри та пошукова система для швидкого доступу до потрібного контенту.
- Персональний кабінет користувачів: можливість додавати в обрані фільми, які сподобалися зі списку.
- Локалізація інтерфейсу користувача: підтримка декількох мов для забезпечення зручності користувачів з різних країн.
- Простий та інтуїтивно зрозумілий для користувача інтерфейс: забезпечення зручного навігаційного досвіду для користувачів усіх вікових категорій.

Користувач взаємодіє з сайтом через реєстрацію персонального акаунта, перегляд інформації наявних фільмів та серіалів, і можливість зберегти улюблені.

### **1.5.2. Вимоги до інформаційної безпеки**

Під терміном "інформаційна безпека" розуміється стан, в якому системи обробки та зберігання даних забезпечують конфіденційність, доступність та цілісність інформації. Це особливо важливо для веб-сайту, який надає доступ до великої кількості медіаконтенту, такого як фільми та серіали. Заходи, спрямовані на захист інформації від несанкціонованого доступу, використання, розголошення, пошкодження чи знищення, включають: валідація вхідних даних, тобто перевірка всіх даних, введених користувачами, для запобігання різних видів атак.

Ці критерії безпеки гарантують, що система використовується та розвивається в інтересах користувачів, надаючи їм безпечний доступ до контенту без ризику втрати особистої інформації або доступу до неї сторонніми особами.

### **1.5.3. Вимоги до складу та параметрів технічних засобів**

Для забезпечення нормального функціонування веб-сайту бібліотеки фільмів та серіалів, необхідно дотримуватись певних вимог до технічних засобів.

Для клієнтської частини:

- Інтернет-браузер: сучасні версії таких браузерів як Google Chrome, Mozilla Firefox, Safari, або Microsoft Edge.
- Доступ до мережі Інтернет: забезпечення стабільного і швидкого інтернет-з'єднання для безперебійного відображення інформації.
- Пристрої вводу: клавіатура та комп'ютерна миша або тачпад для ноутбуків та мобільних пристроїв.

Для серверної частини:

- Операційна система: сучасні версії Linux або Windows Server для оптимальної стабільності та безпеки.
- Оперативна пам'ять: мінімум 8 Гб для забезпечення ефективної обробки запитів користувачів.
- Вільний дисковий простір: мінімум 50 GB для зберігання бази даних, логів системи та резервних копій.
- Інтернет-з'єднання: мінімальна швидкість 1 Гбіт/с для підтримки високого трафіку користувачів та швидкої відповіді сервера.

### **1.5.4. Вимоги до інформаційної та програмної сумісності**

Для забезпечення ефективного функціонування веб-орієнтованої

підсистеми, програмне забезпечення обчислювальної машини, на якій буде розгорнуто веб-додаток, має відповідати вимогам для клієнтської та серверної частин.

Для клієнтської частини:

- Операційна система: будь-яка сучасна система, яка підтримує актуальні версії інтернет-браузерів.
- Інтернет-браузери: останні версії Google Chrome, Mozilla Firefox, Safari, або Microsoft Edge для оптимального відображення веб-інтерфейсу.

Для серверної частини:

- Операційна система: Linux або Windows Server, які забезпечують стабільне середовище для веб-серверів.
- Оперативна пам'ять: мінімум 4 Гб для оптимальної обробки запитів користувачів.
- Дисковий простір: мінімум 20 GB для розміщення веб-додатку.
- Мінімальна швидкість активного підключення до Інтернету: 1 Гбіт/с для забезпечення високої швидкості завантаження контенту.

Для розробки і налагодження:

- Програмне забезпечення: Використання інтегрованих середовищ розробки (IDE), таких як Visual Studio Code, Atom, або Sublime Text для редагування HTML, CSS і JavaScript коду.

Ці технічні вимоги забезпечать, що програмний продукт буде працювати надійно та ефективно, відповідаючи потребам замовника та кінцевих користувачів.

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 2.1. Функціональне призначення системи

Веб-сайт бібліотеки фільмів та серіалів розроблений для зручного перегляду та вибору відеоконтенту. Функціональне призначення цієї інформаційної системи включає наступне:

- Зберігання та управління базою даних фільмів та серіалів.
- Відображення вмісту бібліотеки на веб-сайті.
- Фільтрація фільмів та серіалів за жанрами.
- Пошук фільмів та серіалів за назвою.
- Можливість додавання фільмів та серіалів до вибраного для зручного доступу.
- Перегляд детальної інформації про фільми та серіали, такої як рейтинг та короткий опис.
- Можливість вказати сезон та епізод, на якому закінчився перегляд.
- Можливість додавання користувачами додаткової інформації до карток фільмів та серіалів.
- Авторизація та реєстрація користувачів.
- Можливість управління власним профілем, включаючи додавання та видалення фільмів та серіалів до вибраного.
- Вітаюче вікно з трьома випадковими фільмами або серіалами для заохочення користувачів переглядати новий контент.

Експлуатаційне призначення системи включає:

- Зручний та інтуїтивно зрозумілий інтерфейс для користувачів.
- Можливість використання на різних пристроях, включаючи мобільні телефони та планшети.
- Підтримка стандартів веб-розробки для забезпечення сумісності з

різними браузерами.

- Можливість зручного пошуку та фільтрації контенту.
- Автоматизоване управління додаванням та видаленням фільмів та серіалів.
- Можливість збереження стану перегляду для кожного користувача.

## **2.2. Опис застосованих математичних методів**

У процесі розробки веб-сайту бібліотеки фільмів та серіалів були застосовані різноманітні математичні методи та алгоритми для забезпечення ефективної фільтрації, пошуку та сортування контенту. Нижче описані основні методи, що використовувалися.

Однією з ключових функцій веб-сайту є можливість фільтрації фільмів та серіалів за жанрами. Для реалізації цієї функції була створена спеціальна система, яка визначає обрані жанри користувачем і відфільтровує відповідні фільми та серіали.

Суть методу фільтрації полягає у тому, що для кожного жанру на сторінці сайту є чекбокси. Коли користувач змінює стан чекбоксу, система аналізує обрані жанри і відображає лише ті фільми та серіали, які відповідають цим жанрам. Це забезпечує зручність і швидкість доступу до потрібного контенту, дозволяючи користувачам знаходити фільми та серіали, що відповідають їхнім вподобанням.

Іншою важливою функцією веб-сайту є пошук контенту за ключовими словами. Для реалізації цієї функції використовується метод, що аналізує назви та описи фільмів і серіалів на наявність заданих ключових слів.

Коли користувач вводить ключові слова у поле пошуку, система перевіряє кожен фільм і серіал на відповідність цим ключовим словам. Якщо назва або опис фільму містить ключові слова, цей фільм відображається у результатах пошуку. Такий підхід дозволяє швидко знаходити потрібний контент,

зменшуючи час пошуку і підвищуючи задоволеність користувачів.

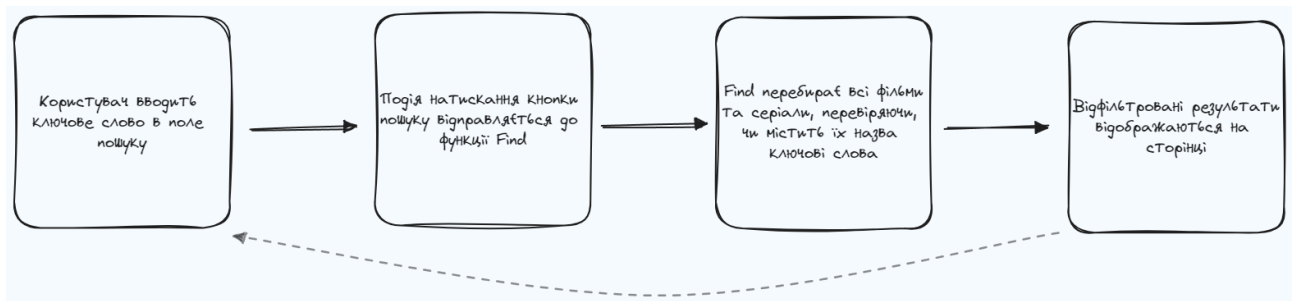


Рис. 2.1. Схема процесу пошуку контенту на сайті

Для покращення користувацького досвіду веб-сайт надає можливість сортування контенту за різними параметрами, такими як оцінка або дата випуску. Цей метод дозволяє впорядковувати фільми та серіали від найвищої до найнижчої оцінки або за хронологією випуску.

Наприклад, користувач може обрати сортування за оцінкою, і система автоматично впорядковує контент таким чином, що фільми з найвищими оцінками будуть відображені першими. Це полегшує вибір контенту для перегляду, забезпечуючи користувачам зручний доступ до найкращих фільмів та серіалів.

### 2.3. Опис використаних технологій та мови програмування

Для розробки веб-сайту бібліотеки фільмів та серіалів були використані наступні технології та мова програмування: JavaScript, HTML та CSS.

JavaScript був основною мовою для створення "frontend" частини додатку. Завдяки JavaScript вдалося реалізувати інтерактивний та динамічний інтерфейс користувача[3]. Ця мова програмування дозволила створити функції для фільтрації контенту, пошуку, сортування, динамічного завантаження даних та взаємодії з елементами інтерфейсу. JavaScript забезпечив гнучкість і функціональність додатку, дозволяючи користувачам зручно працювати з контентом бібліотеки.

Однією з основних функцій, реалізованих за допомогою JavaScript, є відображення списку фільмів. Користувачі можуть переглядати інформацію про фільми, включаючи зображення, назви, описи та рейтинг. Функція також дозволяє додавати фільми до списку обраного контенту, що забезпечує персоналізований підхід до взаємодії з веб-сайтом.

HTML (HyperText Markup Language) та CSS (Cascading Style Sheets) використовувалися для створення структури та оформлення веб-сторінок. HTML забезпечує базову структуру сторінок, визначаючи розташування тексту, зображень, посилань та інших елементів. CSS використовується для оформлення веб-сторінок, задаючи стилі для елементів HTML, такі як кольори, шрифти, відступи та розміри. Завдяки CSS вдалося створити привабливий та сучасний дизайн веб-сайту, що забезпечує зручність і естетичну привабливість для користувачів.

Для прикладу, основна структура веб-сторінки включає заголовок, основний контент і підвал. У заголовку розміщується вітальний текст, у основному контенті - список фільмів, а в футері - контактна інформація. Стили CSS застосовуються для налаштування зовнішнього вигляду кожного з цих елементів, забезпечуючи єдиний та професійний вигляд сторінки.

JavaScript також використовувався для реалізації функціональності пошуку та сортування контенту. Пошук дозволяє користувачам швидко знаходити потрібні фільми та серіали за ключовими словами. Сортування контенту дозволяє впорядковувати фільми за різними критеріями, такими як оцінка або дата випуску, що полегшує вибір контенту для перегляду[4].

Для роботи з даними та забезпечення динамічного відображення контенту була реалізована функціональність імітування роботи з сервером, працюючи з внутрішніми файлами проекту. Це дозволило зберігати і обробляти дані про фільми та серіали, включаючи їхні назви, описи, зображення, жанри, сезони та рейтинги.

Мова програмування та технології були вибрані через їхню популярність,



ефективність, доступність та можливість швидкої розробки веб-додатків. JavaScript як для “frontend”, так і для забезпечення динамічної взаємодії з користувачами забезпечує узгодженість та спрощує процес розробки. HTML та CSS є стандартними технологіями для створення веб-сторінок, що забезпечує сумісність з усіма сучасними браузерами. Використання лише клієнтських технологій дозволяє спростити архітектуру додатку та забезпечити високу швидкість розробки.

Таким чином, використання JavaScript, HTML та CSS дозволило створити ефективний, зручний та привабливий веб-сайт бібліотеки фільмів та серіалів, що відповідає сучасним вимогам користувачів.

### **2.3.1. Огляд використаної мови програмування**

У реалізації веб-сайту бібліотеки фільмів та серіалів використовується мова програмування JavaScript для "frontend" частини додатку. JavaScript дозволяє створювати інтерактивні та динамічні веб-сайти, що забезпечує більшу зручність користувачам у взаємодії з веб-сторінками.

JavaScript був вибраний для цього проекту через його широкі можливості та популярність у веб-розробці. Він дозволяє створювати оновлення контенту, інтерактивні елементи, анімацію та багато іншого, що робить веб-сайт більш привабливим для користувачів. JavaScript є основною мовою програмування для створення інтерактивного та динамічного інтерфейсу користувача, що дозволяє забезпечити швидкий відгук системи на дії користувача без необхідності перезавантажувати сторінку.

JavaScript забезпечує можливість реалізації складної функціональності на веб-сторінках. Завдяки JavaScript можна створювати оновлення контенту, інтерактивні карти, анімовану 2D/3D графіку, музичні плеєри з відео тощо. За допомогою JavaScript була реалізована функціональність фільтрації та пошуку контенту, динамічне завантаження даних, взаємодія з елементами інтерфейсу, а

також локалізація сайту.

JavaScript був введений у 1995 році як засіб для вбудовування програм у веб-сторінки в браузері Netscape Navigator. З того часу інші популярні графічні веб-браузери також прийняли цю мову. Це сприяло появі сучасних веб-програм, з якими можна взаємодіяти безпосередньо, без необхідності перезавантажувати сторінку після кожної дії. JavaScript використовується на традиційних веб-сайтах для надання різноманітної інтерактивності та інтелектуальних можливостей.

Оскільки мова JavaScript постійно розвивається, браузері мають постійно оновлюватись, щоб підтримувати нові функції. Старі версії браузерів можуть не підтримувати всі сучасні можливості. Розробники мови стараються уникати внесення змін, які можуть порушити роботу існуючих програм, тому нові браузері здатні запускати старі програми. JavaScript є кросплатформною мовою, що підтримується всіма сучасними веб-браузерами, включаючи Google Chrome, Firefox, Safari та Edge.

JavaScript використовується не лише у веб-браузерах. Деякі бази даних, такі як MongoDB і CouchDB, використовують JavaScript як мову сценаріїв і запитів. Кілька фреймворків програмування для робочих столів і серверів, зокрема Node.js, надають середовище для виконання JavaScript поза межами браузера. У даному проекті використання JavaScript було обмежено тільки клієнтською частиною для створення інтерактивного інтерфейсу та забезпечення функціональності веб-сайту.

У процесі розробки веб-сайту бібліотеки фільмів та серіалів JavaScript був використаний для реалізації різних аспектів функціональності:

- Фільтрація контенту: за допомогою методу `Array.prototype.filter` були реалізовані функції фільтрації фільмів та серіалів за жанрами.
- Пошук контенту: реалізація пошуку за ключовими словами за допомогою методу `String.prototype.includes`.
- Динамічне завантаження даних: оновлення вмісту сторінки без перезавантаження.

- Взаємодія з елементами інтерфейсу: обробка подій кліків, змін стану елементів та інші взаємодії користувача з веб-сторінкою.
- Локалізація сайту: забезпечення підтримки декількох мов інтерфейсу, включаючи англійську та українську, шляхом зберігання текстових даних у відповідних об'єктах та динамічної зміни мови інтерфейсу.

JavaScript є потужним інструментом у веб-розробці, що дозволяє створювати сучасні та ефективні веб-додатки з багатьма інтерактивними можливостями для користувачів. Завдяки використанню JavaScript вдалося створити ефективний та зручний веб-сайт для перегляду та управління фільмами та серіалами.

