

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор
ШЕЛЕХОВ

(підпис)

червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційне та програмне забезпечення онлайн-кінотеатру»
здобувача групи ІН-93 Поздєєв Євген Анатолійович

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело.

Євген ПОЗДЄЄВ

(підпис)

Керівник,
в.о завідувача кафедри,
кандидат технічних наук, доцент

Ігор ШЕЛЕХОВ

(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»
В.о. завідувача кафедри
Ігор
ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-93 Поздєєва Євгена Анатолійовича

1. Тема роботи: «Інформаційне та програмне забезпечення онлайн-кінотеатру»
Затверджую наказом по СумДУ від _____
2. Термін здачі здобувачем кваліфікаційної роботи _____
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналітичний огляд, постановка й формування завдань. 2) Огляд технологій, що використовуються при створенні онлайн-кінотеатрів. 3) Розробка клієнтської та серверної частини сайту. 4) Представлення отриманих результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____
(підпис)

Керівник _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналітичний огляд, постановка задачі</i>	15.05 - 19.05	
2	<i>Аналіз сучасних сервісів, інформаційна модель системи</i>	19.05 – 23.05	
3	<i>Реалізація серверної та клієнтської частини сайту</i>	23.05 – 29.05	
4	<i>Представлення отриманих результатів</i>	29.05 – 02.06	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	02.06 – 05.06	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

АНОТАЦІЯ

Записка: 60 стр., 20 рис., 1 додаток, 15 використаних джерел.

Обґрунтування актуальності теми роботи – Тема «Інформаційне та програмне забезпечення онлайн-кінотеатру» завжди важлива та актуальна, оскільки фокус на розробці веб-додатку який забезпечує роботу сервісу для перегляду кінематографічних матеріалів. Додаток має мету надати простий та зручний сервіс для перегляду фільмів.

Об'єкт дослідження – процес створення клієнтської та серверної частини веб-сайту.

Мета роботи – розробка клієнтської та серверної частини сайту за допомогою існуючих засобів розробки.

Методи дослідження – аналіз сайтів аналогів, розробка дизайну та функціоналу веб-додатку.

Результати – створено веб-додаток, що має простий та зрозумілий дизайн, для зручного перегляду кінематографічних матеріалів.

ВЕБ-ДОДАТОК, HTML, CSS, PHP, MySQL

ЗМІСТ

ВСТУП	5
1 АНАЛІТИЧНИЙ ОГЛЯД	6
1.1 Сучасні сервіси для зберігання та перегляду кінематографічної продукції	
1.2 Постановка задачі	11
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	12
2.1 Інформаційна модель системи	12
2.2 Структура бази даних	13
2.3 Бізнес-логіка	16
3 ІНФОРМАЦІЙНЕ І ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	17
3.1 Вибір засобів для програмної реалізації	17
3.2 Програмна реалізація клієнтської частини	18
3.3 Програмна реалізація серверної частини	23
3.4 Тестування системи	29
ВИСНОВКИ	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	33
ДОДАТОК	35

ВСТУП

Актуальність: З розвитком інтернету, люди зрозуміли, що нарешті аби подивитися любимий фільм чи серіал не потрібно чекати його по телевізору або йти до кінотеатрів. У інтернет-просторі почали з'являтися сервіси для перегляду фільмів, серіалів, мультфільмів онлайн. Тому створити власний сайт який буде суміщати в собі простоту та зручність у використанні, у порівнянні з іншими аналогами, для усіх шарів суспільства, гарна ідея.

Об'єкт дослідження: Процес створення онлайн-кінотеатру.

Предмет дослідження: Процес розробки інформаційної моделі, структури бази даних, дизайну та програмна реалізація серверної та клієнтської частини веб-додатку.

Гіпотеза: Якщо взяти усі плюси та мінуси сайти аналогів, узяти від них найкраще та виправити помилки, створити власний дизайн та функціонал, які будуть легко використовувати та мати приємний зовнішній вигляд, то можливо переманити глядачів на свій сайт, якщо він буде достатньо гарним у використанні.

Новизна: На відміну від аналогів, у яких на мою думку не дуже зручний та інтуїтивний інтерфейс, цей онлайн-кінотеатр відрізняється мінімалістичним стилем та дизайном, без зайвих кнопок, реклами і т.д.

Структура: Ця робота складається з вступу, аналітичного огляду, проектування інформаційної системи, інформаційного та програмного забезпечення системи, програмної реалізації клієнтської та серверної частини сайту, висновків, списку використаних джерел та додатків.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Сучасні сервіси для зберігання та перегляду кінематографічної продукції

Проаналізуємо сучасні українські сервіси для зберігання та перегляду кінематографічної продукції.

“Megogo” – медіа-сервіс для перегляду телебачення, кіно, спортивних трансляцій, прослуховування аудіо.

Перше, на що слід звернути увагу, це меню навігації (рис. 1.1)



Рисунок 1.1 – Меню навігації

В цілому не погане меню, просте, приємне для ока, але занадто багато додаткових розділів [3-6], які б краще було розмістити у особистому кабінеті для меншого завантаження користувача інформацією (рис. 1.2).

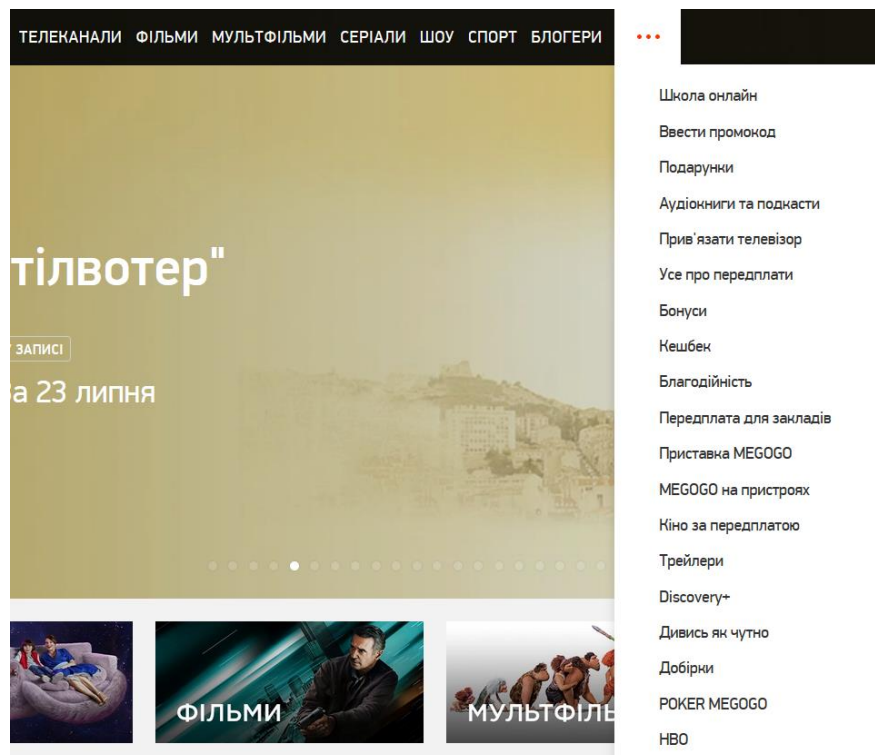


Рисунок 1.2 – Меню особистого кабінету

Зручно, що на головній сторінці (рис. 1.3) є вітрини з прем'єрами [7-9], фільми, які користуються популярністю зараз, та категорії, якщо хочеться обрати якийсь конкретний жанр кіно (рис. 1.4).

