

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

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

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Практична реалізація проекту відеохостингу «R.S.Sh.» з функцією
онлайн-кінотеатру

Виконав: студент IV курсу, групи СТс-42
спеціальності 126 Інформаційні системи та

технології

(шифр і назва спеціальності)

Сербін В. С.

(підпис)

(прізвище та ініціали)

Керівник

Готович В. А.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Шимчук Г. В.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

Рецензент

Микитишин А.Г.

(підпис)

(прізвище та ініціали)

Тернопіль
2022

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

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

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 126 Інформаційні системи та технології
(шифр і назва спеціальності)

Студенту Сербіну Володимиру Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Практична реалізація проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру»

Керівник роботи Готович Володимир Анатолійович, кандидат технічних наук, доцент
кафедри КН

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «16» березня 2022 року № 4/7-162

2. Термін подання студентом завершеної роботи 20 червня 2022р.

3. Вихідні дані до роботи базуються на основі вхідних даних замовника розроблюваного проекту відеохостингу, результатах дослідження предметної області, використаних літературних та інтернет-джерелах.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Постановка завдання. 1.1 Аналіз предметної області. 1.2 Вимоги до розробки проекту відеохостингу «R.S.Sh.». 1.3 Вибір інструментів та середовища розробки. 1.4 Визначення варіантів використання проекту відеохостингу «R.S.Sh.». 1.5 Моделювання архітектури проекту відеохостингу «R.S.Sh.». 1.6 Висновок до першого розділу. 2. Проектування та реалізація. 2.1 Моделювання структури проекту відеохостингу «R.S.Sh.». 2.2 Проектування моделі бази даних. 2.3 Проектування інтерфейсу відеохостингу «R.S.Sh.». 2.4 Практична реалізація та тестування проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру. 2.4.1 Реалізація програмної частини проекту. 2.4.2 Встановлення та тестування проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру. 2.5 Висновок до другого розділу. 3. Безпека життєдіяльності, основи охорони праці. 3.1 Вимоги ергономіки до організації робочого місця оператора ПК. 3.2 Вимоги безпеки з охорони праці та профілактичних медичних оглядів для працівників ПК. 3.3 Висновок до третього розділу. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Вступ, постановка завдання, предметна область, вимоги до веб-сайту, вибір інструментів розробки, варіанти використання, моделювання веб-сайту, головна сторінка, реєстрація та авторизація, профіль користувача, відео, підписки, адмінпанель, бібліотека, сторінка фільму, оновлення, пошук, профіль автора, дякую за увагу.

АНОТАЦІЯ

Практична реалізація проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру // Кваліфікаційна робота освітнього рівня «Бакалавр» // Сербін Володимир Сергійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СТс-42 // Тернопіль, 2022 // С. 80, рис. – 50, табл. – 0, кресл. – , додат. – 10, бібліогр. – 30.

Ключові слова: відеохостинг, онлайн-кінотеатр, моделювання структури проекту, проектування моделей бази даних.

Кваліфікаційна робота присвячена дослідженню сучасних тенденцій ринку веб-послуг та практичній реалізації проекту відеохостингу з функцією онлайн-кінотеатру у вигляді веб-сайту.

Мета роботи полягає у тому, щоб вибрати основні функціональні можливості веб-сайтів типу «онлайн-кінотеатр» і «відеохостинг», та об'єднати їх в рамках одного веб-сервісу для перегляду кінопродукції та контенту по ній.

В першому розділі кваліфікаційної роботи виконано завдання з дослідження предметної області та аналізу ринку веб-послуг, визначення актуальності розробки створюваного проекту, потенційної аудиторії користувачів. В ході аналізу потенційних сервісів-конкурентів були встановлені вимоги до веб-сайту у відповідності з їх рівнем та потребами користувачі. Також, в рамках розділу, проведено пошук оптимальних середовища розробки, інструментів та методів, побудовані діаграми потенційних варіантів використання проекту.

В другому розділі кваліфікаційної роботи, шляхом побудови UML-діаграм та моделей, була проведена робота зі створення загальної структури розроблюваного веб-сайту та абстрактного відображення окремих його

елементів. По завершенню моделювання виконано практичну розробку програмної частини проекту, тестування інтерфейсу та модулів функцій.

В третьому розділі наведено відомості на тему охорони праці та безпеки життєдіяльності розробника на робочому місці та за робочим ПК. Висвітлено питання обов'язків розробника на перед, під час та після робочого процесу.