### **2.3.2. Огляд використаних технологій**

Для розробки веб-сайту бібліотеки фільмів та серіалів були використані наступні технології: HTML (HyperText Markup Language), CSS (Cascading Style Sheets) та JavaScript.

HTML є стандартною мовою розмітки для створення веб-сторінок. Вона використовується для створення структури веб-документів за допомогою тегів. HTML дозволяє визначати заголовки, абзаци, списки, таблиці, посилання та інші елементи веб-сторінки. Використовуючи HTML, розробник може створити структурований та доступний контент для користувачів. HTML також підтримує мультимедійні елементи, такі як зображення, відео та аудіо, що дозволяє створювати багатofункціональні та інтерактивні веб-сторінки.

CSS використовується для оформлення веб-сторінок, надаючи їм вигляд і стиль. За допомогою CSS можна задавати кольори, шрифти, розміри, відступи, рамки та інші властивості елементів веб-сторінки. CSS дозволяє розділити вміст сторінки від її представлення, що спрощує управління виглядом веб-документів та забезпечує їх консистентність на різних пристроях та браузерах. CSS також

підтримує медіа-запити, які дозволяють створювати адаптивні веб-сторінки, що змінюють свій вигляд в залежності від розміру екрану пристрою.

JavaScript є скриптовою мовою програмування, яка використовується для створення інтерактивності на веб-сторінках. Вона дозволяє реалізувати різноманітні функції, такі як валідація форм, анімація, взаємодія з користувачем та зміна вмісту сторінки без перезавантаження. JavaScript є невід'ємною частиною сучасних веб-додатків, яка додає їм динамічність та інтерактивність [10].

Однією з основних функцій, реалізованих за допомогою JavaScript, є фільтрація контенту. Коли користувач обирає певні жанри, система аналізує обрані жанри і відображає лише ті фільми та серіали, які відповідають цим жанрам. Цей метод дозволяє користувачам зручно фільтрувати вміст бібліотеки за їхніми уподобаннями.

Використання HTML, CSS та JavaScript у розробці веб-сайту бібліотеки фільмів та серіалів дозволило створити сучасний та зручний для користувачів інтерфейс. Це забезпечує швидкий та зручний доступ до вмісту сайту, дозволяючи користувачам легко знаходити та переглядати фільми та серіали. HTML забезпечує структуру веб-сторінок, CSS надає їм естетичний вигляд, а JavaScript додає інтерактивність і динамічність, що робить веб-сайт привабливим та функціональним.

## **2.4. Опис структури функціонування системи та алгоритмів**

### **2.4.1. Структура і алгоритми системи**

Система веб-орієнтованого додатку бібліотеки фільмів та серіалів складається з таких основних елементів: розмітка сторінки, стилі, файли, що об'єднують функціональність сайту, база даних фільмів, жанрів та користувачів, а також модулі логіки проекту.

Розмітка сторінки та стилі:

- `index.html`: Основний HTML-документ, що визначає структуру веб-сторінки.
- `style.css`: Файл CSS, що забезпечує стилізацію веб-сторінки.

Файли, що об'єднують функціональність сайту:

- `script.js`: Основний файл JavaScript, що містить логіку взаємодії з користувачем.

База даних фільмів, жанрів та користувачів. Компоненти:

- `data.js`: Файл, що містить дані про фільми та серіали.
- `dataInputs.js`: Файл, що містить дані про жанри.
- `users.js`: Файл, що містить дані про користувачів.

Модулі логіки проекту:

- `generateButtonAction.js`: Модуль, що обробляє дії кнопок.
- `generateCards.js`: Модуль, що генерує картки фільмів та серіалів.
- `generateInputs.js`: Модуль, що обробляє введення користувачів.
- `LanguageSwitcher.js`: Модуль, що забезпечує зміну мови інтерфейсу.
- `loginModal.js`: Модуль для обробки вікна входу.
- `registerModal.js`: Модуль для обробки вікна реєстрації.
- `SearchAndFilters.js`: Модуль, що забезпечує функціональність пошуку та фільтрації.

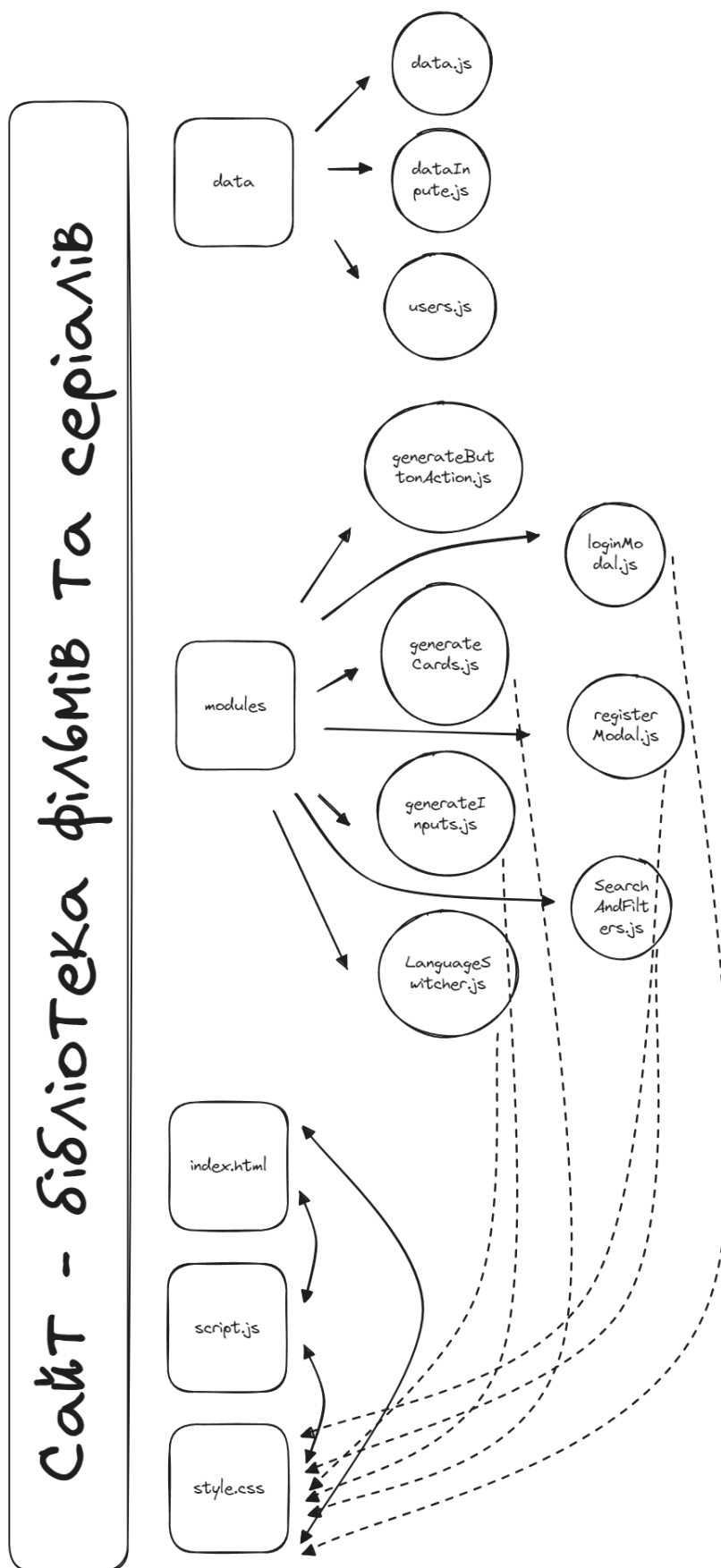


Рис. 2.2. Схема проекту та взаємодії основних файлів між собою  
 Веб-орієнтований додаток бібліотеки фільмів та серіалів є динамічним,  
 реалізованим за допомогою HTML, CSS та JavaScript. Особливість цього проекту

полягає в тому, що JavaScript оновлює лише необхідні компоненти, а не всю сторінку при дії користувача на сайті, що знижує навантаження на систему та підвищує швидкість роботи.

Основна навігація по сайту виконується за допомогою системи пошуку та фільтрації, що реалізовано наступним кодом:

```
import { renderData, showingFavorites, showFavoriteCards } from
'./generateCards.js';
import { filterData } from './generateInputs.js';
import { data as defaultData } from '../data/data.js';

const searchButton = document.querySelector('#search-button');
const searchText = document.getElementById('text-to-find');
const clearSearchButton = document.getElementById('clear-search');

searchButton.addEventListener('click', Find);

searchText.addEventListener('keydown', (event) => {
  if (event.key === 'Enter') Find();
});

clearSearchButton.addEventListener('click', () => {
  searchText.value = '';
  const checkboxes = document.querySelectorAll('.searchNav
input[type="checkbox"]');
  checkboxes.forEach(checkbox => {
    checkbox.checked = false;
    sessionStorage.setItem(checkbox.id, 'false');
  });
  const cards = document.querySelectorAll('.card');
  cards.forEach(card => card.style.display = '');

  if (showingFavorites) {
    const [currentFilters] = filterData();
    showFavoriteCards(currentFilters);
  } else {
    renderData(defaultData);
  }
});

function Find() {
  const searchQuery = searchText.value.trim().toLowerCase();
  const cards = document.querySelectorAll('.card');

  cards.forEach(card => {
```

```

const content = card.textContent || card.innerHTML;
card.style.display = content.toLowerCase().includes(searchQuery) ? '' :
'none';
});
}

```

Веб-орієнтований додаток бібліотеки фільмів та серіалів побудований по принципу роботи лише “frontend” частини, тобто без “backend” частини. Той код, який має працювати як серверна частина – реалізований за допомогою JavaScript.

Для розробки інтерфейсу користувача було використано бібліотеку HTML та CSS, а також сам дизайн був сконфігурований у додатку Figma. Алгоритм функціонування обміну даних між сайтом та користувачем виконується за допомогою записування інформації у об’єкт масивів.

Як було зазначено на початку розділу, веб-сайт складається з компонентів списку карток відеоматеріалів (data.js) та списку фільтрів (dataInput.js). Ці файли поєднані у файлі generateInputs.js, який прив’язує дію (action) пошуку відеоматеріалів по відомим жанрам.

Функція фільтрації контенту дозволяє користувачам вибирати певні жанри та відображати лише ті фільми та серіали, які відповідають обраним жанрам. Цей процес реалізовано за допомогою подій зміни стану чекбоксів (checkboxes) для кожного жанру. При зміні стану чекбоксу функція filterData обробляє обрані жанри та фільтрує дані відповідно до цих жанрів.

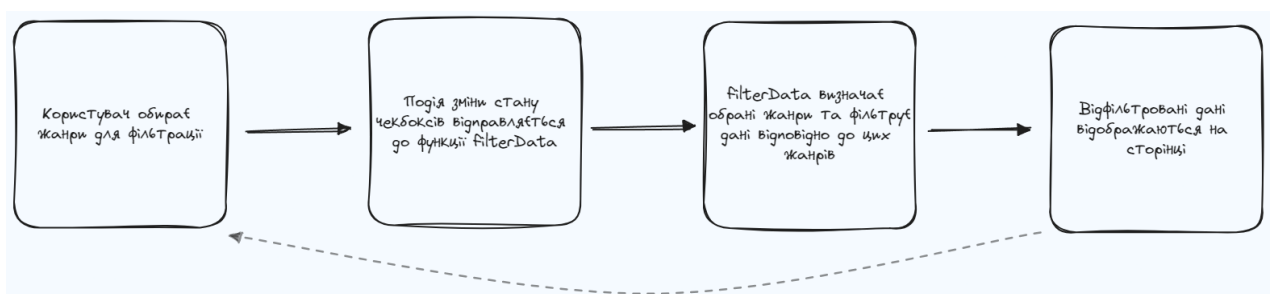


Рис. 2.3. Схема процесу фільтрації контенту фільмів та серіалів на сайті





Рис. 2.4. Фільтри та результат пошуку

Приклад роботи з даними при фільтрації:

```
inputs.forEach((input) => {
  input.onChange = (event) => {
    sessionStorage.setItem(input.id, event.target.checked.toString());
    const [filters, filteredData] = filterData();
    if (showingFavorites) {
      showFavoriteCards(filters);
    } else {
      renderData(filters.length ? filteredData : data);
    }
  }
});
```

У результаті призначення тих чи інших фільтрів ми отримуємо результат списку відеоматеріалів, які мають хоча б один з перелічених жанрів. Фільтри жанрів, які є на сайті:

- Drama;
- Triller;
- Fantastic;
- Horror;
- Comedia;
- Action;
- Detective;

- Criminal.

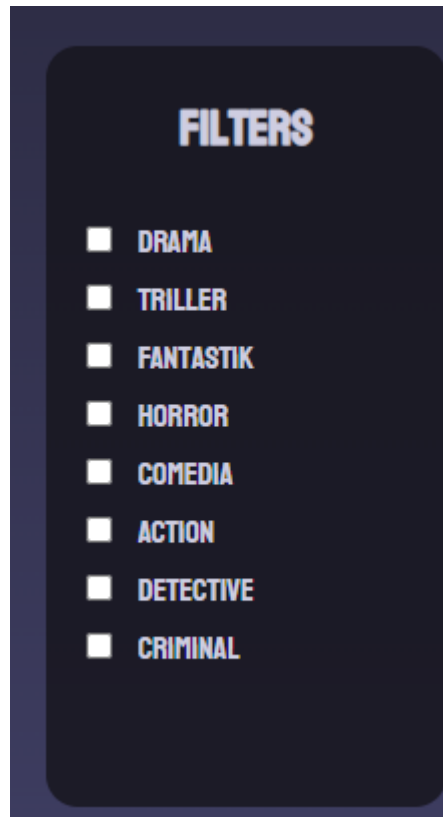


Рис. 2.5. Фільтри

Натиснувши на цікаву користувачу карту з контентом, відбувається детальний рендер цієї карточки, і зона огляду користувача оновлюється, де він вже бачить у масштабі картку. Основна інформація, яка відображається у детальному перегляді контенту, включає:

- Назва відеоматеріалу.
- Оцінка відеоматеріалу від 1 до 10.
- Сезон або серія, на якій було зупинено перегляд матеріалу.
- Опис відеоматеріалу.
- Інтро заставка відеоматеріалу.

Ця інформація дає користувачу уявлення про відеоматеріал який він збирається подивитися. Приклад коду для відображення детальної інформації:

```
card.onclick = function () {
```

```

cards.innerHTML = '';

let cardDetail = document.createElement('div');
cardDetail.classList.add('cardInfo');
cardDetail.id = el.id;

let createImgDetail = document.createElement('img');
createImgDetail.src = el.imgSrc;

let createH1Detail = document.createElement('h1');
createH1Detail.innerHTML = currentLanguage === 'ua' ? el.titleUA :
el.title;
createH1Detail.classList.add('specialH1');

let createPDetail = document.createElement('p');
createPDetail.innerHTML = currentLanguage === 'ua' ? el.descriptionUA :
el.description;

let rating = document.createElement('h2');
rating.innerHTML = el.rating;
rating.classList.add('rating');
if (el.rating > 8) rating.classList.add('green');
else if (el.rating > 5) rating.classList.add('orange');
else rating.classList.add('red');

let season = document.createElement('h2');
season.classList.add('season');
season.innerHTML = texts[currentLanguage].seasonStopped + el.season;

let textWithImg = document.createElement('div');
textWithImg.classList.add('wrapperforimg');
textWithImg.appendChild(createPDetail);
textWithImg.appendChild(createImgDetail);

let backButton = document.createElement('button');
backButton.classList.add('back');
backButton.onclick = () => {
  showCards();
  switchLanguage(currentLanguage);
};

let fontBack = document.createElement('i');
fontBack.classList.add('fa-angle-left', 'fa-solid');

let text = document.createElement('h3');
text.innerHTML = 'Back';

backButton.appendChild(fontBack, text);

cardDetail.appendChild(createH1Detail, rating, season, textWithImg);

cards.appendChild(backButton, cardDetail);

switchLanguage(currentLanguage);
};

```

Потрапляючи на сайт, користувача одразу зустрічають у правому верхньому кутку кнопки реєстрації та логіну.