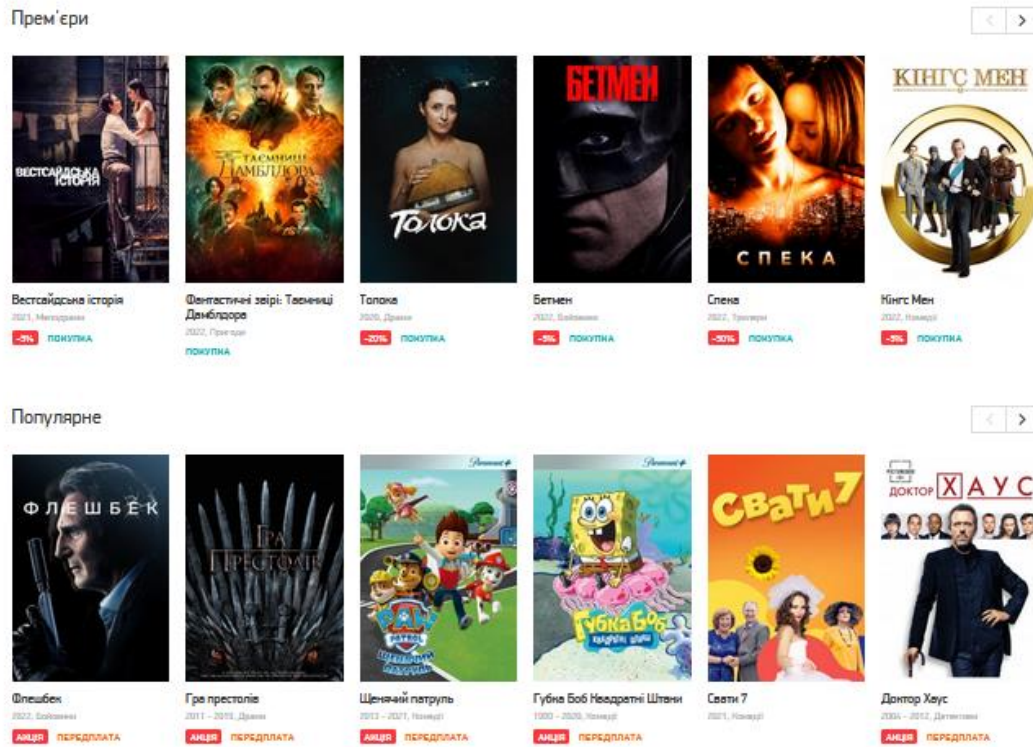


Рисунок 1.3 – Сторінка каталогу



Рисунок 1.4 – Вибір категорії

Сайт працює на комерційній основі, тому треба оформляти щомісячну підписку (рис. 1.5), але відповідна підписка дає доступ до певного списку фільмів, що є мінусом на мою думку. На мою думку ціни достатньо гуманні.

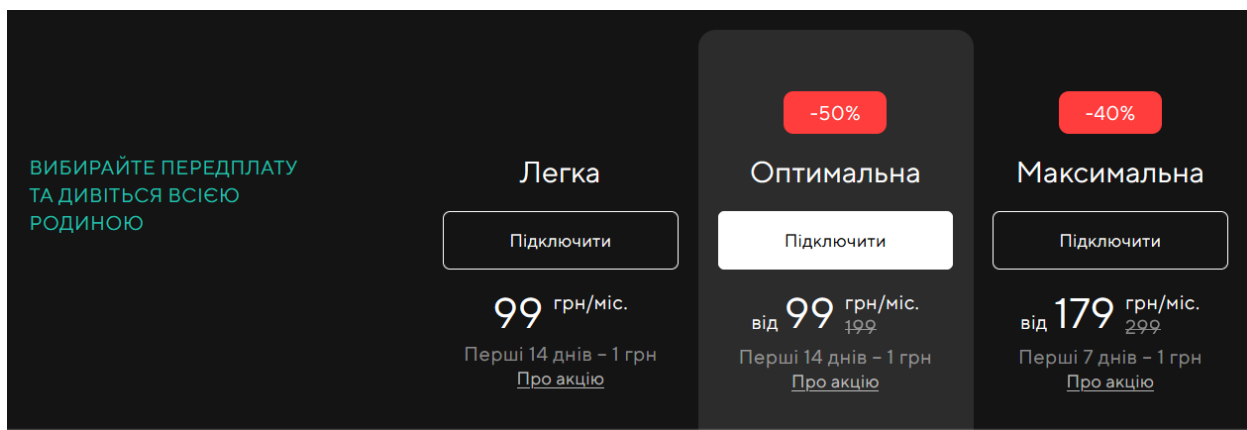


Рисунок 1.5 – Модуль передплати

Проста та швидка реєстрація (рис. 1.6), доступна реєстрація [10-12] через Google, Twitter та Apple ID, що також достатньо зручно, якщо не хочеш втрачати час з підтвердженням даних та додаванням реквізитів оплати.

Реєстрація Вхід ×




Номер телефону

Пароль ПОКАЗАТИ

Натискаючи кнопку "Зареєструватися" ви приймаєте умови [Користувальницької угоди](#) і [Політики обробки персональних даних](#)

Зареєструватися

Вхід через соціальні мережі

→   

Цей сайт захищений сервісом reCAPTCHA, застосовуються [правила використання](#) та [політика конфіденційності](#) від Google.

Рисунок 1.6 – Модуль реєстрації

Національний онлайн-кінотеатр, що відкриває доступ до телебачення, українських фільмів, рейтингових світових кінохітів, мультфільмів і серіалів. «Sweet TV»

Меню навігацій [3, 10-12] (рис. 1.7) майже таке ж як у Megogo, помилка така ж.

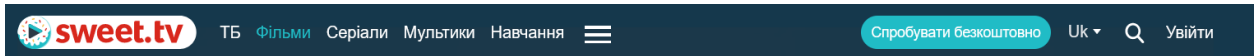


Рисунок 1.7 – Меню навігації

Що стосовно сторінки каталогу [4-5, 9] (рис. 1.8), то у сервісу все теж саме, навіть добавили підбірку фільмів по їх режисерам або автору. Недоліком є те, що ця сторінка дуже довга, замість різних розділів з акторами, авторами, режисерами, усе скинуто до купи та у рандомному порядку.

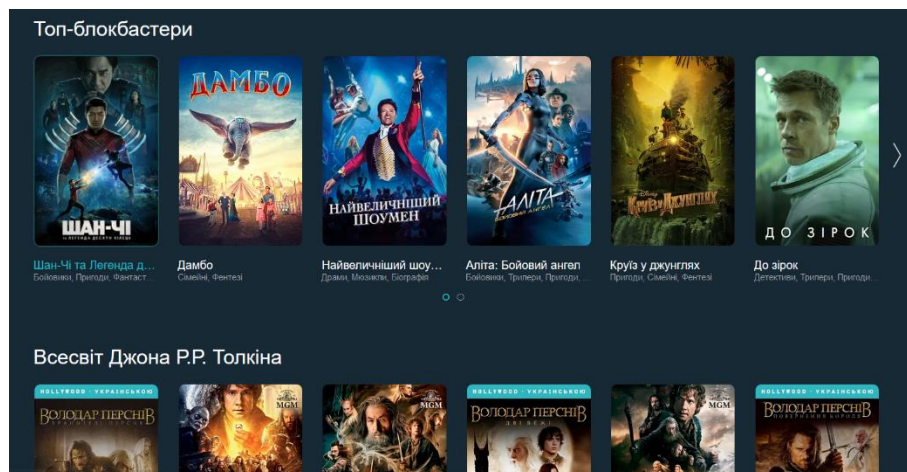


Рисунок 1.8 – Сторінка каталогу

Проте перевагою сайту є список студій, за якими можливо виконати пошук [3-12] (рис. 1.9).

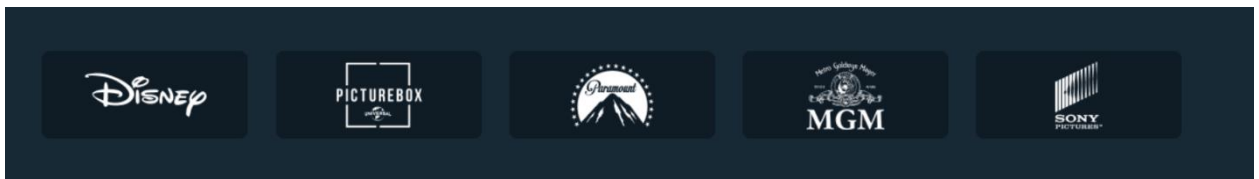


Рисунок 1.9 – Сторінка вибору студій

Стосовно цінової політики (рис. 1.10), ціни значно менші ніж у Megogo, тому це дійсно гарний плюс.