ANNOTATION

Practical implementation of the video hosting project «R.S.Sh.» with the function of an online cinema // Qualification work of the educational level «Bachelor» // Serbin Volodymyr Sergiyovich // Ternopil national technical university named after Ivan Pulyuy, faculty of computer information systems and software engineering, department of computer science, group STs-42 // Ternopil, 2022 // P. 80, pic. – 50, tab. – 0, dra. – , add. – 10, bibliogr. – 30.

Keywords: video hosting, online cinema, project structure modeling, database model design.

Qualification work is devoted to the study of current trends in the web services market and the practical implementation of a video hosting project with the function of an online cinema in the form of a website.

The aim of the work is to select the main functionalities of websites such as "online cinema" and "video hosting", and combine them into one web service for viewing movies and content on it.

In the first section of the qualification work the tasks of research of the subject area and analysis of the web services market, determination of the relevance of the development of this project, the potential audience of users were performed. During the analysis of potential competing services, requirements were set for the website in accordance with their level and user needs. Also, within the section, the search for optimal development environments, tools and methods, diagrams of potential uses of the project.

In the second section of the qualification work, by building UML-diagrams and models, work was carried out to create a general structure of the developed website and abstract display of its individual elements. Upon completion of the simulation, the practical development of the software part of the project, testing of the interface and function modules was performed.

The third section contains information on occupational safety and health of the developer in the workplace and on the work PC. The issues of the developer's responsibilities before, during and after the workflow are covered.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CSS – cascading style sheets (каскадні таблиці стилів).

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

PHP – personal home page tools (інструменти для особистої домашньої сторінки).

SQL – structured query language (мова структурованих запитів).

UML – unified modeling language (уніфікована мова моделювання).

БД – база даних.

ВВ – варіанти використання.

ПЗ – програмне забезпечення.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ	12
1.1 Аналіз предметної області.....	12
1.2 Вимоги до розробки проекту відеохостингу «R.S.Sh.»	17
1.3 Вибір інструментів та середовища розробки	20
1.4 Визначення варіантів використання проекту відеохостингу «R.S.Sh.»	23
1.5 Моделювання архітектури проекту відеохостингу «R.S.Sh.»	27
1.6 Висновок до першого розділу	29
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВІДЕОХОСТИНГУ.....	30
2.1 Моделювання структури проекту відеохостингу «R.S.Sh.»	30
2.2 Проектування моделей бази даних	32
2.3 Проектування інтерфейсу відеохостингу «R.S.Sh.»	34
2.4 Практична реалізація та тестування проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру	36
2.5 Висновок до другого розділу	70
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	71
3.1 Вимоги ергономіки до організації робочого місця оператора ПК.....	71
3.2 Вимоги до безпеки з охорони праці та профілактичних медичних оглядів для працівників ПК	73
3.3 Висновок до третього розділу	75
ВИСНОВКИ.....	76
ПЕРЕЛІК ДЖЕРЕЛ	78
ДОДАТКИ	

ВСТУП

Актуальність теми. Великі відстані між людьми та народами завжди підштовхували людство шукати спосіб швидкого та безперебійного обміну інформацією. Технічний прогрес XIX та XX століть подарував населенню того часу можливість зручно транспортувати листівки, посилки та великі вантажі на далекі відстані. Однак, відправка простого конверту з листком тексту чи фотографією в той час все ще була не достатньо швидкою та фінансово доступною кожній людині.

Наприкінці 1960-х років в світі з'явилась нова технологія, яка дозволяла швидко обмінюватись інформацією на відстані. Інтернет надавав можливість спілкуватись та обмінюватись новинами без необхідності написання фізичного листа, його відправки та довгого очікування відповіді.

З ростом популярності «Всесвітньої мережі» росли і її можливості. В епоху цифрових технологій та віртуального контенту обмін файлами став однією з найбільш корисних для кожної людини переваг нового формату спілкування та пошуку інформації.

Інтернет надав користувачам великий простір та набір інструментів для творчого самовираження в різних сферах діяльності. Особливо популярною сферою для цього став сегмент відео-контенту. З допомогою новітніх цифрових камер люди отримали можливість створювати цікавий, інформативний, або розважальний контент та поширювати його серед великої кількості людей. Зароджувались цілі групи ентузіастів, які спільними зусиллями намагались піднімати якість вихідного матеріалу та розвивати в своїх глядачах інтерес до конкретного типу інформації. З часом, цей рух досягнув такого рівня, що сучасні відеоролики на популярній площадці «YouTube», створені групою з декількох людей, своїми спецефектами та якістю можуть змагатись з повноцінними шедеврами кінематографу.