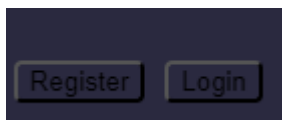


Рис. 2.6. Кнопки реєстрації та логіну

Якщо у користувача ще немає облікового запису, то йому потрібно пройти реєстрацію. Натиснувши кнопку реєстрації, відкривається форма, де потрібно ввести необхідну інформацію, а саме Username та Password (має бути символами латинського алфавіту та містити одну велику букву та цифру).

A registration form titled 'REGISTER' in bold black text. Below the title are two input fields: 'USERNAME:' followed by a text box, and 'PASSWORD:' followed by a text box. To the right of the password field is a red asterisk and the text '\*THE PASSWORD MUST BE AT LEAST 6 CHARACTERS LONG AND CONTAIN AT LEAST ONE LETTER.' At the bottom left of the form are two buttons: 'Register' and 'Back'.

Рис. 2.7. Форма реєстрації

Після успішної реєстрації потрібно перейти до логіну облікового запису, натиснувши кнопку логіну. Після чого з'явиться кнопка особистого кабінету.



Рис. 2.8. Кнопка особистого кабінету

Також будуть оновлені всі картки, так як на них з'являються кнопки зірочки, які відповідають за додавання картки у вибране користувача.



Рис. 2.9. Вигляд картки після логіну в обліковий запис

Користувач може обрати цікаві для нього фільми та відстежувати їх у своєму особистому кабінеті, натиснувши відповідну кнопку. Після чого будуть відображені вибрані картки відеоматеріалів з можливістю видалення.



Рис. 2.10. Вигляд картки в особистому кабінеті

Користувач може видаляти ці картки при необхідності, а також переглядати зміст картки, як і у головному меню. Приклад коду для обробки реєстрації та логіну:

```
const registerButton = document.getElementById('register-button');
const loginButton = document.getElementById('login-button');

registerButton.addEventListener('click', () => {
  // Відкриття модального вікна реєстрації
});

loginButton.addEventListener('click', () => {
  // Відкриття модального вікна логіну
});
```



Рис. 2.11. Схема процесу реєстрації, логіну та взаємодії з вибраним списком

## користувача

Сайт може бути локалізований на дві мови: англійська та українська. По стандарту сайт отримує мову з браузера та локалізує сайт на українську чи англійську. Але користувач самостійно може обрати мову, при наведенні на глобус у правому верхньому кутку.

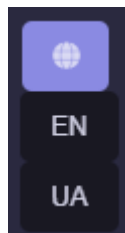


Рис. 2.12. Переключення локалізації сайту

Приклад коду для локалізації:

```
const languageSwitcher = document.getElementById('language-switcher');  
  
languageSwitcher.addEventListener('click', () => {  
  const currentLanguage = document.documentElement.lang;  
  const newLanguage = currentLanguage === 'en' ? 'ua' : 'en';  
  document.documentElement.lang = newLanguage;  
  updateTexts(newLanguage);  
});
```

Таким чином, структура і алгоритми системи забезпечують зручну та інтуїтивну взаємодію користувачів з веб-сайтом бібліотеки фільмів та серіалів. Користувачі можуть легко шукати та фільтрувати контент, переглядати детальну інформацію про відеоматеріали, додавати улюблені фільми до списку вибраного та змінювати мову інтерфейсу відповідно до своїх уподобань. Завдяки використанню сучасних технологій HTML, CSS та JavaScript, веб-сайт забезпечує високу продуктивність та інтерактивність, що значно покращує користувацький досвід.

## 2.4.2. Структура бази даних

Для зберігання інформації про користувачів, фільми, серіали та обраний контент була створена база даних, що включає кілька таблиць. Основні таблиці бази даних та їх структура наведені нижче.

Таблиця 2.1.

### Структура інформації про користувачів веб-сайту

Поле	Тип даних	Опис
username	VARCHAR	Ім'я користувача
password	VARCHAR	Пароль користувача
favorites	ARRAY	Улюблені фільми користувача



Таблиця 2.2.

**Структура інформації про фільми та серіали**

Поле	Тип даних	Опис
title	VARCHAR	Назва відеоматеріалу на англійській
titleUA	VARCHAR	Назва відеоматеріалу на українській
imgSrc	VARCHAR	Посилання на зображення
genres	ARRAY	Жанри відеоматеріалу
id	VARCHAR	Ідентифікатор відеоматеріалу
season	INT	Сезон, на якому було зупинено перегляд
rating	INT	Рейтинг відеоматеріалу
description	TEXT	Опис на англійській мові
descriptionUA	TEXT	Опис на українській мові

## **2.5. Обґрунтування та організація вхідних та вихідних даних програми**

Вхідні дані:

- Дані користувачів (логін, пароль, електронна пошта).
- Інформація про фільми та серіали (назва, опис, жанр, оцінка, зображення, сезон).
- Запити користувачів на фільтрацію та пошук контенту.

Вихідні дані:

- Інформація про знайдені фільми та серіали, що відповідають запитам користувачів.
- Списки обраного контенту для кожного користувача.
- Детальна інформація про конкретний фільм або серіал.

Для забезпечення коректної роботи системи необхідно організувати обробку вхідних даних та формування вихідних даних таким чином, щоб забезпечити швидкий доступ до інформації та високу продуктивність.

## **2.6. Опис розробленої системи**

### **2.6.1. Використані технічні засоби**

Для розробки веб-сайту використовувалися наступні технічні засоби:

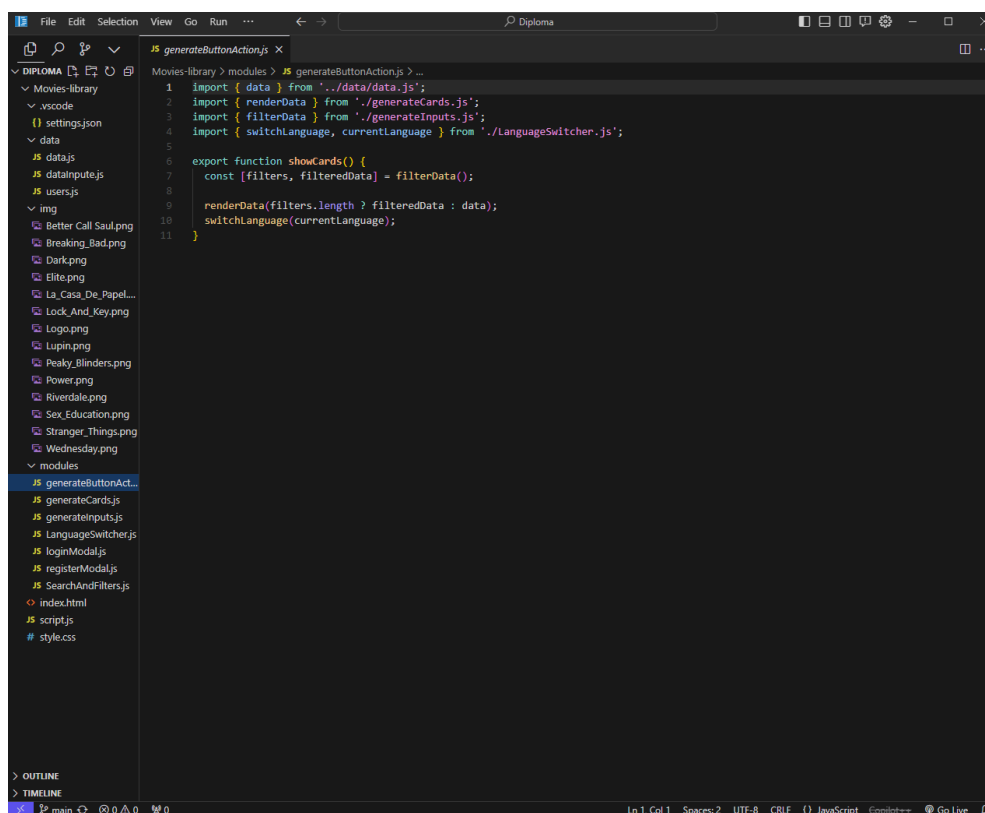
- Комп'ютер з ОС Windows 10.
- Інтернет-браузери для тестування (Google Chrome, Firefox).
- Інструменти розробки (Visual Studio Code, Node.js).

## 2.6.2. Використані програмні засоби

Під час розробки цього застосунку були використані наступні програмні засоби:

- Visual Studio Code;
- Figma.

Visual Studio Code (рис. 2.13) – це редактор коду з відкритим вихідним кодом, розроблений компанією Microsoft, який спеціально оптимізований для розробки та налагодження сучасних веб- та хмарних програм. Він представляє собою легкий, але потужний редактор вихідного коду, доступний для операційних систем Windows, macOS і Linux. У ньому реалізовано багато корисних функцій, включаючи підтримку налагодження, підсвічування синтаксису, інтелектуальне автодоповнення, фрагменти коду, можливість рефакторингу коду та вбудований Git. Крім того, Visual Studio Code надає можливість встановлення розширень, що дозволяють додавати нові мови програмування, теми і інтеграцію з додатковими сервісами.



## Рис. 2.13. Visual Studio Code

Figma – це інструмент для спільного проектування інтерфейсу користувача, який працює в браузері і дозволяє користувачам спільно працювати над створенням яскравих та інтерактивних прототипів. З моменту свого випуску в 2016 році Figma стала популярним рішенням як у сфері веб-дизайну, так і серед онлайн-спільнот. Вона дозволяє користувачам співпрацювати, ділитися шаблонами, дизайнами та віджетами з мільйонами користувачів по всьому світу.

### 2.6.3. Виклик та завантаження програми

Для забезпечення коректної роботи веб-сайту бібліотеки фільмів та серіалів необхідно дотримуватися певної послідовності дій під час виклику та завантаження програми. Програма запускається за допомогою live server через IDE (у нашому випадку Visual Studio Code). Цей підхід забезпечує швидкий і зручний запуск додатку, автоматично підключаючи всі необхідні скрипти та стилі.

Після встановлення розширення "Live Server" у Visual Studio Code, необхідно відкрити головну сторінку index.html у редакторі. Для цього можна скористатися наступними кроками:

1. Відкрити Visual Studio Code і відкрити папку з проектом.
2. Вибрати файл index.html і відкрити його у редакторі.
3. Натиснути правою кнопкою миші на відкритий файл index.html і вибрати опцію "Open with Live Server".

Ці дії дозволяють автоматично запуснути локальний сервер і відкрити веб-сторінку у веб-браузері. Всі необхідні скрипти та стилі будуть підключені автоматично, що забезпечить правильне завантаження та функціонування веб-додатку.

Під час завантаження сторінки, веб-браузер виконує наступні дії:

1. Завантажує та обробляє HTML-код для побудови структури

сторінки.

2. Підключає CSS-файл для стилізації елементів на сторінці.
3. Завантажує та виконує JavaScript-код, що забезпечує динамічну взаємодію з користувачем.

Використання live server дозволяє не лише спростити процес запуску додатку, але й забезпечує автоматичне оновлення сторінки при внесенні змін у файли проекту. Це особливо зручно під час розробки, оскільки дозволяє швидко бачити результати внесених змін у реальному часі без необхідності вручну оновлювати сторінку у веб-браузері.

#### **2.6.4. Опис інтерфейсу користувача**

Інтерфейс користувача включає наступні основні елементи:

- Головна сторінка з привітанням та мовним перемикачем.
- Секція пошуку з полем для введення пошукового запиту та кнопкою для запуску пошуку.
- Секція фільтрів для вибору жанрів фільмів та серіалів.
- Список карток фільмів та серіалів з можливістю перегляду детальної інформації.
- Кнопки для реєстрації та авторизації користувачів.
- Кабінет користувача з можливістю управління обраними фільмами.

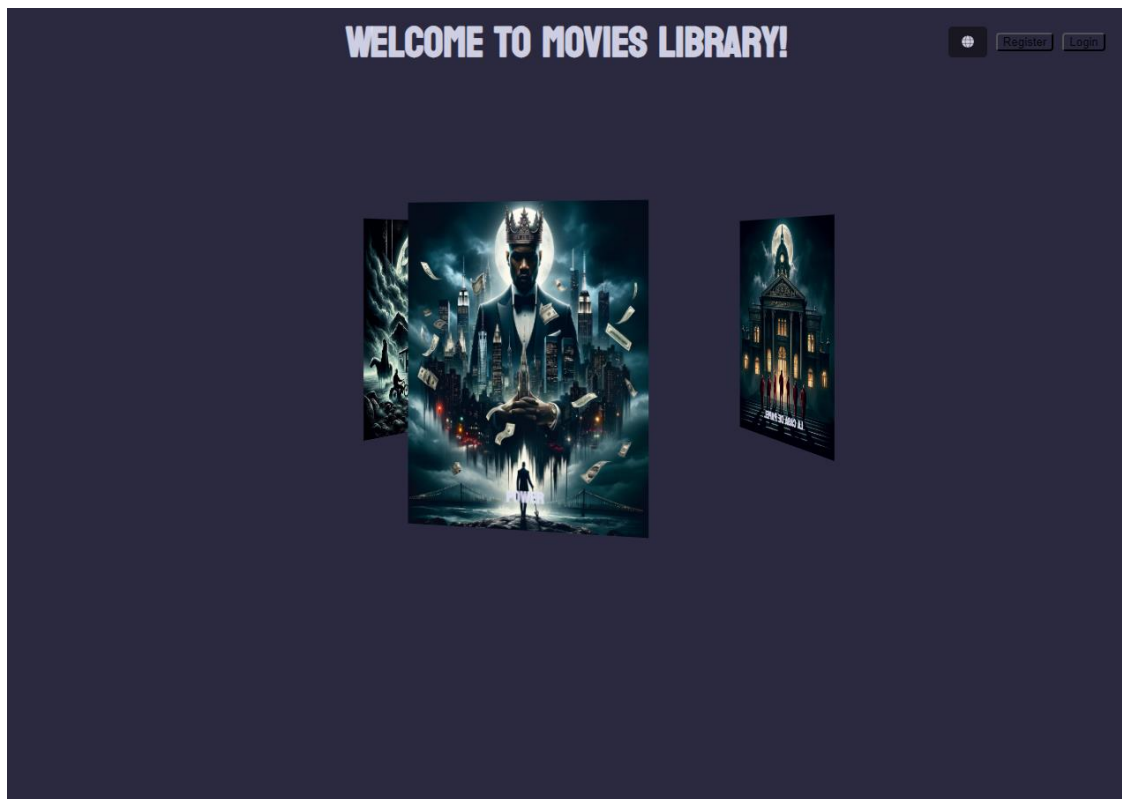


Рис. 2.14. Екран привітання

Екран привітання містить у своїй структурі карточки, які мають анімацію каруселі, меню локалізації та кнопки реєстрації та логіну.