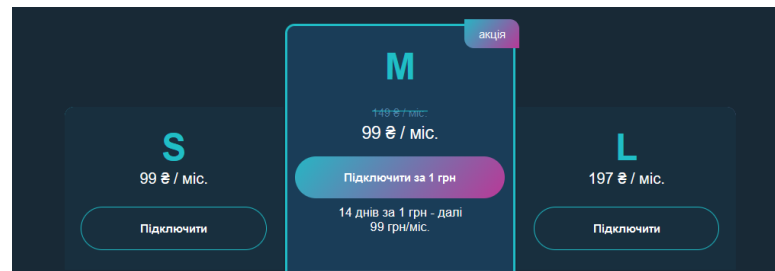


Рисунок 1.10 – Модуль передплати

Реєстрація [10-12] (рис. 1.11) також зручна та проста, способи входу такі
ж

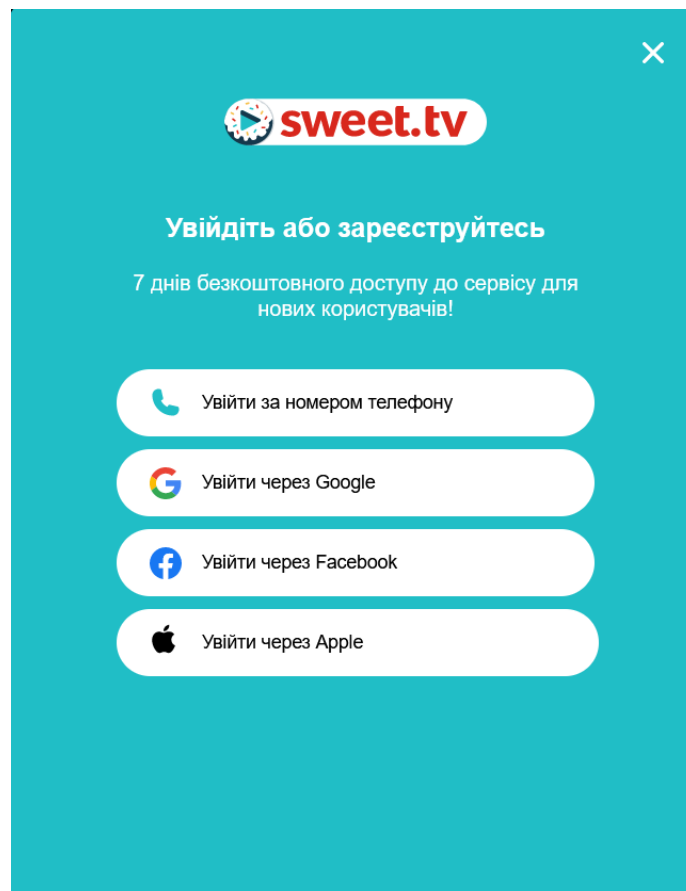


Рисунок 1.11 – Модуль реєстрації

Якщо оцінювати сервіси з точки зору інтерфейсу та дизайну «Megogo» мені подобається більше, але у «Sweet TV» ціна значно менше, тому я віддав би перевагу йому.

1.2 Постановка задачі

Метою роботи є розробка та програмна реалізація інформаційного та програмного забезпечення онлайн-кінотеатру. При цьому основними завданнями, що необхідно виконати для досягнення поставленої мети, є:

1. Розробка інформаційної моделі онлайн-кінотеатру
2. Розробка структури бази даних
3. Визначення бізнес-логіки
4. Вибір засобів програмної реалізації
5. Програмна реалізація клієнтської частини
6. Програмна реалізація серверної частини
7. Тестування системи

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Інформаційна модель системи

Для розробки інформаційної моделі [13-14] системи визначимо основні дії її користувачів.

- Користувач зайшов на сайт
 - Головна сторінка
 - Вхід/реєстрація
 - ✓ Занесення даних до БД
 - Вибір тарифу
 - Сплата
 - ✓ Бібліотека кінематографічної продукції

Подамо ці дії у вигляді діаграм потоків даних DFD [14] (рис. 2.1)

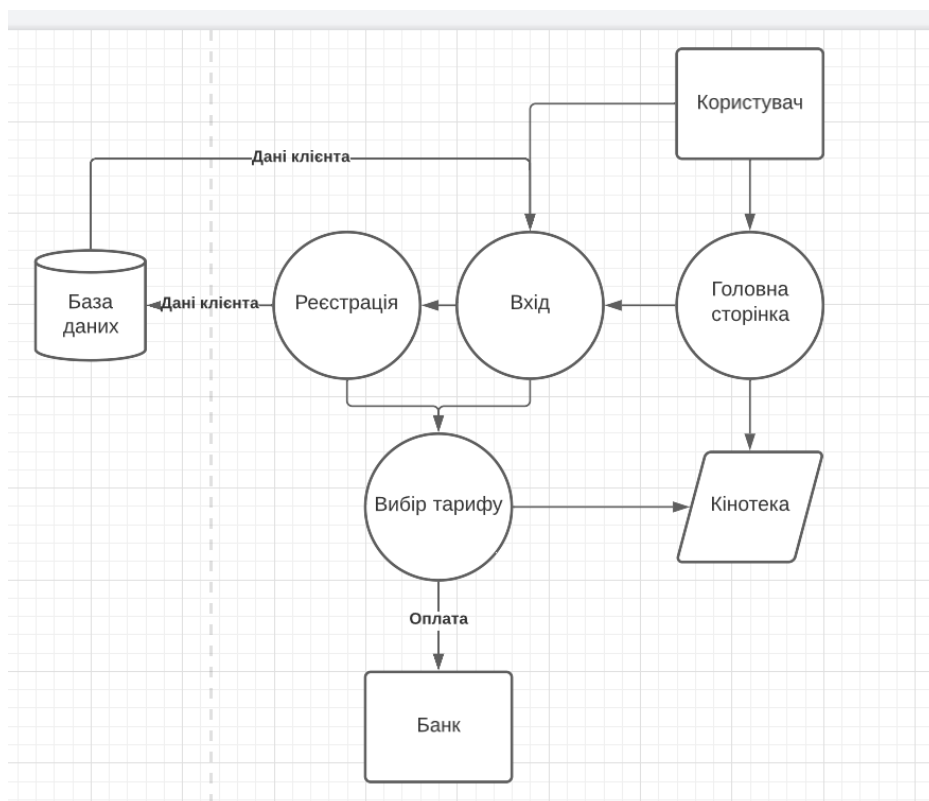


Рисунок 2.1 – Інформаційна модель системи

Виходячи з цієї схеми, клієнт зможе зайти на сайт, зареєструватися, сплатити кошти за тариф щомісячної плати та перейти к перегляду матеріалів, але якщо клієнт потрапить до списків матеріалів без реєстрації та сплати тарифа, то він зможе лише подивитися базу фільмів, але не побачить самі фільми.

2.2 Структура бази даних

Спроектуємо ER-діаграму бази даних [15].

Сутність «Movies» (рис. 2.2) має наступні атрибути:

Id_m;

Name;

Rating;

Genre;

Actors;

Первинний ключ сутності «Movies» це «id_m».

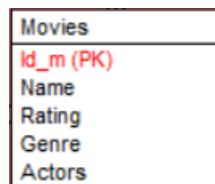


Рисунок 2.2 – Сутність «Movies»

Сутність «User» (рис. 2.3) має наступні атрибути:

Id_u;

Login;

Password;

e-mail;

Tariff;

Date_tariff;

Первинний ключ сутності «User» це «id_u».

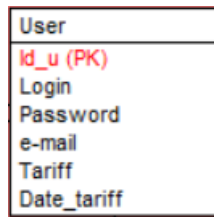


Рисунок 2.3 – Сутність «User»

Сутність «Actors» (рис. 2.4) має такі атрибути:

Id_a;

First_name;

Last_name;

Countre;

Первинний ключ сутності «Actors» це «Id_a».

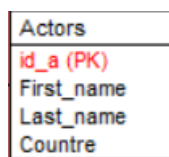


Рисунок 2.4 – сутність «Actors»

Сутність «Tariff» (рис. 2.5) має такі атрибути:

Id_t;

Name;

Cost;

Первинний ключ сутності «Tariff» це «id_t».

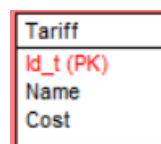


Рисунок 2.5 – сутність «Tariff»

Сутність «Genre» (рис. 2.6) має такі атрибути:

Id_g;

Genre;

Первинний ключ сутності «Genre» це «Id_g».

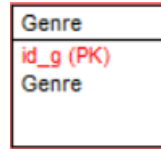


Рисунок 2.6 – сутність «Genre»

ER-діаграма спроектованої бази даних (рис. 2.7).