Творчий підхід багатьох звичайних користувачів Інтернету та висока популярність відеоресурсів привернули увагу великих кіностудій, які у

швидкому рості та розвитку відеосегменту побачили велику вигоду і, в свою чергу, почали більше інтегруватись у «Всесвітню мережу», спонсоруючи існуючі стрімінгові сервіси через показ реклами, або, навіть, створюючи власні. Це сприяло тому, що кінокомпанії почали пропонувати свою продукцію для покупки та показу не лише кінотеатрам та каналам телебачення, але й поширювати її на спеціалізованих сайтах, що дозволить користувачам переглянути будь-який фільм у будь-який час, а корпораціям отримувати дохід з продажу віртуальних копій результату своєї праці без великих витрат, які вони раніше отримували при випуску DVD.

В роки пандемії COVID-19 популяризація стрімінгових сервісів та необхідність перенесення прем'єр кінострічок у «Всесвітню мережу» стала найбільш актуальним рухом та набула великого значення як для кінокомпаній, так і для звичайних споживачів контенту.

Основною метою розробки веб-сайту відеохостингу з функцією онлайн-кінотеатру є об'єднання перегляду кінофільмів та творчого контенту на їх основі в рамках одного спеціалізованого сервісу. Таким чином користувачі, зацікавлені в перегляді відеороликів на конкретну тематику, зможуть швидше та ефективніше здійснювати пошук потрібних їм матеріалів. В свою чергу, люди, зацікавлені у створенні подібного роду контенту, швидше будуть набирати власну аудиторію глядачів.

Для досягнення поставленої мети буде виконано наступний ряд завдань:

- Аналіз предметної області та сфери тематики розроблюваного веб-сайту.
- Визначення використовуваних технологій.
- Аналіз варіантів використання.
- Розробка моделей даних та їх структури.
- Розробка структури сайту та його елементів.
- Реалізація програмного коду з допомогою визначених технологій розробки.
- Виконання перевірки валідності коду.

Для проектування варіантів використання та структури сайту буде застосовано ПЗ з можливістю побудови діаграм уніфікованої мови моделювання UML.

Процес розробки сайту буде здійснюватися з використанням мови програмування PHP для написання функціональної частини елементів сторінок, бази даних MySQL для збереження та виводу даних, мови гіпертекстової розмітки HTML, яка використовується для побудови основних блоків сайту, мови каскадних таблиць стилів CSS та фреймворку Bootstrap для задання налаштувань стилів.

Практичне значення одержаних результатів полягає у об'єднанні двох типів сервісів в один з мінімально необхідним користувачам функціоналом, пріоритетом на зручність використання та фокусом на конкретному напрямку контенту.

РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз предметної області

Перед початком безпосередньої розробки проекту відеохостингу «R.S.Sh.» необхідно провести збір інформації про його предметну область, її розвиток та актуальність. Також необхідно визначитись з призначенням розроблюваного веб-сайту, структурною складовою проекту та потенційними варіантами використання для подальшого формування вимог до проекту, інструментарію і методів розробки.

«Вперше Інтернет з'явився в США наприкінці 60-х років в якості проекту Міністерства оборони країни з побудови мережі для комутації пакетів Агентства перспективних дослідницьких проектів ARPANET. Перший Інтернет-сервер був встановлений цією організацією 1 вересня 1969 року» [1]. В ході активного розвитку можливості новоствореної розробки зростали і, на даний момент, їх можна вважати безмежними. Сьогодні важко знайти інформацію, якої б не було у «Всесвітній мережі»: відповіді на будь-які запитання, навчальні матеріали з усіх галузей науки, завжди нова та актуальна інформація. Користувачі з різних країн світу мають можливість завжди бути на зв'язку та віддалено взаємодіяти між собою в будь-якому місці і в будь-який час. Також «Всесвітня мережа» стала способом заробітку. Кожен може почати створювати цифровий контент будь-якого плану: від написання простих розповідей та аматорських арт-робіт до повноцінних шедеврів анімації та проектів ігрової індустрії, продаючи результати своєї праці зацікавленим особам на відповідних площадках.