**REGISTER**

USERNAME:

PASSWORD:  \*THE PASSWORD MUST BE AT LEAST 6 CHARACTERS LONG AND CONTAIN AT LEAST ONE LETTER.

Рис. 2.15. Форма реєстрації

Локалізацію сайту можна змінювати у «хедері» (рис. 2.12). Вибір цього параметру вплине на мову тексту.

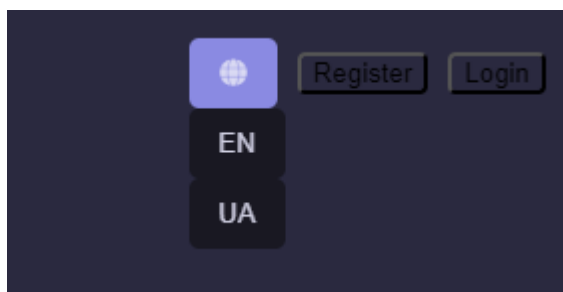


Рис. 2.16. Вибір локалізації

На сайті реалізована можливість реєстрації та авторизації користувача до системи (рис. 2.12). Якщо користувач авторизований, у «хедері» сайту можна побачити значок користувача за натисканням на нього можна відкрити обраний список улюблених відеоматеріалів користувача.

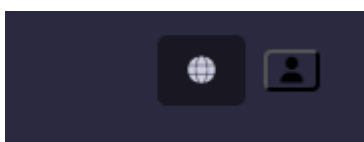


Рис. 2.17. Кнопка особистого профілю

Основною секцією сайту є зона з картками відеоматеріалів, фільтром по жанрам то зоною пошуку.

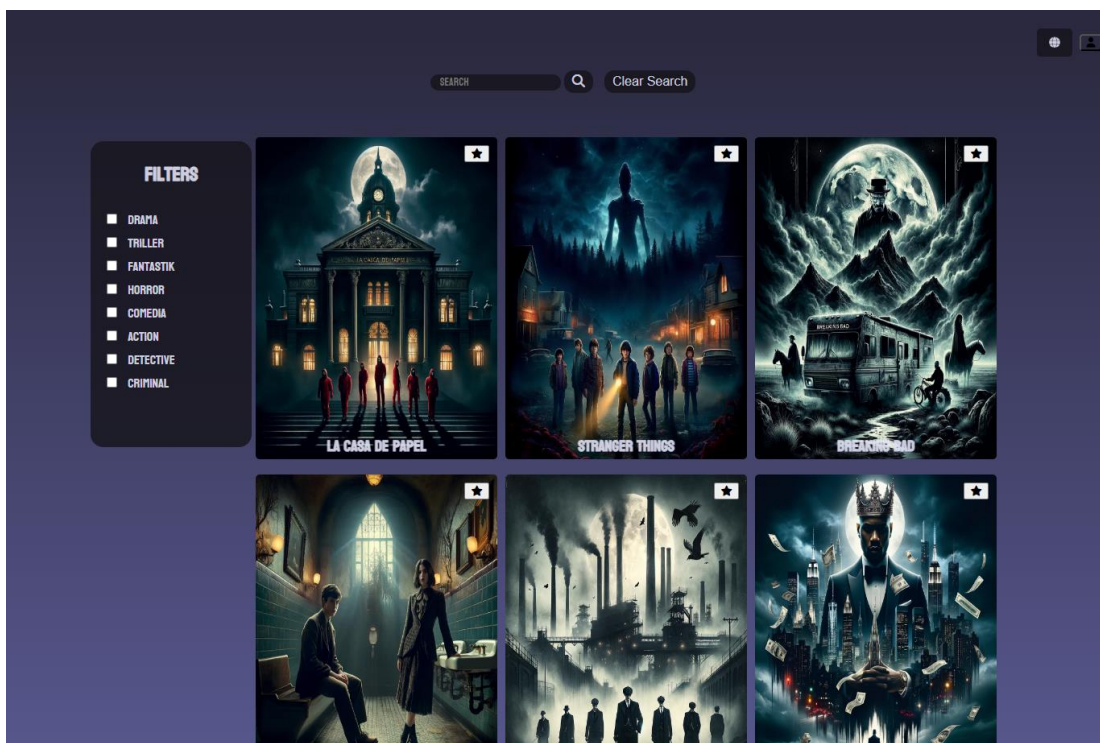


Рис. 2.18. Основна секція сайту

У основній секції є зона фільтрів, яка відповідає за фільтрування карток по відповідним жанрам. Тобто обравши Comedia та Action будуть відображатися лише ті карти, які відповідають хоча б одному з цих жанрів.

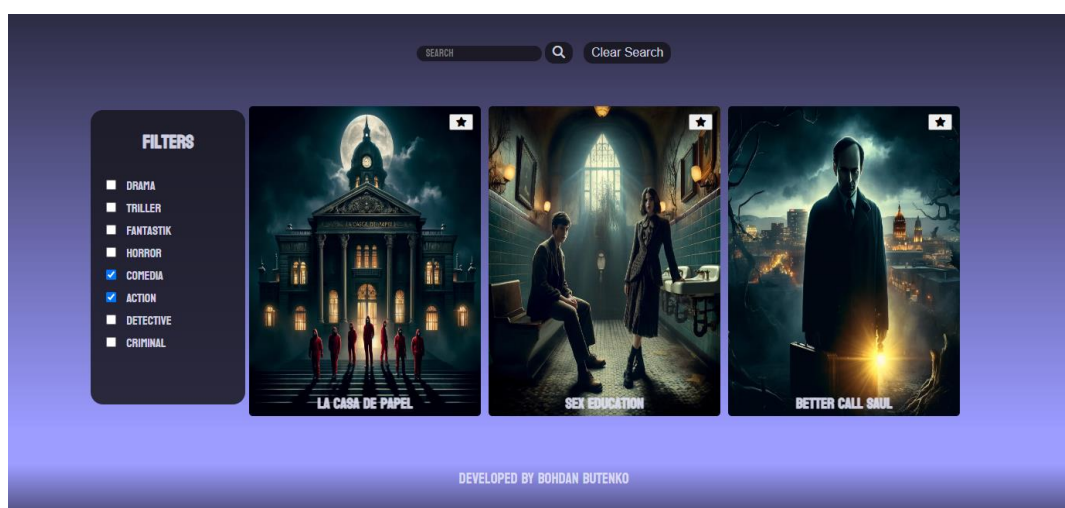


Рис. 2.19. Приклад роботи фільтрів з обраними жанрами

Також основна секція складається з зони пошуку. Зона дозволяє, дивлячись з локалізації сайту шукати відеоматеріал на сайті. Тобто, якщо сайт на українській мові, то пошук виконується на українській мові, якщо на англійській



– то мова англійська. Система активізує пошук при натисканні “Enter”, або натискаючи кнопку на сайті. Також реалізована кнопка, яка відмінляє результат пошуку та очищує текст у пошуковій строці.

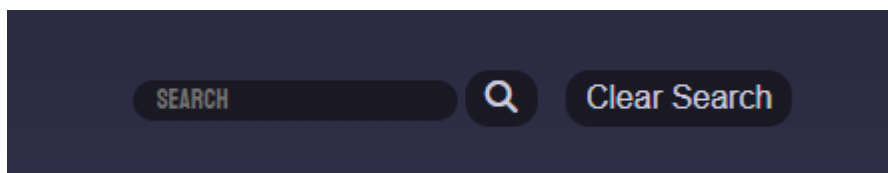


Рис. 2.20. Зона пошуку на картках на сайті

На сайті також реалізовано два стани карток. Відрізняються ці стани тим, як вони відображаються для авторизованого користувача та неавторизованого користувача. У випадку з неавторизованим користувачем, картка виглядає наступним чином. (Рис. 2.17)



Рис. 2.21. Картка неавторизованого користувача

У випадку з авторизованим користувачем, з’являється ще кнопка додавання в список улюблених відеоматеріалів. Для того, щоб користувач міг у особистому кабінеті відслідковувати лише ті картки, які вважає за потрібне

відслідковувати.



Рис. 2.22. Картка авторизованого користувача

Натиснувши на обрану картку, користувач може отримати більш детальну інформацію про відеоматеріал. У цьому стані сайту, користувач зможе побачити назву, рейтинг, сезон чи серію, на якому був зупинений перегляд та опис серіалу.

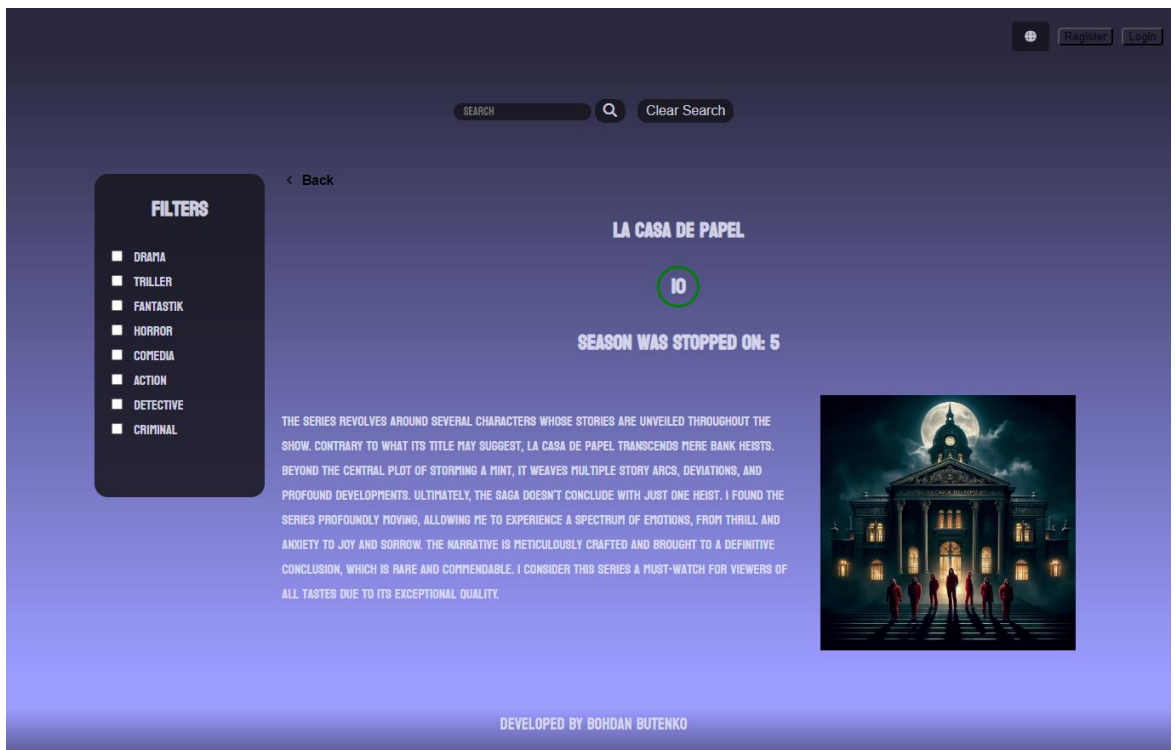


Рис. 2.23. Детальний опис обраного відеоматеріалу

## РОЗДІЛ 3

### ЕКОНОМІЧНИЙ РОЗДІЛ

#### 3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Вхідні дані:

- передбачуване число операторів програми – 1500;
- коефіцієнт складності програми – 1,4;
- коефіцієнт корекції програми в ході її розробки – 0,1;
- годинна заробітна плата програміста – 255,7 грн/год.

Середня годинна зарплата Junior Front-End розробника в Україні була отримана використовуючи дані з «Української спільноти програмістів (DOU)». Станом на квітень 2024 року зарплата Junior Full-Stack розробника становить приблизно 1200 доларів. При курсі валют НБУ на початок квітня 2024 року один американський долар дорівнює 37,50 грн, тому середня зарплата в гривнях дорівнює 45 000 грн. При стандартному графіку (176 годин/місяць) зарплата за годину буде становити близько 255,7 грн.

- коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,3;
- коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1;
- вартість машино-години ЕОМ – 25.57 грн/год.

Оскільки для цього проекту потрібна значна потужність ПК для роботи серверу та підтримки зберігання значної кількості даних на сервері баз даних, гарним рішенням буде оренда ноутбуку на час розробки додатку. Вартість оренди ноутбука місяць становить 4500 грн. При стандартному графіку (176 годин/місяць) вартість машино-години ЕОМ за годину роботи буде становити приблизно 25.57 грн. В цю вартість входить ремонт за гарантією та базовий комплект гарнітури (клавіатура та миша).

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d, \quad (3.1)$$

де  $t_o$  – витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

$t_u$  – витрати праці на дослідження алгоритму рішення задачі;

$t_a$  – витрати праці на розробку блок-схеми алгоритму;

$t_n$  – витрати праці на програмування по готовій блок-схемі;

$t_{oml}$  – витрати праці на налагодження програми на ЕОМ;

$t_d$  – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де  $q$  – передбачуване число операторів (1500);

$C$  – коефіцієнт складності програми (1,4);

$p$  – коефіцієнт корекції програми в ході її розробки (0,1).

Звідси за формулою (3.2) умовне число операторів в програмі:

$$Q = 1500 * 1,4 * (1 + 0,1) = 2310 \text{ людино-годин} \quad (3.3)$$

Витрати праці на вивчення опису задачі  $t_u$  визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \quad (3.4)$$

де  $B$  – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

$k$  – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 2 до 3 років він складає 1.

Прийmemo збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ( $B = 1,3$ ). З урахуванням коефіцієнта кваліфікації  $k = 1$ , отримуємо витрати праці на вивчення опису завдання за формулою (3.4):

$$t_u = (2310 * 1,3) / (85 * 1) = 35,3 \text{ людино-годин} \quad (3.5)$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \quad (3.6)$$

де  $Q$  – умовне число операторів програми;

$k$  – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.6), отримуємо:

$$t_a = 2310 / (25 \cdot 1) = 92,4 \text{ людино-годин} \quad (3.7)$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k} \quad (3.8)$$

За формулою (3.8) та значенням параметру  $Q$ , обчислюємо витрати праці на програмування по готовій блок-схемі:

$$t_n = 2310 / (25 \cdot 1) = 92,4 \text{ людино-годин} \quad (3.9)$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4 \dots 5) \cdot k} \quad (3.10)$$

Витрати праці на налагодження програми на ЕОМ за умови автономного налагодження одного завдання за формулою (3.10):

$$t_{oml} = 2310 / (5 \cdot 1) = 462 \text{ людино-годин} \quad (3.11)$$

- за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml} \quad (3.12)$$

Витрати праці на налагодження програми на ЕОМ за умови комплексного налагодження завдання за формулою (3.12):

$$t_{oml}^k = 1,5 \cdot 462 = 693 \text{ людино-годин} \quad (3.13)$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \quad (3.14)$$

де  $t_{\partial p}$  – трудомісткість підготовки матеріалів і рукопису:

$$t_{\partial p} = \frac{Q}{(15..20) \cdot k} \quad (3.15)$$

$t_{\partial o}$  – трудомісткість редагування, печатки й оформлення документації:

$$t_{\partial o} = 0,75 \cdot t_{\partial p} \quad (3.16)$$

За формулами (3.15), (3.16) та (3.14) обчислюємо витрати праці на документацію:

$$t_{\partial p} = 2310 / (20 \cdot 1) = 115,5 \text{ людино-годин}, \quad (3.17)$$

$$t_{\partial o} = 0,75 \cdot 115,5 = 86,6 \text{ людино-годин}, \quad (3.18)$$

$$t_{\partial} = 115,5 + 86,6 = 202,1 \text{ людино-годин}. \quad (3.19)$$

Повертаючись до формули (3.1), отримуємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 35,3 + 92,4 + 92,4 + 462 + 202,1 = 934,2 \text{ людино-годин}. \quad (3.20)$$

### 3.2. Витрати на створення програмного забезпечення

Витрати на створення ПЗ  $K_{ПО}$  включають витрати на заробітну плату виконавця програми  $Z_{ЗП}$  і витрат машинного часу, необхідного на налагодження

програми на ЕОМ:

$$K_{\text{ПО}} = Z_{\text{ЗП}} + Z_{\text{МВ}} \quad (3.21)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{\text{ЗП}} = t \cdot C_{\text{ПР}}, \quad (3.22)$$

де:  $t$  – загальна трудомісткість, людино-годин;

$C_{\text{ПР}}$  – середня годинна заробітна плата програміста, грн/година.