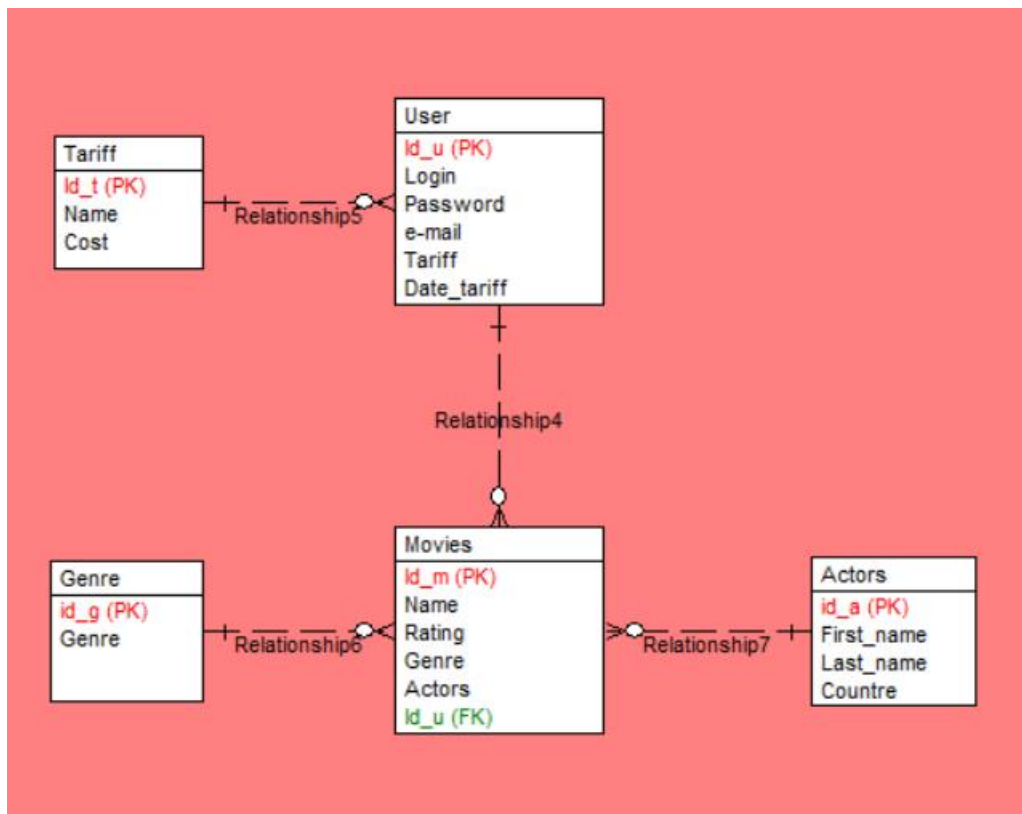


Рисунок 2.7 – ER-діаграма

2.3 Бізнес-логіка

Побудова бізнес-логіки [13-14] онлайн кінотеатру.

1. Користувач заходить на сайт
2. Web-додаток дивиться авторизований користувач чи ні
3. Чекає поки користувач заповнить форму авторизації або реєстрації
4. Обробляє отримані дані
 - Перевіряє вірно чи ні заповнені поля
 - Пошук користувача за логіном
 - Перевірка чи вірний пароль
 - Оновлення дати, часу та пристрою останнього входу
 - Видає помилку якщо поля заповнені не вірно
 - Вносить дані до бази даних
5. Пропонує обрати тарифний план
 - Проводить оплату
 - Визначає до яких матеріалів має доступ з конкретним тарифом
6. Оновлюються дані
7. Користувач може користуватися Web-додатком

3 ІНФОРМАЦІЙНЕ І ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Вибір засобів для програмної реалізації

Для реалізації поставленої мети, тобто створення веб-сайту онлайн-кінотеатру я обрав такі засоби:

- Мови програмування для front-end розробки: HTML/CSS
- Мову програмування для роботи з базами даних та back-end розробки: PHP
- Для реалізації серверної частини обрав ПО: OpenServer
- Бази даних розроблялися в: MySQL

3.1.1 HTML/CSS

HTML (HyperText Markup Language) – це мова розмітки, за допомогою якої створюються веб-сайти та документи. Вона визначає структуру та семантику контенту на веб-сторінці, тобто, розміщення фото, відео, тексту та т.д. Для розмітки використовуються теги, наприклад тег `` використовується як строковий контейнер та у ньому частіше за все тримають текст.

У парі з HTML використовують CSS, в свою чергу CSS (Cascading Style Sheets) – це мова створена для опису стилів, тобто, для задання зовнішнього вигляду веб-сторінок. Само CSS визначає як повинні бути відображенні елементи HTML, включаючи кольори, шрифти, розміри, відступи та інше. Змінюючи всього 1 CSS файл, можливо редагувати вигляд всього сайту. Простіше кажучи, CSS це набір правил для браузера. CSS підтримує концепцію каскаду, тому можливо легко перевизначать та надавати пріоритети стилів для окремих елементів.

3.1.2 PHP

PHP (Hypertext Preprocessor) – це скриптова мова програмування, призначена для розробки веб-додатків. PHP використовують у back-end розробці, тобто на сервері, результати виконання скриптів PHP передаються

до браузера у вигляді HTML сторінки. Особливість PHP полягає у тому, що він може бути вбудований безпосередньо у код HTML файлу, що дозволяє створювати динамічний контент який залежить від умов. PHP має великий функціонал, що спрощує роботу с формами, базами даних, файлами та іншими. PHP може працювати з різними базами даних, наприклад, MySQL або Oracle.

3.1.3 OpenServer

OpenServer – це локальний веб-сервер, який надає середовище для розробки та тестування веб-додатків. Він забезпечує пакетне рішення, яке включає в себе веб-сервер Apache, базу даних MySQL, мову програмування PHP та інші компоненти, необхідні для роботи з динамічними веб-сайтами. Також він спрощує процес налаштування та управління веб-сервером, він дозволяє створювати віртуальні хости та налаштовувати параметри сервера. Це зручно, оскільки дозволяє вносити зміни та перевіряти функціонал сайту без підключення до веб-сервера.

3.1.4 MySQL

MySQL - це система управління базами даних (СУБД), яка забезпечує зберігання, керування та доступ до структурованих даних. Вона є однією з найпопулярніших відкритих СУБД і широко використовується у веб-розробці. MySQL забезпечує надійне та ефективне зберігання даних. Ця СУБД дозволяє змінювати, видаляти дані в базі даних, а також виконувати складні операції, такі як з'єднання таблиць, фільтрація, сортування та агрегація.

3.2 Програмна реалізація клієнтської частини

Реалізацію клієнтської частини сайту я розпочав з стандартної розмітки HTML, тобто, задав <titile> - текст який буде відображаються на вкладці сайту у браузері, та задаванню <meta> - тегів, ними задав текст який буде виводиться на сторінці браузера у пошуку, теги за якими буде здійснюватися пошук пошукової системи, та атрибут «charset="UTF-8"», який використовується в HTML та інших мовах розмітки для визначення кодування символів, яке

використовується на веб-сторінці. . Тегом <link> під'єднав CSS файл, в якому будуть задані основні стилі сторінки та іконку для вкладки сторінки у вікні браузера.

У тегу <body>, використовується для визначення тіла або контенту веб-сторінки, в першу чергу додав тег <div>, блочний елемент який використовується для групування та стилізації інших елементів, та задав йому "id = wrapper". Це було потрібно для стилізації сторінки, "wrapper" проходить через усю сторінку, то задає границі для контенту на сторінці. Для самого <body> додав "background-image", тобто задав картинку для фону який буде розміщений за "wrapper" та зафіксував її для того щоб при прокрутці сторінки, фон був статичний по усій висоті сторінки. Для "wrapper" задав зовнішні відступи від правого та лівого краю, "margin", задав колір, ти прибрав усе зайве параметром "overflow: hidden".

Створив навігацію на сайті, за для комфортнішого пошуку бажаного контенту у тегу <header>, використовується для визначення заголовка або верхньої частини веб-сторінки, використовуючи тег <nav>, використовується як тег навігації у HTML, а в середину тега <nav> додав список з тегом , маркований список, з тегами , використовуються для елементів списку, які являються посиланнями на інші сторінки сайту, які містять у собі відповідний контент.

Усього елементів п'ять:

- «Логотип сайту»
- «Головна»
- «Фільми»
- «Серіали»
- «Мультфільми»

Також у навігаційній панелі міститься тег <div> , який використовується для кнопок-посилань: «Войти» та «Реєстрація». Що стосується стилізації панелі навігації, то для тегу <nav> було задано параметр

“display:flex”, для того щоб елементи списку були розташовані у рядок, та зовнішній відступ від “top”, тобто від верху. Для самого ж списку , було прибрано стиль списку, та задані зовнішні та внутрішні відступи. Для елементів списку було задано розмір шрифту та внутрішній відступ. Посиланням було прибрано текстові декорації, “text-decoration: none”, та задано колір з шрифтом, також після наведення на посилання буде з’являтися нижнє підкреслення.

Про вході в аккаунт з’являється ім’я користувача, при натиску на нього, відчиняється сторінка профілю. На сторінці профілю розміщено фотокартку, під нею ім’я користувача, та посилання на донат розробникам.

На усіх сторінках присутній рядок під назвою «Останні надходження» з <div> класом «nm» для заголовку <h2> із заданими стилями, тобто, вирівнювання по центру сторінки та колір. У цьому рядку відображаються п’ять останніх доданих фільмів.

Для коректного відображення фільмів, створено конструкцію з <div> тегів. Перший тег з класом “ost” загальний, для усіх вкладених тегів, перший вкладений тег це <div class=”mv-cover”>, з посиланням на сторінку фільму, та постером фільму з встановленим розміром. Наступний тег <div class=”mv-rodpr”>, містить в собі назву фільму, рік випуску, країну що зробила цей фільм та підрозділ, тобто вид матеріалу, мультфільм, фільм або серіал. Для класу “ost” задано такі стилі: зовнішній відступ, положення тексту по центру під постером, колір тексту, “cursor: pointer”, тобто, якщо навести курсор на назву або на постер, то курсор змінить свій зовнішній вигляд, вказано параметр дисплея “inline-block”, зроблено це для вирівнювання фільмів в рядок на однаковій відстані від друг друга, вказано розмір шрифту та сам шрифт, висоту букв, та зовнішній відступ від верху, також вирівнювання усього контенту по верху.