З урахуванням того, що середня годинна зарплата програміста становить 145,4 грн / год, за формулою (3.22) отримуємо:

$$Z_{\text{ЗП}} = 934,2 \cdot 255,7 \approx 238\,875 \text{ грн}, \quad (3.23)$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{\text{МВ}} = t_{\text{отл}} \cdot C_{\text{МЧ}}, \quad (3.24)$$

де:  $t_{\text{отл}}$  – трудомісткість налагодження програми на ЕОМ, год;

$C_{\text{МЧ}}$  – вартість машино-години ЕОМ, грн/год.

За формулою (3.24) вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{\text{МВ}} = 462 \cdot 25,57 \approx 11\,813 \text{ грн} \quad (3.25)$$

Звідси, за формулою (3.21), витрати на створення програмного продукту:

$$K_{\text{ПО}} = 238\,875 + 11\,813 = 250\,688 \text{ грн} \quad (3.26)$$

Очікуваний період створення програмного застосунку:

$$T = \frac{t}{B_k \cdot F_p}, \text{ мес.}, \quad (3.27)$$

де:  $B_k$  – число виконавців (дорівнює 1);

$F_p$  – місячний фонд робочого часу (при 40 годинному робочому тижні  $F_p = 176$  годин).

Звідси, за формулою (2.27), витрати на створення програмного продукту:



$$T = 934,2 / (1 \cdot 176) \approx 5,3 \text{ місяців} \quad (3.28)$$

**Висновки:** програмне забезпечення розроблено для забезпечення доступу користувачів до бібліотеки фільмів та серіалів, ефективної взаємодії між користувачем та сайтом. Вартість даного програмного забезпечення становить 250 688 грн і не вимагає додаткових витрат як при розробці програми. Очікуваний час розробки становить 934,2 годин, тобто 5,3 місяці. Цей термін пов'язаний з наявним числом операторів та незначним коефіцієнтом кваліфікації програміста, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

## ВИСНОВКИ

Метою даного кваліфікаційної роботи було створення веб-орієнтованої інформаційної системи з використанням сучасних технологій HTML, CSS, та JavaScript для управління та відображення контенту фільмів і серіалів, яка би підвищила ефективність доступу користувачів до відеоконтенту.

Актуальність теми пов'язана з розширенням ринку фільмів, серіалів та зростаючою потребою у зручних та функціональних веб-сайтах для перегляду та управління інформацією про відеоматеріали. В умовах сучасного світу, коли багато користувачів віддають перевагу онлайн-платформам для перегляду фільмів і серіалів, розробка такого сайту є надзвичайно актуальною.

У даному веб-сайті було реалізовано функціонал для реєстрації та авторизації користувачів, локалізація інтерфейсу на дві мови (українська та англійська), фільтрація та пошук контенту, додавання до списку обраного та перегляд детальної інформації про фільми та серіали. Такий функціонал забезпечує зручний та ефективний доступ до контенту.

Практичне призначення даної системи полягає у розробці веб-орієнтованого додатку, що сприяє ефективному функціонуванню платформи для перегляду контенту шляхом поліпшення взаємодії з користувачами та розширення аудиторії завдяки можливості фільтрації та пошуку контенту. Додаток забезпечує процеси реєстрації, авторизації, управління обраними фільмами та серіалами, а також надає детальну інформацію про них. Крім того, розроблена система надає можливість зручно знаходити потрібні фільми та серіали, надавати повну інформацію про них та управляти обраним контентом. В результаті створення такої системи користувачі зможуть покращити свій досвід перегляду контенту фільмів та ефективно використовувати можливості платформи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. HTML: Hypertext Markup Language, MDN Web Docs. Доступно за: <https://developer.mozilla.org/en-US/docs/Web/HTML> (01.05.2024).
2. CSS: Cascading Style Sheets, MDN Web Docs. Доступно за: <https://developer.mozilla.org/en-US/docs/Web/CSS> (01.05.2024).
3. JavaScript Guide, MDN Web Docs. Доступно за: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (01.05.2024).
4. Haverbeke, Marijn. Eloquent JavaScript: A Modern Introduction to Programming. No Starch Press, 2018.
5. The Modern JavaScript Tutorial, JavaScript.info. Доступно за: <https://javascript.info/> (01.05.2024).
6. freeCodeCamp - HTML, CSS, and JavaScript Tutorials, GitHub - freeCodeCamp. Доступно за: <https://github.com/freeCodeCamp> (01.05.2024).
7. Wes Bos. JavaScript30 - Build 30 Things with Vanilla JS, GitHub - JavaScript30. Доступно за: <https://github.com/wesbos/JavaScript30> (01.05.2024).
8. JavaScript Tutorials for Beginners, The Net Ninja, YouTube. Доступно за: <https://www.youtube.com/c/TheNetNinja> (01.05.2024).
9. Liguori, Eve. JavaScript Essential Training, LinkedIn Learning. Доступно за: <https://www.linkedin.com/learning/javascript-essential-training> (01.05.2024).
10. Flanagan, David. JavaScript: The Definitive Guide. O'Reilly Media, 2020.
11. Simpson, Kyle. You Don't Know JS. O'Reilly Media, 2014.
12. Duckett, Jon. HTML and CSS: Design and Build Websites. John Wiley & Sons, 2011.
13. Meyer, Eric A. and Estelle Weyl. CSS: The Definitive Guide. O'Reilly Media, 2017.
14. A Complete Guide to Grid, CSS Tricks. Доступно за: <https://css-tricks.com/snippets/css/complete-guide-grid/> (01.05.2024).
15. A Complete Guide to Flexbox, CSS Tricks. Доступно за: <https://css-tricks.com/snippets/css/complete-guide-flexbox/> (01.05.2024).

tricks.com/snippets/css/a-guide-to-flexbox/ (01.05.2024).

16. Pilgrim, Mark. Dive Into HTML5. Доступно за: <http://diveintohtml5.info/> (01.05.2024).

17. Understanding CSS Grid: Creating A Grid Container, Smashing Magazine. Доступно за: <https://www.smashingmagazine.com/understanding-css-grid-creating-a-grid-container/> (01.05.2024).

18. Building a Simple CRUD Application with JavaScript, freeCodeCamp. Доступно за: <https://www.freecodecamp.org/news/javascript-crud-application-tutorial/> (01.05.2024).

19. JavaScript Fundamentals: Your Cheatsheet for the Basics, Codecademy. Доступно за: <https://www.codecademy.com/articles/javascript-fundamentals> (01.05.2024).

20. Can I use. Доступно за: <https://caniuse.com/> (01.05.2024).

21. Васильєва, Т. А. Економічний аналіз: підручник. — Київ: Центр навчальної літератури, 2020. — 376 с.

22. Грищенко, І. М. Фінансовий менеджмент: підручник. — Харків: Вид-во ХНЕУ, 2021. — 416 с.

23. Єрмошенко, М. М., Шапран, В. В. Інвестиційний аналіз: навч. посіб. — Київ: КНЕУ, 2019. — 288 с.

24. Коваленко, Д. В. Маркетинг: підручник. — Київ: Ліра-К, 2020. — 472 с.

25. Кредісов, А. І., Семенов, Г. А. Економічна ефективність інвестицій: навч. посіб. — Київ: КНЕУ, 2018. — 296 с.

26. Ларіонова, К. Л., Павленко, Ю. В. Стратегічний менеджмент: підручник. — Київ: Академвидав, 2020. — 320 с.

27. Мельник, М. І., Савчук, В. С. Економічна оцінка інвестиційних проектів: підручник. — Київ: Ліра-К, 2019. — 352 с.

28. Романенко, О. Р. Економіка підприємства: підручник. — Київ: Центр навчальної літератури, 2020. — 400 с.

29. Савлук, М. І., Мороз, А. М., Панченко, В. М. Фінанси підприємств: підручник. — Київ: КНЕУ, 2021. — 480 с.

30. Шевчук, В. О. Управління фінансами: підручник. — Львів: Вид-во ЛНУ, 2020. — 368 с.

## КОД ПРОГРАМИ

## Лістинг файлу index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Movies</title>
  <link rel="dns-prefetch" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Koulen&display=swap"
rel="stylesheet">
  <script
src="https://kit.fontawesome.com/68a34cfcac.js"
crossorigin="anonymous" async></script>
  <link rel="stylesheet" href="style.css">
  <link rel="icon" type='image/x-icon' href="img/Logo.png">
</head>

<body>
  <div class="container">
    <div class="intro">
      <h1>Welcome to movies library!</h1>
      <div class="top-bar">
        <div class="language-switcher">
          <button class="lang-toggle" aria-label="Toggle language
options">
            <i class="fa-solid fa-globe"></i>
          </button>
          <div class="lang-options hidden">
            <button id="lang-en" aria-label="Switch to
English">EN</button>
            <button id="lang-ua" aria-label="Switch to
Ukrainian">UA</button>
          </div>
        </div>
      </div>

      <div class="auth-buttons">
        <button id="register-button" aria-label="Register new
account">Register</button>
        <button id="login-button" aria-label="Log in to
account">Login</button>
        <button id="profile-button" style="display: none;"
aria-label="Profile">
          <i class="fa-solid fa-user"></i>
        </button>
      </div>
    </div>
  </div>

```

```

        <div class="subintro">
            <figure class="icon-cards">
                <div class="icon-cards__content"></div>
            </figure>
        </div>
    </div>

    <main class="main">
        <section class="search">
            <input type="text" id="text-to-find" placeholder="search"
            aria-label="Search field">
            <button type="submit" id="search-button" aria-
            label="Search">
                <i class="fa-solid fa-magnifying-glass"></i>
            </button>
            <button type="button" id="clear-search" class="clear-
            button" aria-label="Clear search">
                <i class="fa-solid fa-xmark"></i> Clear Search
            </button>
        </section>
        <section class="gallery">
            <div class="searchNav">
                <h1>Filters</h1>
            </div>
            <div class="cards">
            </div>
        </section>
    </main>
    <footer>
        <p>Developed by Bohdan Butenko</p>
    </footer>
</div>
<script type="module" src="script.js"></script>
</body>

</html>

```

## Лістинг файлу style.css

```

@charset "UTF-8";

:root {
    --primary-color: rgba(22, 21, 29, 0.85);
    --secondary-color: rgba(155, 155, 255, 0.85);
    --accent-color: rgba(128, 159, 247, 0.7);
    --text-color: rgba(225, 225, 245, 0.9);
}

html, body {
    background-color: var(--secondary-color);
    font-size: 16px;
    padding: 0;
    margin: 0;
    border: 0;
    font-family: "Koulen", cursive;

```

```

    box-sizing: border-box;
}

*, *::before, *::after {
    box-sizing: inherit;
}

::-webkit-scrollbar {
    width: 7px;
    height: 4px;
}

::-webkit-scrollbar-button,
::-webkit-scrollbar-track {
    background-color: var(--primary-color);
}

::-webkit-scrollbar-thumb {
    border-radius: 2px;
    background-color: var(--secondary-color);
}

.intro {
    height: 100vh;
    background-color: var(--primary-color);
    color: var(--text-color);
}

.intro h1 {
    margin-top: 0;
    font-size: 3rem;
    text-align: center;
}

.intro p {
    font-size: 2rem;
    text-align: center;
}

.subintro {
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
}

.icon-cards {
    position: relative;
    width: 20vw;
    height: 10vw;
    max-width: 300px;
    max-height: 400px;
    margin-top: 100px;
    color: var(--text-color);
    perspective: 1000px;
    transform-origin: center;
}

.icon-cards h1 {
    font-size: 1.2rem;
}

```



```

    width: 100%;
}
.icon-cards__content {
    position: absolute;
    width: 300px;
    height: 400px;
    transform-origin: center;
    transform-style: preserve-3d;
    transform: translateZ(-30vw) rotateY(0);
    animation: carousel 10s infinite cubic-bezier(0.77, 0, 0.175, 1) forwards;
}
.icon-cards__content.step-animation {
    animation-duration: 8s;
}
.icon-cards__item {
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    width: 300px;
    height: 400px;
    transform-origin: center;
}
.icon-cards__item h1 {
    position: absolute;
    top: 90%;
    right: 50%;
    transform: translate(50%, -50%);
}
.icon-cards__item img {
    width: 100%;
    height: 100%;
}
.icon-cards__item:nth-child(1) {
    transform: rotateY(0) translateZ(30vw);
}
.icon-cards__item:nth-child(2) {
    transform: rotateY(120deg) translateZ(30vw);
}
.icon-cards__item:nth-child(3) {
    transform: rotateY(240deg) translateZ(30vw);
}

@keyframes carousel {
    0%, 100% {
        transform: translateZ(-35vw) rotateY(0deg);
    }
    25% {
        transform: translateZ(-35vw) rotateY(-120deg);
    }
    50% {
        transform: translateZ(-35vw) rotateY(-240deg);
    }
    75% {
        transform: translateZ(-35vw) rotateY(-360deg);
    }
}

```

```

}

.main {
  background: linear-gradient(var(--primary-color), var(--secondary-color));
  padding-top: 50px;
}

.gallery {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  margin-top: 50px;
}

.cards {
  display: flex;
  flex-wrap: wrap;
  max-width: 980px;
  width: 70%;
}

.card {
  position: relative;
  cursor: pointer;
  width: auto;
  overflow: hidden;
  margin: 5px;
  border-radius: 5px;
  color: var(--text-color);
}

.card h1 {
  position: absolute;
  top: 90%;
  left: 50%;
  transform: translate(-50%, -50%);
  font-size: 1.2rem;
  width: 100%;
  text-align: center;
  overflow: hidden;
  white-space: nowrap;
  text-overflow: ellipsis;
}

.card h1, .card h2, .card p {
  pointer-events: none;
}

.card img {
  width: 300px;
  height: 400px;
  border-radius: 5px;
  transition: opacity 0.2s ease-out;
}

.card img:hover {

```

```

    opacity: 0.5;
}

.search {
  text-align: center;
}

.search input {
  border-radius: 12px;
  height: 20px;
  background-color: var(--primary-color);
  color: var(--text-color);
  border: none;
  padding-left: 12px;
  font-family: "Koulen", cursive;
}

.search input:focus {
  outline: 3px ridge var(--text-color);
  border-radius: 2rem;
}

.search button {
  cursor: pointer;
  border-radius: 12px;
  background-color: var(--primary-color);
  color: var(--text-color);
  font-size: 1rem;
  padding: 5px 10px;
  border: none;
  transition: 0.5s;
}

.search button:hover {
  box-shadow: inset 0 3rem 0 -1px var(--accent-color);
}

.searchNav {
  background-color: var(--primary-color);
  color: var(--text-color);
  border-radius: 16px;
  margin-top: 10px;
  display: flex;
  width: 200px;
  height: 380px;
  flex-direction: column;
}

.searchNav h1 {
  text-align: center;
}

.boxtype input {
  display: inline;
  margin-left: 20px;
  padding-left: 10px;
  cursor: pointer;
}

.boxtype label {
  display: inline;
  margin-left: 10px;
}

```

```

    transform: scale(0.3);
  }

  footer {
    background: linear-gradient(var(--secondary-color), var(--primary-color));
    height: 10vh;
    text-align: center;
    margin-top: 50px;
  }

  footer p {
    font-size: 1.2rem;
    color: var(--text-color);
  }

  .back, .clear-button {
    cursor: pointer;
    border: none;
    border-radius: 10px;
  }

  .back {
    margin: 0 10px;
    max-height: 35px;
    padding: 5px;
    display: flex;
    align-items: center;
    justify-content: center;
    background-color: transparent;
  }

  .back i {
    margin: 0 10px;
  }

  .back:hover {
    background-color: var(--accent-color);
    transition: 1s;
  }

  .cardInfo h2, .wrapperforimg p, .specialH1 {
    color: var(--text-color);
  }

  .cardInfo h2, .specialH1 {
    text-align: center;
  }

  .wrapperforimg {
    display: flex;
    padding: 20px;
  }

  .wrapperforimg img {
    max-width: 300px;
    max-height: 400px;
  }

```

```

    margin-left: 20px;
}

.rating {
    border-radius: 50%;
    max-width: 50px;
    margin: 0 auto;
}

.green, .orange, .red {
    border-width: 3px;
    border-style: solid;
}

.green {
    border-color: green;
}

.orange {
    border-color: orange;
}

.red {
    border-color: red;
}

.clear-button {
    background-color: #f2f2f2;
    padding: 10px 20px;
    margin-left: 10px;
    font-size: 16px;
}

.clear-button:hover {
    background-color: #e1e1e1;
}

.top-bar {
    position: fixed;
    top: 20px;
    right: 20px;
    display: flex;
    align-items: center;
    padding: 5px;
    gap: 10px;
    z-index: 1000;
}

.language-switcher, .auth-buttons {
    display: flex;
    gap: 10px;
}

.language-switcher {
    position: relative;
}

```

```

.lang-toggle, .lang-options button {
  cursor: pointer;
  background-color: var(--primary-color);
  color: var(--text-color);
  padding: 10px 15px;
  border-radius: 5px;
  border: none;
}

.lang-options {
  display: none;
  position: absolute;
  right: 0;
  top: 35px;
  border-radius: 5px;
  width: 102%;
}

.lang-toggle:hover + .lang-options, .lang-options:hover {
  display: block;
}

.lang-toggle:hover, .lang-options button:hover {
  background-color: var(--secondary-color);
}

.auth-buttons button {
  background-color: var(--primary_color);
  border-radius: 4px;
  cursor: pointer;
}

.auth-buttons button:hover {
  background-color: var(--secondary-color);
}

.favorite-button {
  position: absolute;
  top: 10px;
  right: 10px;
  cursor: grab;
  transition: background-color 0.3s ease, transform 0.3s ease;
  display: none;
}

.favorite-button:hover {
  background-color: var(--accent-color);
}

.active {
  color: white;
  border: 2px solid var(--primary-color);
  box-shadow: inset 0 4px 6px rgba(0, 0, 0, 0.2);
}

.unfavorite-button {
  position: absolute;
}

```

```

top: 15px;
right: 10px;
background-color: transparent;
border: none;
cursor: pointer;
color: red;
}

.unfavorite-button i {
  font-size: 18px;
}

```

### Лістинг файлу script.js

```

import './modules/LanguageSwitcher.js';
import './modules/loginModal.js';
import './modules/generateCards.js';
import './modules/generateInputs.js';
import './modules/SearchAndFilters.js';
import './modules/registerModal.js';

```

### Лістинг файлу SearchAndFilters.js

```

import { renderData, showingFavorites, showFavoriteCards } from
'./generateCards.js';
import { filterData } from './generateInputs.js';
import { data as defaultData } from '../data/data.js';

const searchButton = document.querySelector('#search-button');
const searchText = document.getElementById('text-to-find');
const clearSearchButton = document.getElementById('clear-search');

searchButton.addEventListener('click', Find);

searchText.addEventListener('keydown', (event) => {
  if (event.key === 'Enter') Find();
});

clearSearchButton.addEventListener('click', () => {
  searchText.value = '';
  const checkboxes = document.querySelectorAll('.searchNav
input[type="checkbox"]');
  checkboxes.forEach(checkbox => {
    checkbox.checked = false;
    sessionStorage.setItem(checkbox.id, 'false');
  });
  const cards = document.querySelectorAll('.card');
  cards.forEach(card => card.style.display = '');

  if (showingFavorites) {
    const [currentFilters] = filterData();
    showFavoriteCards(currentFilters);
  } else {
    renderData(defaultData);
  }
});

```

```

    }
  });

function Find() {
  const searchQuery = searchText.value.trim().toLowerCase();
  const cards = document.querySelectorAll('.card');

  cards.forEach(card => {
    const content = card.textContent || card.innerText;
    card.style.display = content.toLowerCase().includes(searchQuery) ? '' :
'none';
  });
}

```

## Лістинг файлу registerModal.js

```

import { users } from '../data/users.js';
import { currentLanguage, updateTexts } from './LanguageSwitcher.js';

document.getElementById('register-button').addEventListener('click',
function() {
  if (document.getElementById('registration-modal') ||
document.getElementById('login-modal')) {
    return;
  }

  const modal = document.createElement('div');
  modal.id = 'registration-modal';
  modal.style.position = 'fixed';
  modal.style.left = '0';
  modal.style.top = '0';
  modal.style.width = '100%';
  modal.style.height = '100%';
  modal.style.backgroundColor = 'rgba(0, 0, 0, 0.5)';
  modal.style.display = 'flex';
  modal.style.justifyContent = 'center';
  modal.style.alignItems = 'center';
  modal.innerHTML = `
    <div style="background: white; padding: 20px; border-radius: 5px;">
      <h2>Register</h2>
      <form id="registration-form">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required
minlength="4"><br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password"
required minlength="6" pattern="(?!.*[0-9]) (?!.*[a-zA-Z]).*">
        <label for="password" style="font-size: 0.8em; color: grey;
margin-left: 10px;">*The password must be at least 6 characters long and
contain at least one letter.</label><br><br>
        <button type="submit">Register</button>
        <button type="button"
onclick="document.body.removeChild(document.getElementById('registration-
modal'))">Back</button>
      </form>
    </div>
  `;

```



```

document.body.appendChild(modal);

updateTexts(currentLanguage);

document.getElementById('registration-
form').addEventListener('submit', function(event) {
    event.preventDefault();

    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    const user = {
        username: username,
        password: password,
        favorites: [],
    };

    users.push(user);

    document.body.removeChild(modal);
});
});

```

### Лістинг файлу loginModal.js

```

import { users } from '../data/users.js';
import { currentLanguage, updateTexts } from './LanguageSwitcher.js';

export let loggedInUser = null;

document.getElementById('login-button').addEventListener('click',
function() {
    if (document.getElementById('login-modal') ||
document.getElementById('registration-modal')) {
        return;
    }

    const loginModal = document.createElement('div');
    loginModal.id = 'login-modal';
    loginModal.style.position = 'fixed';
    loginModal.style.left = '0';
    loginModal.style.top = '0';
    loginModal.style.width = '100%';
    loginModal.style.height = '100%';
    loginModal.style.backgroundColor = 'rgba(0, 0, 0, 0.5)';
    loginModal.style.display = 'flex';
    loginModal.style.justifyContent = 'center';
    loginModal.style.alignItems = 'center';
    loginModal.innerHTML = `
        <div style="background: white; padding: 20px; border-radius: 5px;">
            <h2>Login</h2>
            <form id="login-form">
                <label for="login-username">Username:</label>
                <input type="text" id="login-username" name="username"
required><br><br>
                <label for="login-password">Password:</label>

```

```

        <input type="password" id="login-password" name="password"
required><br><br>
        <button type="submit">Login</button>
        <button
                                type="button"
onclick="document.body.removeChild(document.getElementById('login-
modal'))">Back</button>
        </form>
    </div>
`;
    document.body.appendChild(loginModal);

    updateTexts(currentLanguage);

    document.getElementById('login-form').addEventListener('submit',
function(event) {
    event.preventDefault();

    const username = document.getElementById('login-username').value;
    const password = document.getElementById('login-password').value;

    const user = users.find(user => user.username === username &&
user.password === password);

    if (user) {
        loggedInUser = user;
        document.getElementById('register-button').style.display =
'none';
        document.getElementById('login-button').style.display =
'none';
        document.getElementById('profile-button').style.display =
'inline-block';

        const favoriteButtons = document.querySelectorAll('.favorite-
button');
        favoriteButtons.forEach(button => {
            button.style.display = 'inline-block';
            button.addEventListener('click', function() {
                const movieId = button.getAttribute('data-movie-id');
                if (!user.favorites.includes(movieId)) {
                    user.favorites.push(movieId);
                    console.log('Added to favorites:', user.favorites);
                }
            });
        });

        document.body.removeChild(loginModal);
    } else {
        loggedInUser = null;
        alert('Invalid username or password!');
    }
});
});

```

### Лістинг файлу LanguageSwitcher.js

```

import { data } from '../data/data.js';
import { dataInput } from '../data/dataInput.js';

```

```

import { data as defaultData } from '../data/data.js';

export const texts = {
  en: {
    welcome: "Welcome to movies library!",
    searchPlaceholder: "search",
    clearSearch: "Clear Search",
    filters: "Filters",
    developedBy: "Developed by Bohdan Butenko",
    seasonStopped: "Season was stopped on: ",
    register: "Register",
    login: "Login",
    username: "Username:",
    password: "Password:",
    submitRegister: "Register",
    submitLogin: "Login",
    back: "Back",
    passwordRequirements: "*The password must be at least 6 characters long
and contain at least one letter."
  },
  ua: {
    welcome: "Ласкаво просимо до бібліотеки фільмів!",
    searchPlaceholder: "пошук",
    clearSearch: "Очистити пошук",
    filters: "Фільтри",
    developedBy: "Розроблено Богданом Бутенко",
    seasonStopped: "Сезон, на якому було зупинено перегляд: ",
    register: "Реєстрація",
    login: "Вхід",
    username: "Ім'я користувача:",
    password: "Пароль:",
    submitRegister: "Зареєструватися",
    submitLogin: "Увійти",
    back: "Назад",
    passwordRequirements: "*Пароль має бути не менше 6 символів та містити
хоча б одну літеру."
  }
};

export function updateTexts(language) {
  document.getElementById('register-button').textContent =
texts[language].register;
  document.getElementById('login-button').textContent =
texts[language].login;

  const registrationModal = document.getElementById('registration-modal');
  if (registrationModal) {
    registrationModal.querySelector('h2').textContent =
texts[language].register;
    registrationModal.querySelector('label[for="username"]').textContent
= texts[language].username;
    registrationModal.querySelector('label[for="password"]').textContent
= texts[language].password;
    registrationModal.querySelector('button[type="submit"]').textContent
= texts[language].submitRegister;

    registrationModal.querySelectorAll('label[for="password"]')[1].textContent
= texts[language].passwordRequirements;
  }
}

```

```

        registrationModal.querySelector('button[type="button"]').textContent
= texts[language].back;
    }

    const loginModal = document.getElementById('login-modal');
    if (loginModal) {
        loginModal.querySelector('h2').textContent = texts[language].login;
        loginModal.querySelector('label[for="login-username"]').textContent
= texts[language].username;
        loginModal.querySelector('label[for="login-password"]').textContent
= texts[language].password;
        loginModal.querySelector('button[type="submit"]').textContent
= texts[language].submitLogin;
        loginModal.querySelector('button[type="button"]').textContent
= texts[language].back;
    }
}

document.getElementById('lang-en').addEventListener('click',      ()    =>
updateTexts('en'));
document.getElementById('lang-ua').addEventListener('click',      ()    =>
updateTexts('ua'));

export let currentLanguage = localStorage.getItem('preferredLanguage') ||
'en';

export function switchLanguage(lang) {
    currentLanguage = lang;
    localStorage.setItem('preferredLanguage', lang);

    document.querySelector('.intro h1').textContent = texts[lang].welcome;
    document.getElementById('text-to-find').setAttribute('placeholder',
texts[lang].searchPlaceholder);
    document.getElementById('clear-search').textContent
= texts[lang].clearSearch;
    document.querySelector('.searchNav h1').textContent
= texts[lang].filters;
    document.querySelector('footer p').textContent = texts[lang].developedBy;

    document.querySelectorAll('.back h3').forEach(button => {
        button.textContent = texts[lang].back;
    });

    updateFilterTexts(lang);
    updateCardTexts(lang);
    updateCarouselTexts(lang);
}

function updateFilterTexts(language) {
    dataInput.forEach(filter => {
        const filterElement = document.getElementById(filter.id);
        if (filterElement) {
            filterElement.nextElementSibling.textContent = language === 'ua' ?
filter.labelUA : filter.label;
        }
    });
}
}

```

```

function updateCardTexts(language) {
  const allCards = document.querySelectorAll('.card, .cardInfo');
  allCards.forEach(card => {
    const cardData = data.find(d => d.id === card.id);
    if (cardData) {
      const titleElement = card.querySelector('[data-title]') ||
card.querySelector('.specialH1');
      const descriptionElement = card.querySelector('[data-description]')
|| card.querySelector('p:not(.rating):not(.season)');
      const seasonElement = card.querySelector('.season');
      if (titleElement) titleElement.textContent = language === 'ua' ?
cardData.titleUA : cardData.title;
      if (descriptionElement) descriptionElement.textContent = language ===
'ua' ? cardData.descriptionUA : cardData.description;
      if (seasonElement) seasonElement.innerHTML =
texts[language].seasonStopped + cardData.season;
    }
  });
}

function updateCarouselTexts(language) {
  const carouselItems = document.querySelectorAll('.icon-cards__item [data-
title]');
  carouselItems.forEach(item => {
    const cardData = defaultData.find(d => d.id === item.getAttribute('data-
title'));
    if (cardData) {
      item.textContent = language === 'ua' ? cardData.titleUA :
cardData.title;
    }
  });
}

document.getElementById('lang-en').addEventListener('click', () =>
switchLanguage('en'));
document.getElementById('lang-ua').addEventListener('click', () =>
switchLanguage('ua'));

document.addEventListener('DOMContentLoaded', () => {
  switchLanguage(localStorage.getItem('preferredLanguage') || 'en');
});

```

## Лістинг файлу generateInputs.js

```

import { data } from '../data/data.js';
import { dataInput } from '../data/dataInput.js';
import { renderData } from './generateCards.js';
import { showingFavorites, showFavoriteCards } from './generateCards.js';

const searchNav = document.querySelector('.searchNav');

dataInput.forEach((el) => {
  const inputBlock = document.createElement('div');
  inputBlock.classList.add('boxtype');

  const createLabel = document.createElement('label');
  createLabel.innerHTML = el.label;

```