Далі йде «Каталог» усіх матеріалів присутніх у базі, у назви розділу задан параметр заголовку <h1> з класом “ml”, та стилями які задають колір тексту, позицію по центру то зовнішній відступ від верху.

Після заголовку «Каталог» йде пошук серед матеріалів по назві, для пошуку вказано тег <div> з класом “search-form” з заданими стилями дисплею “block”, вирівнювання по центру та відступ від наступного елемента. В пошуку використовується форма з методом “get”, та тег <input>, використовується для створення різноманітних вхідних полів на веб-сторінці, з типом “text”, у цю форму треба ввести назву або частину назви фільму для пошуку на сторінці, пошук здійснюється через натиск кнопки з класом “srbtn” та типом “submit”. Для кнопки вказано такі стилі: колір фону, відключено рамку, та розмір шрифту.

Після пошуку йде сам каталог фільмів, розташованих таким же чином як і в «Останні надходження», але відрізняються класи та задані стилі, наприклад у блочних елементах у каталозі є задана ширина, та застосовано до спільного класу “startmv” параметр “overflow:hidde”, який скриває частини які більші за заданий параметр.

Після каталогу вже йде “footer” де вказано електронну адресу підтримки, та назву самого кінотеатру. Для нього задано відступ від минулого елемента та відступ від низу сторінки, текст вирівняно по центру, вказано розмір шрифту та колір.

Після переходу на сторінку реєстрації, користувача зустрічає дизайн як на інших сторінках, але з формою з методом “POST”, та трьома полями,

- «Ім'я користувача:»
- «Email:»
- «Пароль»

До сторінки з реєстрацією під'єднано власний CSS файл, с заданими стилями : для <body> та “wrapper” все так само, для заголовка <h2> з написом «Реєстрація», задано колір, позицію по центру та розмір шрифту. Для трьох полів, тобто тегів <label> задано колір, та усі класи займають позицію по центру. Також на сторінці присутня мінімальна навігація, у вигляді логотипу

кінотеатру, при натисканні на який користувача переводить на головну сторінку. Також додано <footer>, з тими самими параметрами та стилями.

При переході на сторінку «Вхід» , користувача зустрічає форма для входу в аккаунт, в якій присутньо два <label> тега та три <input> тега, з ім'ям:

- «Ім'я користувача»
- «Пароль»

третьій тег <input> з типом “submit” відповідає за вхід та переадресацію на головну сторінку. Також на сторінці присутня кнопка «Реєстрація», для користувачів які не зареєстровані, вона переводить на сторінку реєстрації. Що стосовно стилів, то стилі такі ж як і на сторінці реєстрації, та має таку ж навігацію та <footer>.

На сторінках навігації,

- «Мультфільми»
- «Фільми»
- «Серіали»

все так само як на головній, відрізняються лише назви заголовку та контент сторінки, але про це у наступному розділі.

При створенні сторінки фільму використав ті ж самі meta теги, переніс меню навігації та <footer>.

Додам назву, використовуючи <div> з класом <div class = “name”> для стилізації назви. Додам <div> з класом <div class = “ops”> який використаю для розміщення постеру фільму. Для назви задав колір та вирівняв по центру сторінки. А для постеру задав “display: flex” для розміщення в рядок з описом та інформацією о фільмі.

Створив блочний елемент в якому буде зберігатися інформація о фільмі, в середині нього створив таблицю використовуючи тег <table>.

Для першої строчки яку позначають тегом <tr> додаю текст «Рейтинг:», у цій строчці зберігається рейтинг фільму на різних площадках, у цьому випадку у нього два сайта з яких взявся рейтинг, тому використовуючи теги

<td>, додає стовпчик в рядок, та , використовується для тексту, створюю два стовбця.

Наступна строчка це «Дата виходу», для цього рядка потрібно лише два стовбця. Використовуючи цей метод с таблицями, додаю ще п'ять строчок:

- «Країна»
- «Режисер»
- «Жанр»
- «Вікові обмеження»
- «Актори»

Для спільного класу для інформації, задано лише ширину, та починаючи з постера зовнішній відступ від лівого краю. Для самого тексту у таблицях, задано колір, розмір шрифту, та зовнішній відступ від верху.

Під постером та інформації о фільмі розміщено коротке описання сюжету фільму загорнутого у . Стилями текст розміщено по центру сторінки, та задано ширину з верхнім зовнішнім відступом.

Для розміщення плеєра, створюю блочний елемент, та використовую тег “video”, задаю ширину для вікна у згорнутому вигляді, вказую на постер для нього та задаю параметр “controls” для використання інтерфейсу плеєру без обмежень.

3.3 Програмна реалізація серверної частини

Серверну частину я реалізовував за допомогою PHP та OpenServer.

Завантаживши OpenServer у налаштуваннях обираємо HTTP протокол “Apache_2.4”, обираємо версію PHP 7.2. Обираємо також базу даних та її версію, я обрав MySQL версією 8.0

Використовуючи засоби OpenServer створюємо базу даних для користувачів та фільмів через SQL менеджер.

При створення таблиці користувачів створюємо чотири поля (рис. 3.1),

- “id”
- “username”

- “email”
- “password”

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию	Комментарий
1	id	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...	
2	username	VARCHAR	50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	
3	email	VARCHAR	50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	
4	password	VARCHAR	32	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	

Рисунок 3.1 – Таблица користувачів

Після підключив базу до файлу “register.php”, спочатку вхід у базу, використовуючи засоби PHP, вказати ім’я хосту, користувача, пароль та саме головне, назву бази даних до якої від’єднуватися, приєднання виконується через команду

```
$conn = mysqli_connect
```

Додаємо строку з перевіркою підключення використовуючи “if”.

Після підключення, використовуючи метод “\$_POST”, збираємо дані введені користувачем у форму реєстрації.

Додаємо перевірку ім’я користувача за допомогою SQL запиту і якщо ім’я зайняте виводимо помилку «Це ім’я користувача вже зайняте».

Тепер потрібно після перевірки занести данні користувача до бази. Використовуючи SQL запит

```
$sql = "INSERT INTO users (username, email, password)
VALUES ('$username', '$email', '$password')";
```

Після додавання даних користувача, перенаправимо його на сторінку входу де він повинен увійти у свій аккаунт.

На сторінці входу також підключаємося до бази та перевіряємо підключення.

Додаємо перевірку, відправлена чи ні форма логіну, потім отримуємо дані з форми, додаємо перевірку, відповідає чи ні комбінація імені користувача і пароля даним в базі даних.

```
$sql = "SELECT * FROM users WHERE username='$username'
      AND password='$password'";
$result = mysqli_query($conn, $sql);
```

Якщо користувач існує, встановлюємо сесію та перенаправляємо на головну сторінку. Встановлюємо сесію через строчку `$_SESSION`

З використанням імені та додаєм cookie для користувача, він буде онлайн протягом одного години, після того як час вийде, користувач автоматично вийде з акаунту якщо не буде присутній на сайті.

Якщо користувача не існує, виводимо повідомлення «Невірна комбінація імені користувача і пароля».

Додам кнопки виходу з акаунта, використовуючи спосіб з cookie, але тепер у користувача буде відійматися одна година.

Створимо таблицю у базі даних з назвою “films”, додам вісім строчок: “id”, “title”, “year”, “country”, “type”, “image”, “link”, “date_added” (рис. 3.2).

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию	Комментарий
1	id	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...	
2	title	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	
3	year	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	
4	country	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	
5	type	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	
6	image	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	
7	link	VARCHAR	7000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL	
8	date_added	TIMESTAMP		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CURRENT_TIMES...	

Рисунок 3.2 – Таблица фільмів

“id” використовую як “PRIMERY KEY”, “title” це назва фільму, “year” рік випуску фільму, “country” країна виготовлення продукту, “type” вказує

категорію матеріалу, “image” додає постер к фільму, “link” посилання на сторінку з фільмом.

“date_added” – вказує поточну дату та час додавання до бази фільму, потрібна ж для того щоб була можливість додати п’ять останніх доданих фільмів у «Останні надходження»

Після додавання фільму у бази даних с заповненими полями, використовуючи засоби PHP, створю функціонал який буде витягати фільм з бази та автоматично додавати його на сторінку каталогу.

Спочатку записую SQL запит який буде сортувати фільми у порядку спадання, від старіших до новіших фільмів.

Далі додаєм конструкцію “if” щоб перевірити що в базі існують строки з фільмами.

Створюємо цикл “while” який буде безпосередньо витягувати фільми за бази,

використовуючи “echo”, це вбудована функція в PHP, яка використовується для виведення даних на екран або передачі їх до веб-браузера. Вона може виводити рядки тексту, значення змінних або поєднувати їх.