```

    const createInput = document.createElement('input');
    createInput.type = 'checkbox';
    createInput.id = el.id;

    inputBlock.append(createInput, createLabel);
    searchNav.appendChild(inputBlock);
  });

  const inputs = Array.from(document.querySelectorAll('.searchNav input'));

  inputs.forEach((input) => {
    input.onChange = (event) => {
      sessionStorage.setItem(input.id, event.target.checked.toString());
      const [filters, filteredData] = filterData();
      if (showingFavorites) {
        showFavoriteCards(filters);
      } else {
        renderData(filters.length ? filteredData : data);
      }
    };

    if (sessionStorage.getItem(input.id) === 'true') {
      input.checked = true;
    }
  });

  const [filters, filteredData] = filterData();
  renderData(filters.length ? filteredData : data);

  export function filterData() {
    const filters = inputs.filter(input => input.checked).map(input =>
input.id);
    const filteredData = data.filter(film => filters.some(filter =>
film.genres.includes(filter)));
    return [filters, filteredData];
  }
}

```

## Лістинг файлу generateCards.js

```

import { data as defaultData } from '../data/data.js';
import { showCards } from './generateButtonAction.js';
import { switchLanguage, currentLanguage, texts } from
'./LanguageSwitcher.js';
import { loggedInUser } from './loginModal.js';
import { filterData } from './generateInputs.js';

export let showingFavorites = false;

export function showFavoriteCards(filters = []) {
  let cards = document.querySelector('.cards');
  cards.innerHTML = '';
  const filteredFavorites = loggedInUser.favorites.filter(favId => {
    const el = defaultData.find(item => item.id === favId);
    return filters.length ? filters.some(filter =>
el.genres.includes(filter)) : true;
  });
}

```

```

});

filteredFavorites.forEach(favId => {
  const el = defaultData.find(item => item.id === favId);
  if (el) {
    let card = document.createElement('div');
    card.classList.add('card');
    card.id = el.id;

    let createImg = document.createElement('img');
    createImg.src = el.imgSrc;

    let createH1 = document.createElement('h1');
    createH1.innerText = currentLanguage === 'ua' ? el.titleUA : el.title;
    createH1.setAttribute('data-title', el.id);

    let unfavoriteButton = document.createElement('button');
    unfavoriteButton.classList.add('unfavorite-button');
    unfavoriteButton.innerHTML = '<i class="fa fa-trash" aria-
hidden="true"></i>';
    unfavoriteButton.setAttribute('data-movie-id', el.id);
    unfavoriteButton.onclick = function(event) {
      event.stopPropagation();
      loggedInUser.favorites = loggedInUser.favorites.filter(id => id !==
el.id);
      console.log('Removed from favorites:', loggedInUser.favorites);
      card.remove();
    };

    card.onclick = function () {
      cards.innerHTML = '';

      let cardDetail = document.createElement('div');
      cardDetail.classList.add('cardInfo');
      cardDetail.id = el.id;

      let createImgDetail = document.createElement('img');
      createImgDetail.src = el.imgSrc;

      let createH1Detail = document.createElement('h1');
      createH1Detail.innerText = currentLanguage === 'ua' ? el.titleUA :
el.title;
      createH1Detail.classList.add('specialH1');

      let createPDetail = document.createElement('p');
      createPDetail.innerText = currentLanguage === 'ua' ?
el.descriptionUA : el.description;

      let rating = document.createElement('h2');
      rating.innerText = el.rating;
      rating.classList.add('rating');
      if (el.rating > 8) rating.classList.add('green');
      else if (el.rating > 5) rating.classList.add('orange');
      else rating.classList.add('red');

      let season = document.createElement('h2');
      season.classList.add('season');

```

```

        season.innerText = texts[currentLanguage].seasonStopped +
el.season;

    let textWithImg = document.createElement('div');
    textWithImg.classList.add('wrapperforimg');
    textWithImg.appendChild(createPDetail);
    textWithImg.appendChild(createImgDetail);

    let backButton = document.createElement('button');
    backButton.classList.add('back');
    backButton.onclick = () => {
        const [currentFilters] = filterData();
        showFavoriteCards(currentFilters);
        switchLanguage(currentLanguage);
    };

    let fontBack = document.createElement('i');
    fontBack.classList.add('fa-angle-left', 'fa-solid');

    let text = document.createElement('h3');
    text.innerText = 'Back';

    backButton.append(fontBack, text);

    cardDetail.append(createH1Detail, rating, season, textWithImg);

    cards.append(backButton, cardDetail);

    switchLanguage(currentLanguage);
};

card.append(createImg, createH1, unfavoriteButton);
cards.append(card);
}
});
}

document.getElementById('profile-button').addEventListener('click',
function() {
    const profileButton = document.getElementById('profile-button');
    const [currentFilters] = filterData();
    if (showingFavorites) {
        showCards(defaultData);
        showingFavorites = false;
        profileButton.classList.remove('active');
    } else {
        showFavoriteCards(currentFilters);
        showingFavorites = true;
        profileButton.classList.add('active');
    }
});

export function renderData(data) {
    let cards = document.querySelector('.cards');
    cards.innerHTML = '';
    data.forEach((el) => {
        let card = document.createElement('div');

```



```

card.classList.add('card');
card.id = el.id;

let createImg = document.createElement('img');
createImg.src = el.imgSrc;

let createH1 = document.createElement('h1');
createH1.innerText = currentLanguage === 'ua' ? el.titleUA : el.title;
createH1.setAttribute('data-title', el.id);

let createP = document.createElement('p');
createP.innerText = currentLanguage === 'ua' ? el.descriptionUA :
el.description;
createP.setAttribute('data-description', el.id);

let favoriteButton = document.createElement('button');
favoriteButton.classList.add('favorite-button');
favoriteButton.innerHTML = '<i class="fa fa-star" aria-
hidden="true"></i>';
favoriteButton.setAttribute('data-movie-id', el.id);
favoriteButton.onclick = function(event) {
  event.stopPropagation();
  if (loggedInUser) {
    const movieId = this.getAttribute('data-movie-id');
    if (!loggedInUser.favorites.includes(movieId)) {
      loggedInUser.favorites.push(movieId);
      console.log('Added to favorites:', loggedInUser.favorites);
    }
  } else {
    alert('Please log in to add favorites.');
```

```

else rating.classList.add('red');

let season = document.createElement('h2');
season.classList.add('season');
season.innerText = texts[currentLanguage].seasonStopped + el.season;

let textWithImg = document.createElement('div');
textWithImg.classList.add('wrapperforimg');
textWithImg.appendChild(createPDetail);
textWithImg.appendChild(createImgDetail);

let backButton = document.createElement('button');
backButton.classList.add('back');
backButton.onclick = () => {
  showCards();
  switchLanguage(currentLanguage);
};

let fontBack = document.createElement('i');
fontBack.classList.add('fa-angle-left', 'fa-solid');

let text = document.createElement('h3');
text.innerText = 'Back';

backButton.append(fontBack, text);

cardDetail.append(createH1Detail, rating, season, textWithImg);

cards.append(backButton, cardDetail);

switchLanguage(currentLanguage);
};

card.append(createImg, createH1, favoriteButton);
cards.append(card);
});

updateFavoriteButtonsVisibility();
}

function updateFavoriteButtonsVisibility() {
  const favoriteButtons = document.querySelectorAll('.favorite-button');
  if (loggedInUser) {
    favoriteButtons.forEach(button => {
      button.style.display = 'inline-block';
    });
  } else {
    favoriteButtons.forEach(button => {
      button.style.display = 'none';
    });
  }
}

function shuffle(array) {
  for (let i = array.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [array[i], array[j]] = [array[j], array[i]];
  }
}

```

```

    }
    return array;
}

const preArray = [...defaultData];
const randomArr = shuffle(preArray).splice(0, 3);

let iconCards = document.querySelector('.icon-cards__content');
iconCards.innerHTML = '';

randomArr.forEach((el) => {
    let card = document.createElement('div');
    card.classList.add('icon-cards__item');

    let createImg = document.createElement('img');
    createImg.src = el.imgSrc;

    let createH1 = document.createElement('h1');
    createH1.setAttribute('data-title', el.id);
    createH1.innerText = currentLanguage === 'ua' ? el.titleUA : el.title;

    card.append(createImg, createH1);

    iconCards.append(card);
});

```

### Лістинг файлу generateButtonAction.js

```

import { data } from '../data/data.js';
import { renderData } from './generateCards.js';
import { filterData } from './generateInputs.js';
import { switchLanguage, currentLanguage } from './LanguageSwitcher.js';

export function showCards() {
    const [filters, filteredData] = filterData();

    renderData(filters.length ? filteredData : data);
    switchLanguage(currentLanguage);
}

```

### Лістинг файлу users.js

```

export const users = [];

```

### Лістинг файлу dataInpute.js

```

export const dataInput = [
    {
        label: 'Drama',
        labelUA: 'Драма',
        id: 'Drama',
    },
    {

```

```

    label: 'Triller',
    labelUA: 'Трилер',
    id: 'Triller',
  },
  {
    label: 'Fantastik',
    labelUA: 'Фантастика',
    id: 'Fantastik',
  },
  {
    label: 'Horror',
    labelUA: 'Жахи',
    id: 'Horror',
  },
  {
    label: 'Comedia',
    labelUA: 'Комедія',
    id: 'Comedia',
  },
  {
    label: 'Action',
    labelUA: 'Екшн',
    id: 'Action',
  },
  {
    label: 'Detective',
    labelUA: 'Детектив',
    id: 'Detective',
  },
  {
    label: 'Criminal',
    labelUA: 'Кримінал',
    id: 'Criminal',
  },
  },
];

```

## Лістинг файлу data.js

```

export const data = [
  {
    title: 'La casa de papel',
    titleUA: 'Паперовий будинок',
    imgSrc: 'img/La_Casa_De_Papel.png',
    genres: ['Triller', 'Action', 'Criminal', 'Detective'],
    id: 'La-casa-de-papel',
    season: '5',
    rating: '10',
    description: `The series revolves around several characters whose
stories are unveiled throughout the show. Contrary to what its title may
suggest, La Casa De Papel transcends mere bank heists. Beyond the central
plot of storming a mint, it weaves multiple story arcs, deviations, and
profound developments. Ultimately, the saga doesn't conclude with just one
heist. I found the series profoundly moving, allowing me to experience a
spectrum of emotions, from thrill and anxiety to joy and sorrow. The
narrative is meticulously crafted and brought to a definitive conclusion,
which is rare and commendable. I consider this series a must-watch for
viewers of all tastes due to its exceptional quality.`
  }
];

```

descriptionUA: `Серіал розгортається навколо кількох персонажів, чії історії розкриваються протягом шоу. На відміну від того, що може натякати назва, La Casa De Papel виходить за рамки звичайних пограбувань. Окрім головного сюжету про напад на монетний двір, він включає багато сюжетних ліній, відхилень та значних розвитків. В кінцевому підсумку сага не завершується лише одним пограбуванням. Цей серіал справив на мене глибоке враження, дозволивши відчути цілий спектр емоцій – від хвилювання та тривоги до радості та смутку. Сюжет ретельно продуманий і доведений до логічного завершення, що є рідкісним та вартим захоплення. Я вважаю цей серіал обов'язковим до перегляду для глядачів з будь-якими уподобаннями через його виняткову якість.`,

},

{

title: 'Stranger Things',  
titleUA: 'Дивні дива',  
imgSrc: 'img/Stranger\_Things.png',  
genres: ['Triller', 'Drama', 'Fantastik', 'Detective', 'Horror'],  
id: 'Stranger-Things',  
season: '4',  
rating: '10',

description: `The series revolves around a group of kids in the small town of Hawkins, Indiana, where strange occurrences begin to unfold. The title Stranger Things accurately reflects the show's essence, as it delves into supernatural events and government conspiracies. Alongside the main storyline involving the disappearance of a young boy and the mysterious appearance of a girl with psychic abilities, the show explores several subplots involving friendship, family, and coming of age. The series is thrilling and evocative, engendering a range of emotions such as fear, excitement, and nostalgia for 1980s culture. It is meticulously crafted with a clear conclusion to each season, making it a must-watch recommendation for fans of science fiction and drama.`,

descriptionUA: `Серіал розгортається навколо групи дітей у маленькому містечку Хокінс, штат Індіана, де починають відбуватися дивні події. Назва Stranger Things точно відображає сутність шоу, оскільки воно занурюється в світ надприродних явищ та урядових змов. Окрім головної сюжетної лінії, що стосується зникнення молодого хлопця та загадкової появи дівчини з психічними здібностями, шоу досліджує декілька підсюжетів, пов'язаних з дружбою, сім'єю та дорослішанням. Серіал є захопливим та емоційним, викликаючи такі емоції, як страх, захоплення та ностальгію за культурою 1980-х років. Він ретельно продуманий з чітким завершенням кожного сезону, тому я рекомендую його до перегляду всім шанувальникам наукової фантастики та драми.`,

},

{

title: 'Breaking Bad',  
titleUA: 'Пуститися берега',  
imgSrc: 'img/Breaking\_Bad.png',  
genres: ['Triller', 'Criminal', 'Drama'],  
id: 'Breaking-Bad',  
season: '5',  
rating: '9',

description: `The series revolves around Walter White, a high school chemistry teacher turned methamphetamine manufacturing drug dealer, following his cancer diagnosis. The title Breaking Bad perfectly encapsulates the series, as it explores the transformation of a seemingly ordinary man into a criminal mastermind. Beyond the central plot of drug production and distribution, the show delves into complex relationships, moral dilemmas, and the consequences of choices. It evokes a wide array of emotions, from tension and despair to sympathy and disbelief. The narrative is brilliantly crafted, leading to a profound and conclusive ending. It is an essential recommendation for its outstanding narrative, character

development, and insight into human nature.`,

descriptionUA: `Серіал розгортається навколо Волтера Вайта, вчителя хімії у високій школі, який перетворився на виробника та дилера метамфетаміну після діагнозу раку. Назва Breaking Bad ідеально відображає суть серіалу, оскільки вона досліджує перетворення здавалося б звичайної людини в кримінального генія. Окрім головного сюжету про виробництво та розповсюдження наркотиків, шоу занурюється у складні взаємини, моральні дилеми та наслідки вибору. Воно викликає широкий спектр емоцій, від напруги та відчаю до співчуття та недовіри. Наратив чудово продуманий, призводячи до глибокого та завершеного кінця. Це обов'язкова рекомендація завдяки його видатному сюжету, розвитку персонажів та інсайту в людську натуру.`,

},

{

title: 'Sex Education',  
titleUA: 'Сексуальна освіта',  
imgSrc: 'img/Sex\_Education.png',  
genres: ['Comedia', 'Drama'],  
id: 'Sex-Education',  
season: '3',  
rating: '9',

description: `The series revolves around Otis Milburn, a socially awkward high school student who, despite his own sexual inexperience, starts a sex therapy clinic at his school. The title Sex Education directly reflects the central theme of the show, exploring issues of sexuality, relationships, and personal identity among teenagers. Alongside the main plot involving Otis and his peers, the show also touches on themes of family dynamics, societal expectations, and the journey of self-discovery. The series is engaging and heartfelt, eliciting emotions such as amusement, empathy, and enlightenment. With its well-rounded character development and insightful handling of delicate topics, it is highly recommended for its educational and entertaining value.`,

descriptionUA: `Серіал розгортається навколо Отіса Мілберна, соціально незграбного старшокласника, який, незважаючи на власний сексуальний не досвід, започатковує клініку сексуальної терапії у своїй школі. Назва Sex Education безпосередньо відображає основну тему шоу, досліджуючи питання сексуальності, стосунків та особистої ідентичності серед підлітків. Окрім основної сюжетної лінії, пов'язаної з Отісом та його однолітками, шоу також торкається тем сімейних відносин, суспільних очікувань та подорожі самовідкриття. Серіал захоплюючий та широкий, викликаючи такі емоції, як розвага, співчуття та просвітлення. Завдяки гарно розробленим персонажам та проникливому висвітленню делікатних тем, його високо рекомендують за його освітню та розважальну цінність.`,

},

{

title: 'Peaky Blinders',  
titleUA: 'Загострені кашкети',  
imgSrc: 'img/Peaky\_Blinders.png',  
genres: ['Drama', 'Criminal'],  
id: 'Peaky-Blinders',  
season: '6',  
rating: '10',

description: `The series revolves around the Shelby family, particularly Thomas Shelby, who leads the Peaky Blinders, a notorious gang based in Birmingham after World War I. The title Peaky Blinders aptly encapsulates the series, as it delves into the world of organized crime, political maneuvering, and social upheaval of the era. Beyond the central plot of gang rivalry and expansion, the show explores complex themes such as loyalty, power, and redemption. The series is intense and gripping, evoking emotions such as thrill, admiration, and sometimes fear. With its sharp narrative, strong character development, and authentic period detail, it is highly recommended for its compelling portrayal of early 20th-century British

society and criminal underworld.`,

descriptionUA: `Серіал розгортається навколо сім'ї Шелбі, зокрема Томаса Шелбі, який очолює Peaky Blinders, поганой слави банду в Бірмінгемі після Першої світової війни. Назва Peaky Blinders доречно відображає суть серіалу, оскільки він занурюється в світ організованої злочинності, політичних маніпуляцій та соціальних потрясінь того періоду. Окрім основного сюжету про бандитські суперечки та експансію, шоу досліджує складні теми, такі як вірність, влада та викуплення. Серіал є інтенсивним та захоплюючим, викликаючи такі емоції, як хвилювання, захоплення та іноді страх. Завдяки своїй гострій наративі, міцному розвитку персонажів та автентичному зображенню доби, він високо рекомендований за своє переконливе зображення британського суспільства та кримінального підсвіту початку 20-го століття.`,

},