У кожному рядку з функцією “echo” вказуємо конструкцію, за якою будуть відображатися фільми на сторінці веб-браузера.

```
<?php
$sql_latest = "SELECT * FROM films ORDER BY date_added DESC LIMIT 5";
$result_latest = $conn->query($sql_latest);
if ($result_latest->num_rows > 0) {
    while ($row_latest = $result_latest->fetch_assoc()) {
        echo '<div class="ost">';
        echo '<div class="mv-cover"><a href="'. $row_latest["link"] .'>';
        echo '<a href="movie1.html">' . $row_latest["title"] . '</a>';
        echo '<div>' . $row_latest["year"] . ', ' . $row_latest["country"] . ', ' . $row_latest["type"] . '
        echo '</div>';
        echo '</div>';
    }
} else {
    echo "Ничего не найдено.";
}
?>
```

Рисунок 3.3 – Програмна конструкція відображення інформації про фільми

У моєму випадку я використовую “\$row_latest”, це не вбудована функція в PHP, її використовую для позначення змінних в яких зберігаються результати попереднього SQL запиту до бази даних.

Що стосується конструкції, то

- “\$row_latest[“link”]”, додає посилання для постеру,
- “\$row_latest[“image”]”, додає зображення для самого постеру,
- “\$row_latest[“title”]”, надає назву фільму,
- “\$row_latest[“year”]”, виводить рік створення фільму
- “\$row_latest[“country”]”, виводить країну створення фільму
- “\$row_latest[“type”]”, виводить класифікацію матеріалу.

Якщо ж у базі відсутні будь-які строки, тобто в базі відсутні фільми та вона пуста, виведеться повідомлення що «Нічого не знайдено».

Це була структура для категорії «Останні надходження», таку ж структуру використовую для самого «Каталог», але без заданого ліміту в п'ять фільмів та з іншими іменами класів, які були описані висче.

Після у самому кінці файлу треба закрити з'єднання з базою.

```
<?php
$conn->close();
?>
```

На панелі навігації сайту, є поділ на різні типи :

- «Мультфільми»
- «Фільми»
- «Серіали»

При переході на один із розділів, база повинна сортувати матеріали, та виводити лише обрані.

Тому, у файлах підрозділів потрібно вказати “type”, для сторінки «Мультфільми» це :

```

$sql = "SELECT * FROM films WHERE type =
'Mультфільм'";
$result = $conn->query($sql);
?>

```

Цю конструкцію потрібно вставити після підключення до бази даних, воно виведе усі «Мультфільми» з бази на сторінку.

Для сторінки «Серіали» конструкція виглядає так:

```

$sql = "SELECT * FROM films WHERE type = 'Серіал'";
$result = $conn->query($sql);
?>

```

А для сторінки «Фільми» ось так:

```

$sql = "SELECT * FROM films WHERE type = 'Фільм'";
$result = $conn->query($sql);
?>

```

Конструкції відрізняються лише зміненням типом з бази.

Ми використовуємо файли cookie, але якщо не додати відповідні структури, ім'я не буде відображатися на сторінці та не буде можливості потрапити на сторінку акаунту. Тому використовуючи засоби PHP створюємо конструкцію “if” яка буде перевірять файл cookie на присутність входу та ім'я

```

<?php
        if($_COOKIE['user'] == ''):
        ?>

```

Якщо користувач зайшов у аккаунт то йому буде доступна версія навігації з ім'ям, якщо ж користувач не авторизован у нього відобразиться навігація з кнопками «Вхід» та «Реєстрація».

3.4 Тестування системи

В першу чергу проведено юзабіліті тестування, тобто це перевірка на комфортність та правильний зовнішнього вигляду сайту та інтерфейсу.

Іноді при неправильно підібраній кольоровій гаммі виникає відчуття безпорядку, тобто, для комфортного перегляду сторінки користувачем краще обирати однотонні тони які поєднуються між собою. Мій сайт розроблений у темних тонах лише постери виділяються на фоні самої сторінки.

Є помилка у розмішені <footer>, замалий відступ від основного контенту, але не критично.

Елементи сторінки знаходяться чітко на своїх місцях, у визначених рамках елемента “wrapper”.

Усі елементи мають чітко задані розміри, для правильного позиціонування на сторінці та комфортного перегляду та користування.

Всі зображення також мають чіткий розмір, та не відрізняються один від одного.

Для більш комфортного сприйняття можливо додати відступ від фільмів в «Останніх надходжень» до рамки яка проходить на границі сегменту.

Для більшого розвантаження сторінки та сприйняття користувача, сайт розроблений у мінімалістичному стилі, без зайвих посилань, кнопок, реклами і т.д.

Не створена адаптація сайту, тобто сайт може некоректно відображатися на мобільних пристроях, наприклад на телефоні або планшеті.

На сторінці аккаунти виділяється з кольорової гамми аватар та ссилка, можливо замінити, але не критично.

На сторінках реєстрації та входу не стилізовані кнопки та поля для вводу даних.

На сторінці каталогу фільми йдуть у порядку додавання до бази даних, без усіякого сортування.

Тепер перейдемо до тестування функціоналу.

При бажанні зайти на сторінку фільму, переадресація працює і при натисканні на постер і на назву.

Навігація на усіх сторінках сайту повністю справна та переадресує на потрібні сторінки.

Форма реєстрації працює, користувачі заносяться в базу даних, але дані захищені мінімально. Також повністю працює сторінка входу, при введенні вірних даних йде переадресація на головну сторінку, але якщо дані не співпадають, виводиться помилка.

Відсутня форма для відновлення доступу до аккаунту. Якщо користувач забувся пароль або логін, потрапити в свій аккаунт без допомоги адміністрації він не зможе.

При ручному вводі користувача у базу, можливо користуватися цим аккаунтом.

Поле пошуку за назвою без нарікань працює на головній сторінці з каталогом, але при переході на сторінку, наприклад, з серіалами поле пошуку буде шукати по назві не у типі серіал, а також як на головній з усіх фільмів у базі даних.

На сторінці фільму, у плеєра немає вибори якості відео, із за цього можливо втратити користувачів зі слабким інтернетом.

При додаванні фільму в базу даних з вказанням усіх параметрів, фільм коректно відображається на сторінці, при введенні неправильних форматів даних або їх пропуск, фільм також додається до сайту без втрати правильної позиції та конструкції.

Якщо додати новий фільм, то його буде відображено не тільки у каталозі но також і у «Останні надходження», замінивши самий старий фільм з п'яти,

також при видаленні любого з п'яти фільмів, на його місце становиться фільм який має одну з п'яти найменших дат.

При видаленні одного з рядків у базі даних фільмів, зі сторінки його видалить також, зі зміщенням до останнього елемента перед видаленням.

ВИСНОВКИ

Таким чином, в роботі приведено початкові матеріали для розробки онлайн-кінотеатру. При цьому виконано такі основні завдання:

1. Виконано інформаційно-аналітичний огляд сучасних сервісів для зберігання та перегляду кінопродукції.
2. Розроблено інформаційну модель онлайн-кінотеатру
3. Розроблено структуру бази даних
4. Визначено бізнес-логіку
5. Виконано програмну реалізацію клієнтської частини
6. Виконано програмну реалізацію серверної частини
7. Виконано тестування системи

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Malevanny N. Online Music Distribution - How Much Exclusivity Is Needed?: A Study of International, European, German and U.S. Copyright Systems and Their Objectives. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2019. – 423 p.
2. Crisp V. Film Distribution in the Digital Age : Pirates and Professionals. – Palgrave Macmillan London, 2015. – 206 p.
3. McFedries Paul. Web Design Playground: HTML & CSS the Interactive Way.– Manning Publications Co., 2019. — 439 p.
4. Nielsen Lene. Personas - User Focused Design, 2nd Edition. — Springer, 2019. — 170 p.
5. Wood Kenny. Confident Web Design. – Kogan Page, 2018. — 268 p.
6. Hong Philippe. Practical Web Design. – Packt Publishing, 2018. — 368 p.
7. Soares M.M., Rebelo F., Ahram T.Z. Handbook of Usability and User-Experience. – Boca Raton: CRC Press, 2022. — 371 p.
8. Hillmann C. UX for XR: User Experience Design and Strategies for Immersive Technologies. – Apress, 2021. — 273 p.
9. Levy Jaime. UX Strategy: Product Strategy Techniques for Devising Innovative Digital Solutions. – 2nd edition. — O'Reilly Media, 2021. — 302 p.
10. Yablonski Jon. Laws of UX: Using Psychology to Design Better Products & Services. – O'Reilly Media, Inc., 2020. — 152 p.
11. Knight Westley. UX for Developers. – Apress, 2019. — 166 p.
12. Wathan Adam, Schoger Steve. Refactoring UI. – 2018. — 252 p.
13. Tsui Frank, Karam Orlando, Bernal Barbara. Essentials of Software Engineering. – 5th Edition. — Jones & Bartlett Learning, 2022, — 450 p.
14. Tsui Frank, Karam Orlando, Bernal Barbara. Essentials of Software Engineering . – 5th Edition. — Jones & Bartlett Learning, 2022, — 450 p.

15. Domdouzis K., Lake P., Crowther P. Concise Guide to Databases: A Practical Introduction . – 2nd edition. — Springer, 2021. — 407 p.

ДОДАТОК

Лістинг програмного коду

```
<?php

$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = "my_database";

$conn = new mysqli($db_host, $db_user, $db_pass, $db_name);
if ($conn->connect_error) {
    die("Ошибка подключения к базе данных: " . $conn->connect_error);
}

//Отримання пошукового запиту из URL-параметру.
$search = $_GET['search'];

// Запит на отримання фільмів за заданою назвою
$sql = "SELECT * FROM films WHERE title LIKE '%$search%'";
$result = $conn->query($sql);

?>

<!DOCTYPE html>
<html>

<head>
<title>WatchMV</title>
<meta charset="UTF-8">
<meta name="description" content="Перегляд кіно онлайн">
```

```
<meta name="keywords" content="кіно, онлайн кінотеатр, фільми">
```

```
<meta name="author" content="Поздєєв">
```

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```
<link rel="icon" type="image/x-icon" href="/favicon.ico">
```

```
</head>
```

```
<body>
```

```
<div id="wrapper">
```

```
<?php
```

```
    if($_COOKIE['user'] == "):
```

```
        ?>
```

```
<header>
```

```
<nav>
```

```
<ul>
```

```
    <li></li>
```

```
    <li><a href="index.php">Головна</a></li>
```

```
    <li><a href="films.php">Фільми</a></li>
```

```
    <li><a href="series.php">Серіали</a></li>
```

```
    <li><a href="mult.php">Мультфільми</a></li>
```

```
</ul>
```

```
<div class="pravo">
```

```
    <a href="login.html">Войти</a>
```

```
    <a href="register.html">Реєстрація</a>
```

```
</div>
```

```
</nav>
```

```
</header>
```

```
<?php else: ?>
```

```
<header>
```

```
<nav>
```

```
<ul>
```

```
<li></li>
```

```
<li><a href="index.php">Головна</a></li>
```

```
<li><a href="films.php">Фільми</a></li>
```

```
<li><a href="series.php">Серіали</a></li>
```

```
<li><a href="mult.php">Мультфільми</a></li>
```

```
</ul>
```

```
<div class="btn-all">
```

```
<li><a class="aboba" href='account.php'>
```

```
<?=$_COOKIE['user']?>
```

```
</a></li>
```

```
<li><a class='btn_logout' href='logout.php'>Вихід з аккаунту</a></li>
```

```
</div>
```

```
</nav>
```

```
</header>
```

```
<?php endif;?>
```

```
<main>
```

```

<div class="slide">
<h2 class="nm"> Останні надходження </h2>

<?php
    $sql_latest = "SELECT * FROM films ORDER BY date_added DESC
LIMIT 5";
    $result_latest = $conn->query($sql_latest);
    if ($result_latest->num_rows > 0) {
        while ($row_latest = $result_latest->fetch_assoc()) {
            echo '<div class="ost">';
            echo '<div class="mv-cover"><a href="'. $row_latest["link"] ."'></a></div>';
            echo '<div class="mv-podp">';
            echo '<a href="movie1.html">' . $row_latest["title"] . '</a>';
            echo '<div>' . $row_latest["year"] . ', ' . $row_latest["country"] . ', ' .
$row_latest["type"] . '</div>';
            echo '</div>';
            echo '</div>';
        }
    } else {
        echo "Ничего не найдено.";
    }
?>

</div>

<h1 class="ml"> Каталог </h1>

```

```

<div class="search-form">
  <form method="get" action="index.php">
    <input class="inbt" type="text" name="search" placeholder="Пошук за
назвою">
    <button class="srbtn" type="submit">Пошук</button>
  </form>
</div>

```

```

<div class="startmv">

<?php
if ($result->num_rows > 0) {

    while ($row = $result->fetch_assoc()) {
        echo '<div class="mv">';
        echo '<div class="mv-cover"><a href="'. $row["link"] ."'></a></div>';
        echo '<div class="mv-podp">';
        echo '<a href="movie1.html">' . $row["title"] . '</a>';
        echo '<div>' . $row["year"] . ', ' . $row["country"] . ', ' . $row["type"]
. '</div>';

        echo '</div>';
        echo '</div>';
    }
} else {
    echo "Нічого не знайдено";
}
?>
</main>

```

```
<footer>
&copy;WatchMV. Онлайн кінотеатр. Усі права захищені.<br>
<span>Служба підтримки : 111111@gmail.com</span>
</footer>
</div>
</body>
</html>
<?php
$conn->close();
?>
body {
    height: auto;
    background-image: url("123.jpg");
    background-attachment: fixed;
    background-size: 100%;
}

#wrapper {
    margin: 0px 300px 0px 300px;
    background-color: black;
    overflow: hidden;
}

.pravo {

    word-spacing: 20px;
    font-weight: bold;
    font-size: 20px;
    padding-left: 30.5rem;
```



```
}
```

```
nav {  
  display: flex;  
  margin-top: 1rem;  
}
```

```
nav ul {  
  list-style: none;  
  margin: 0;  
  padding: 0;  
  display: flex;  
}
```

```
nav li {  
  font-size: 20px;  
  padding: 0 0.50pc;  
}
```

```
nav a {  
  text-decoration: none;  
  color: #666666;  
  font-weight: bold;  
}
```

```
nav a:hover {  
  text-decoration: underline;  
}
```

```
.btn-all {  
  display: flex;  
  padding-left: 27rem;  
}
```

```
.slide {  
  border: solid;  
  border-color: #666;  
  
}
```

```
.nm {  
  color: #666;  
  margin-top: 2rem;  
  text-align: center;  
}
```

```
.ml {  
  color: #666;  
  text-align: center;  
  margin-top: 2rem;  
  
}
```

```
.ost {  
  margin-left: 43px;
```

```
text-align: center;
color: #666;
cursor: pointer;
display: inline-block;
font-size: 14px;
font-weight: bold;
line-height: 18px;
margin-top: 15px;
vertical-align: top;
```

```
}
```

```
.startmv {
  float: left;
  width: 100%;
```

```
}
```

```
.mv {

  color: #666;
  cursor: pointer;
  display: inline-block;
  font-size: 14px;
  font-weight: bold;
  line-height: 18px;
  margin-left: 60px;
  margin-top: 15px;
  overflow: hidden;
  vertical-align: top;
```

```
    width: 180px;
}

.photo {
    margin: 40px 0 0 40px;
}

.nameacc {
    color: #666;
    font-weight: bold;
    font-size: 30px;
    text-align: center;
    width: 300px;
}

.mvst {
    color: #666;
    text-align: center;
}

.namo {
    color: #666;
    text-align: center;
}

.ops {
    display: flex;
}

.poster {
```

```
    margin-left: 90px;
}
```

```
.infa {
    width: 100%;
}
```

```
table {
    width: 100%;
    padding-left: 20px;
    font-size: 25px;
}
```

```
td {
    color: #666;
    padding-top: 15px;
}
```

```
.tdtex {
    margin-left: 20px;
}
```

```
.opis{
    margin-top:20px;
    font-size: 22px;
    color: #666;
    text-align: center;
}
```

```
.opisan{
    display: inline-block;
```

```
    width: 80%;  
  }  
  
.muvik{  
  display: block;  
  text-align: center;  
  margin-top: 25px;  
}  
  
.accofor{  
  display: flex;  
}  
  
.donate{  
  color:#666;  
}  
  
.dont{  
  margin-top:50px;  
  text-align: center;  
  font-size: 30px;  
}  
  
.search-form{  
  
  display: block;  
  text-align: center;  
  margin-bottom: 20px;  
}
```

```
.srbtn{  
  
background-color: #666;  
border: none;  
font-size: 15px;  
  
}  
  
.inbt{  
    background-color: #666;  
    font-size: 15px;  
}  
  
footer {  
    margin-top: 5rem;  
    margin-bottom: 20px;  
    text-align: center;  
    font-size: 1.3rem;  
    color: #666;  
    height: auto;  
}
```

```
<!DOCTYPE html>
<html>

<head>
<title>WatchMV</title>
<meta charset="UTF-8">
<meta name="description" content="Перегляд кіно онлайн">
<meta name="keywords" content="кіно, онлайн кінотеатр, фільми">
<meta name="author" content="Поздєєв">
<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>
<div id="wrapper">