```
{
  title: 'Power',
  titleUA: 'Влада у нічному місті',
  imgSrc: 'img/Power.png',
  genres: ['Drama', 'Criminal'],
  id: 'Power',
  season: '6',
  rating: '9',
```

description: `The series revolves around James St. Patrick, known as 'Ghost,' a wealthy New York City nightclub owner who lives a double life as a drug kingpin. The title Power symbolizes not only the pursuit of political and economic control but also the personal power struggles within Ghost's life. Beyond the central plot of his dual identity crisis, the show delves into themes of betrayal, love, and ambition. The series is richly layered, evoking emotions such as suspense, anger, and excitement. With its compelling narrative, complex characters, and a deep exploration of the consequences of one's choices, it is highly recommended for its gritty portrayal of the pursuit of power and its impact on personal relationships.`,

descriptionUA: `Серіал розгортається навколо Джеймса Сент-Патріка, відомого як 'Привид', багатого власника нічного клубу в Нью-Йоркському місті, який веде подвійне життя як наркобарон. Назва Power (Влада) символізує не тільки прагнення до політичного та економічного контролю, але й особистісні боротьби за владу в житті Привіда. Окрім основного сюжету його кризи подвійної ідентичності, шоу занурюється в теми зради, любові та амбіцій. Серіал має багатий шари, викликаючи емоції, такі як напруга, гнів та захват. Завдяки своєму захоплюючому наративу, складним персонажам та глибокому дослідженню наслідків вибору, він високо рекомендований за своє суворе зображення прагнення до влади та його вплив на особисті стосунки.`,

},

```
{
  title: 'Elite',
  titleUA: 'Еліта',
  imgSrc: 'img/Elite.png',
  genres: ['Drama', 'Criminal', 'Triller'],
  id: 'Elite',
  season: '5',
  rating: '8',
```

description: `The series revolves around the students at Las Encinas, an elite private school in Spain, where the arrival of three scholarship students disrupts the status quo. The title Elite reflects the exclusive and privileged environment of the school, yet the series delves into much more than just the lifestyles of the wealthy. It explores themes of class conflict, love, jealousy, and crime, as secrets and tensions lead to a murder. The series is provocative and thrilling, stirring emotions such as intrigue and empathy. With its gripping narrative, complex characters, and exploration of societal issues, it is highly recommended for its engaging portrayal of drama and mystery in an educational setting.`,

descriptionUA: `Серіал розгортається навколо студентів Las Encinas,

елітної приватної школи в Іспанії, де прибуття трьох стипендіатів порушує звичний порядок. Назва Elite відображає ексклюзивне та привілейоване середовище школи, проте серіал занурюється в набагато більше, ніж просто життя заможних. Він досліджує теми класового конфлікту, любові, ревності та злочинності, оскільки секрети та напруження призводять до вбивства. Серіал є провокативним та захоплюючим, викликаючи емоції, такі як інтрига та співчуття. Завдяки своєму захоплюючому наративу, складним персонажам та дослідженню суспільних питань, його високо рекомендують за його захоплююче зображення драми та містики в освітньому середовищі.`,

```
    },  
    {  
      title: 'Lock and key',  
      titleUA: 'Замок і ключ',  
      imgSrc: 'img/Lock_And_Key.png',  
      genres: ['Drama', 'Horror', 'Fantastik'],  
      id: 'Lock-and-key',  
      season: '1',  
      rating: '5',  
      description: `The series revolves around the Locke family, who move into their ancestral home, Keyhouse, following the father's mysterious death. They soon discover that the house contains magical keys that offer various powers and secrets. The title Locke & Key hints at the fantastical elements of the series, which aims to blend family drama with supernatural adventure. Despite its intriguing premise involving mystical keys and hidden dimensions, I found the series to be rather dull and it failed to maintain my interest. The narrative seems to lack depth and the pacing can be slow, making it hard to recommend enthusiastically.`,
```

```
      descriptionUA: `Серіал розгортається навколо сім'ї Лок, яка переїжджає до свого родового дому, Ключовий Дім, після таємничої смерті батька. Вони незабаром виявляють, що будинок містить магичні ключі, які пропонують різноманітні сили та секрети. Назва Locke & Key натякає на фантастичні елементи серіалу, який прагне поєднати сімейну драму з надприродною пригодою. Незважаючи на інтригуючу премісу, пов'язану з містичними ключами та прихованими вимірами, мені серіал здався досить нудним і не зміг утримати мій інтерес. Наратив здається поверховим, а темп може бути повільним, що ускладнює його рекомендацію з ентузіазмом.`,
```

```
    },  
    {  
      title: 'Riverdale',  
      titleUA: 'Рівердейл',  
      imgSrc: 'img/Riverdale.png',  
      genres: ['Drama', 'Criminal', 'Detective'],  
      id: 'Riverdale',  
      season: '6',  
      rating: '9.5',  
      description: `The series revolves around the lives of the teenagers in the small, seemingly idyllic town of Riverdale, which is shadowed by dark mysteries and secrets. The title Riverdale suggests a simple town story, but the series quickly diverges into a complex web of intrigue involving murder, mystery, and forbidden romance. While it incorporates elements of classic Archie Comics, it reinvents them with a darker, more mature twist. The series is known for its dramatic twists and turns, which can be engaging. However, some viewers might find the plot occasionally overwrought and the tone inconsistent, which could impact its overall enjoyment.`,
```

```
      descriptionUA: `Серіал розгортається навколо життя підлітків у маленькому, на перший погляд ідилічному містечку Рівердейл, яке огорнуте темними таємницями та секретами. Назва Riverdale натякає на просту історію містечка, але серіал швидко перетворюється на складну мережу інтриг, що включає вбивства, таємниці та заборонені романи. Хоча в ньому використовуються елементи класичних коміксів про Арчі, вони переосмислені з темнішим, більш дорослим поворотом. Серіал відомий своїми драматичними
```



поворотами, які можуть бути захоплюючими. Однак деякі глядачі можуть вважати сюжет часом надто накладеним та стиль неоднорідним, що може вплинути на загальне сприйняття.` ,

},

{

title: 'Better call Saul',  
titleUA: 'Краще телефонуйте Солу',  
imgSrc: 'img/Better Call Saul.png',  
genres: ['Comedia', 'Criminal'],  
id: 'Saul',  
season: '6',  
rating: '10',

description: `The series revolves around the character Jimmy McGill, a struggling lawyer who eventually transforms into Saul Goodman, the morally ambiguous attorney known from Breaking Bad. The title Better Call Saul cleverly hints at this transformation and his eventual role as a 'criminal' lawyer. The show combines elements of legal drama with dark comedy, exploring themes of morality, ambition, and identity. Although it serves as a prequel to Breaking Bad, it stands on its own with a compelling narrative, complex characters, and slow-burning tension. The series is acclaimed for its meticulous storytelling and character development, making it a standout for fans of intricate dramas.` ,

descriptionUA: `Серіал розгортається навколо персонажа Джиммі Макгілла, адвоката, що має труднощі, який з часом перетворюється на Сола Гудмана, морально сумнівного адвоката, відомого з Breaking Bad. Назва Better Call Saul вдало натякає на це перетвілення і його майбутню роль як адвоката 'кримінальних' справ. Шоу поєднує елементи юридичної драми з чорною комедією, досліджуючи теми моралі, амбіцій та ідентичності. Хоча це передісторія до Breaking Bad, вона стоїть окремо з захоплюючим наративом, складними персонажами та повільно наростаючою напругою. Серіал здобув визнання за своє ретельне оповідання та розвиток персонажів, що робить його визначним для шанувальників складних драм.` ,

},

{

title: 'Wednesday',  
titleUA: 'Венсдей',  
imgSrc: 'img/Wednesday.png',  
genres: ['Drama', 'Fantastik', 'Triller'],  
id: 'Wednesday',  
season: '1',  
rating: '9.5',

description: `The series revolves around Wednesday Addams, the iconic daughter of the Addams Family, as she navigates life at Nevermore Academy, a school for outcasts and supernatural beings. The title Wednesday focuses on this character's unique perspective and her adventures, which include solving a spate of mysteries and murders while also exploring her psychic abilities. The show combines elements of dark comedy, mystery, and supernatural, reflecting Wednesday's macabre yet endearing personality. While the series introduces a fresh take on the Addams Family lore, some viewers might find the plot occasionally predictable. However, it's well-appreciated for its stylish presentation, witty dialogue, and the strong performance by the lead actress, making it appealing particularly to fans of gothic and quirky narratives.` ,

descriptionUA: `Серіал розгортається навколо Венсдей Аддамс, іконічної доньки сім'ї Аддамс, коли вона намагається знайти своє місце в академії Невермор, школі для вигнанців та надприродних істот. Назва Wednesday зосереджується на унікальній перспективі цього персонажа та її пригодах, які включають розгадування серії таємниць і вбивств, а також дослідження її психічних здібностей. Шоу поєднує елементи чорної комедії, містики та надприродного, відображаючи макабричну, але привабливу особистість Венсдей. Хоча серіал представляє новий погляд на легенду сім'ї Аддамс, деякі глядачі

можуть вважати сюжет часом передбачуваним. Проте він високо оцінений за свою стильну презентацію, дотепний діалог та сильну гру головної актриси, що робить його особливо привабливим для шанувальників готичних та ексцентричних наративів.`

},

```
{
  title: 'Dark',
  titleUA: 'Темрява',
  imgSrc: 'img/Dark.png',
  genres: ['Drama', 'Triller', 'Detective', 'Fantastik'],
  id: 'Dark',
  season: '3',
  rating: '10',
```

description: `The series revolves around the mysterious disappearances of children in the small German town of Winden, uncovering a complex web of time travel, secrets, and conspiracies that span several generations. The title Dark aptly reflects the series' tone and the profound mysteries it explores. Beyond its central time-travel theme, the show delves into philosophical questions about fate, free will, and the interconnectedness of human lives. While the plot is highly intricate and demands viewer attention, it is praised for its deep storytelling, rich character development, and atmospheric intensity. However, its complexity might be challenging for some viewers. Overall, Dark is highly recommended for those who appreciate a mind-bending narrative that combines drama, science fiction, and thriller elements.`

descriptionUA: `Серіал розгортається навколо загадкових зникнень дітей у маленькому німецькому містечку Вінден, розкриваючи складну мережу подорожей у часі, секретів та змов, що охоплюють кілька поколінь. Назва Dark вдало відображає тон серіалу та глибокі таємниці, які він досліджує. Окрім основної теми подорожей у часі, шоу занурюється в філософські питання про долю, вільний вибір та взаємозв'язок людських життів. Хоча сюжет є дуже складним і вимагає уваги глядача, його хвалять за глибоке оповідання, багате розвиток персонажів та атмосферну напруженість. Проте його складність може бути викликом для деяких глядачів. Загалом, Dark високо рекомендований для тих, хто цінує запутану наративну лінію, що поєднує драму, наукову фантастику та елементи трилера.`

},

```
{
  title: 'Lupin',
  titleUA: 'Люпен',
  imgSrc: 'img/Lupin.png',
  genres: ['Triller', 'Detective', 'Criminal'],
  id: 'Lupin',
  season: '2',
  rating: '8',
```

description: `The series revolves around Assane Diop, who seeks to avenge his father's unjust imprisonment by using his skills as a master of disguise and cunning inspired by the fictional gentleman thief Arsène Lupin. The title Lupin not only refers to the famous literary character but also signifies the clever and stylish approach the series takes in blending heist elements with personal drama. The show is captivating and full of twists, exploring themes of justice, family, and social inequality. While generally well-received for its thrilling pace and charismatic lead, some critics point to occasional plot inconsistencies. However, its engaging storytelling and the protagonist's ingenious schemes make it a recommended watch for fans of adventure and crime drama.`

descriptionUA: `Серіал розгортається навколо Ассана Діопа, який прагне помститися за несправедливе ув'язнення свого батька, використовуючи свої навички майстра перевтілення та хитрості, натхненні фіктивним джентльменом-злодієм Арсеном Люпеном. Назва Lupin не тільки відсилає до відомого літературного персонажа, але й означає розумний та стильний підхід серіалу

у поєднанні елементів крадіжки з особистою драмою. Шоу захоплює та повне поворотів, досліджуючи теми справедливості, сім'ї та соціальної нерівності. Хоча воно загалом добре сприйняте за свій захоплюючий темп та харизматичного головного героя, деякі критики вказують на періодичні недоречності у сюжеті. Однак захоплює оповідання та геніальні плани головного героя роблять його рекомендованим до перегляду для шанувальників пригод та кримінальної драми.`,

},  
];

**ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ**

## ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файла	Опис
Пояснювальні документи	
Кваліфікаційна робота_Бутенко.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота_Бутенко.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і відкомпільовану програму
Презентація	
Презентація Бутенко.pptx	Презентація кваліфікаційної роботи