<?php
    if($_COOKIE['user'] == ""):
        ?>
<header>
<nav>
<ul>
<li></li>
<li><a href="index.php">Головна</a></li>
<li><a href="films.php">Фільми</a></li>
<li><a href="series.php">Серіали</a></li>
<li><a href="mult.php">Мультифільми</a></li>

</ul>
```



```
<div class="pravo">
  <a href="login.html">Войти</a>
  <a href="register.html">Реєстрація</a>
</div>
</nav>
</header>

<?php else: ?>

<header>
  <nav>
    <ul>
      <li></li>
      <li><a href="index.php">Головна</a></li>
      <li><a href="films.php">Фільми</a></li>
      <li><a href="series.php">Серіали</a></li>
      <li><a href="mult.php">Мультфільми</a></li>
    </ul>
    <div class="btn-all">
      <li><a class="aboba" href='account.php'>
        <?=$_COOKIE['user']?>
      </a></li>
      <li><a class='btn_logout' href='logout.php'>Вихід з аккаунту</a></li>
    </div>
  </nav>
</header>
```

```
<?php endif;?>
```

```
<main>
```

```
<div class="name">
```

```
<h1>Приклад назви</h1>
```

```
</div>
```

```
<div class="ops">
```

```
<div class="poster">
```

```

```

```
</div>
```

```
<div class="info">
```

```
<table class="tblinf">
```

```
<tbody>
```

```
<tr>
```

```
<td>
```

```
Рейтинг:
```

```
</td>
```

```
<td>
```

```
<span class="tdtex"> IMDb:
```

```
<span>8.8</span>
```

```
</span>
```

```
&nbsp;&nbsp;&nbsp;
```

```
<span class="tdtex"> TAKFLIX:
```

```
<span>9.0</span>
```


</td>

</tr>

<tr>

<td>

Дата виходу:

</td>

<td>

 29.01.2002

</td>

</tr>

<tr>

<td>

Країна:

</td>

<td>

Україна

</td>

</tr>

<tr>

<td>

Режисер:

</td>

<td>

 Іен Б.МакДональд, Кристофер Чулак


```
</td>
</tr>

<tr>
  <td>
    Жанр:
  </td>
  <td>
    <span class="tdtex"> Драма, Детектив</span>
  </td>
</tr>

<tr>
  <td>
    Вікові обмеження:
  </td>
  <td>
    <span class="tdtex">18+</span>
  </td>
</tr>

<div>
  <tr>

    <td>

      Актори:
    </td>
    <td>
```

 Уильям Х. Мэйси, Итан Каткоски, Джереми Аллен Уайт,

Шанола Хэмптон,

Стив Хоуи, Эмма Кинни,

та інші.

</td>

</tr>

</div>

</div>

</table>

</tbody>

</div>

</div>

<div class="opis">

У звичайній київській школі в 11 класі навчаються друзі Маша (Марія Федоченко), Яна (Яна Ісаєнко) та Сеня

(Арсеній Марков). Їм більше подобається проводити час один з одним, ніж у галасливій компанії решти класу. Бажання спілкуватися ще з кимось виникає,

коли Маша закохується в однокласника Сашка (Олександр Іванов).

Мелодрама «Стоп-Земля» є дітищем українського режисера та сценаристки Катерини Горностай.

Починаючи з 2017 року, вона шукала можливість реалізувати свій проект, беручи участь у різноманітних пітчінгах. Акторів обирали серед

столичних школярів, які не мають досвіду роботи в кіно. Із 250 претендентів відібрали 25. Протягом двох місяців тричі на тиждень вони займалися з режисером

та його помічниками. Удосконалювали знання українців та опановували акторську майстерність. Прем'єра відбулася на Берлінському кінофестивалі, де фільм виборов

нагороду Кришталевий ведмідь юнацького журі.

Далі був Одеський фестиваль. Тут фільм отримав гран-прі.

</div>

<div class="muvik">

<video poster="3.jpg" width="80%" controls>

<source src="vid1.mp4" type="video/mp4">

</video>

</div>

</main>

<footer>

<div class="fotfix">

© WatchMV. Онлайн кінотеатр. Усі права захищені.

Служба підтримки : 11111@gmail.com

</div>

</footer>

</div>

</body>

```
<?php
$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = "my_database";

// Підключення до бази даних
$conn = mysqli_connect($db_host, $db_user, $db_pass, $db_name);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Отримання даних з форми реєстрації
$username = $_POST['username'];
$email = $_POST['email'];
$password = $_POST['password'];

// Перевірка, чи ім'я користувача не зайняте
$sql = "SELECT * FROM users WHERE username='$username'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    echo "Це ім'я користувача вже зайняте";
    exit();
}

// $password = md5($password."sss")

// Додавання користувача до бази даних
```

```
$sql = "INSERT INTO users (username, email, password) VALUES  
('$username', '$email', '$password)";
```

```
if (mysqli_query($conn, $sql)) {  
    header('Location: login.html'); // перенаправлення на сторінку логіну  
    exit();  
} else {  
    echo "Помилка: " . mysqli_error($conn);  
}
```

```
mysqli_close($conn);
```

```
?>
```

```
body {  
    height: auto;  
    background-image: url("123.jpg");  
    background-attachment: fixed;  
}
```

```
#wrapper {  
    margin: 0px 300px 0px 300px;  
    background-color: black;  
    overflow: hidden;  
}
```

```
h2 {  
    color: #666;  
    text-align: center;  
    font-size: 40px;  
}
```



```
label {
  color: #666;
}

.navigation {
  margin-top: 50px;
  text-align: center;
}

form {
  text-align: center;
  font-size: 20px;
}

.namele {
  margin-left: -77px;
}

.otst {
  margin-bottom: 15px;
}

.emaot {
  margin-left: 14px;
}
```

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Вхід</title>
  <link rel="stylesheet" href="stylelog.css">
</head>

<body>
  <div id="wrapper">
    <header>
      <nav class="navigation">
        <div class="container">
          <a href="/index.php" class="logo"></a>
        </div>
      </nav>
    </header>

    <form method="POST" action="auto.php">
      <h2>Вхід</h2>
      <label class="namle">Ім'я користувача:</label>
      <input type="text" name="username" required><br><br>
      <label>Пароль:</label>
      <input type="password" name="password" required><br><br>
```

```

        <input class="otst" type="submit" name="login"
value="Увійти"><br>
        <button onclick="window.location.href =
'register.html';">Реєстрація</button>
    </form>

```

```

    <footer>
        <div class="fotfix">
            &copy; WatchMV. Онлайн кінотеатр. Усі права захищені.<br>
            <span>Служба підтримки : 11111@gmail.com</span>
        </div>
    </footer>
</div>
</body>
</html>

```

```

body {
    height: auto;
    background-image: url("123.jpg");
    background-attachment: fixed;
}

```

```

#wrapper {
    margin: 0px 300px 0px 300px;
    background-color: black;
    overflow: hidden;
}

```

```

h2 {

```

```
    color: #666;
    text-align: center;
    font-size: 40px;
}

label {
    color: #666;
}

.navigation {
    margin-top: 50px;
    text-align: center;
}

form {
    text-align: center;
    font-size: 20px;
}

.namele {
    margin-left: -77px;
}

.otst {
    margin-bottom: 15px;
}
```

```

<!DOCTYPE html>
<html>
  <head>
    <title>Онлайн кінотеатр</title>
    <meta charset="UTF-8">
    <meta name="description" content="Перегляд кіно онлайн">
    <meta name="keywords" content="кіно, онлайн кінотеатр, фільми">
    <meta name="author" content="Поздєєв">
    <link rel="stylesheet" type="text/css" href="style.css">

  </head>

  <body>
    <div id="wrapper">

      <header>
        <nav>
          <ul>
            <li></li>
            <li><a href="index.php">Головна</a></li>
            <li><a href="films.php">Фільми</a></li>
            <li><a href="series.php">Серіали</a></li>
            <li><a href="mult.php">Мультфільми</a></li>

          </ul>

          <div class="btn-all">
<li><a class="aboba"
href='account.php'> <?=$_COOKIE['user']?></a></li>

```

```

        <li><a class='btn_logout' href='logout.php'>Вихід з
аккаунту</a></li>
    </div>
</nav>

</header>

<main>

<div class="accofor">
    <div class="photo">
        
        <div class="nameacc">
            <span ><?=$_COOKIE['user']?></span>
        </div>
    </div>

    <div class="dont">
        <span class="donate">Сервіс працює безкоштовно для Українців.
Якщо хочете допомогти проекту, можете задонатити за реквізитами
нижче.</span>
        <a href="https://www.paypal.com/ru/home" target="_blank"></a>
    </div>

</div>
</main>
<footer>
<div class="fotfix">
    &copy; WatchMV. Онлайн кінотеатр. Усі права захищені.<br>

```

```
<span>Служба підтримки : 11111@gmail.com</span>  
</div>  
</footer>  
  
</div>  
</body>
```