Однак, на мою думку, головною можливістю, яку надав людям цей винахід, є зберігання та поширення громістких файлів чи документів. Для прикладу, сьогодні немає необхідності зберігати на полицях десятки папок з файлами, десятки відеокaset чи сотні дисків з фото- та відеоматеріалами. Всі ці

об'єкти тепер завжди можна зберігати в оцифрованому вигляді з можливістю перегляду не лише їх власником, але й іншими користувачами. Особливо це актуально в сфері відео-хостингів.

Дослідивши темпи росту та популяризації «Всесвітньої павутини», а також порядок входу та розвитку різних сфер в мережу, можна виділити декілька основних етапів.

На початковому етапі, до кінця 1990-х років, головною цінністю нової технології були швидкий доступ до інформації та зв'язок між людьми на відстані. В цей час найпопулярнішими сервісами стали пошукові системи, «поштові скриньки», інформаційні ресурси та особисті блоги. Таку тенденцію відображено у рейтингу переглядів сайтів за 1996-й рік на рисунку 1.1.

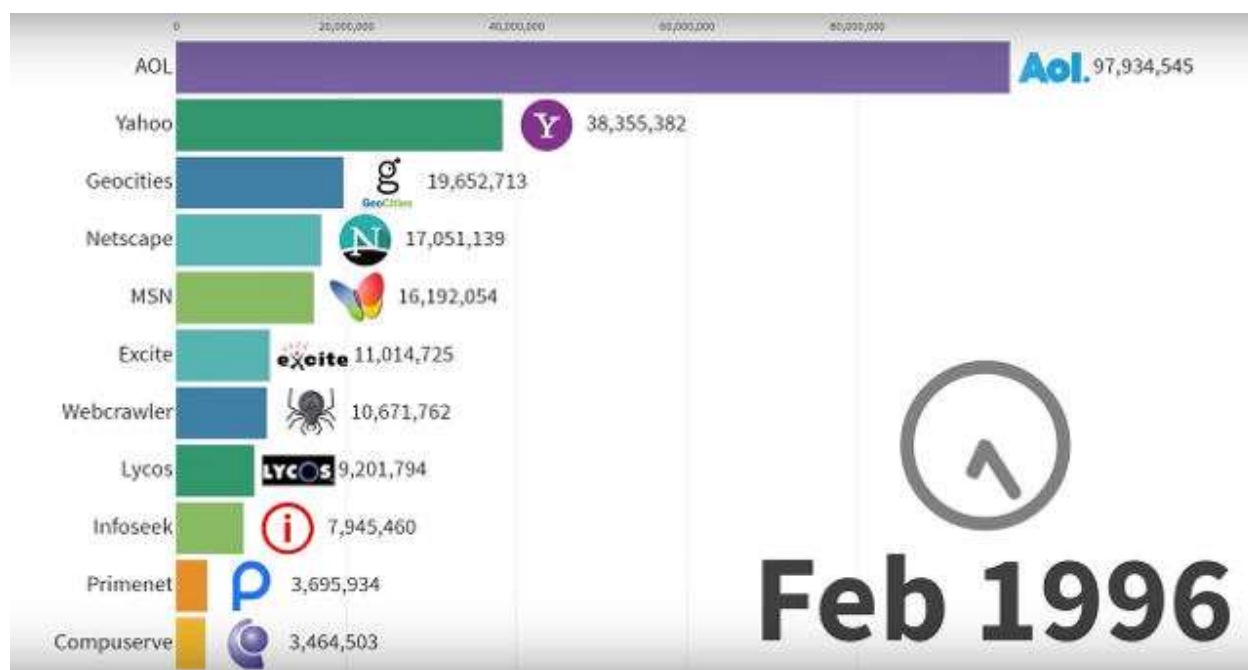


Рисунок 1.1 – Рейтинг популярності сайтів на 1996-й рік

Оскільки Інтернет доволі стрімко розширював коло користувачів, подібних ресурсів ставало все більше і конкуренція зростала. Люди, які витрачали немалі гроші на щомісячну оплату провайдером за доступ до «Всесвітньої мережі», хотіли отримати більше можливостей. В 1999-му році технологічний прогрес почав розповсюджувати доступ в Інтернет і на мобільні

пристрої, що ще більше розширювало охоплення користувачів. Всі ці події заклали подальші напрямки розвитку веб-сервісів «Всесвітньої павутини».

На початку 2000-х років почався новий етап розвитку Інтернету, який був більше орієнтований на створення та просування розважального контенту. Саме тоді масово почали з'являтися сайти з flash та онлайн-іграми. В силу доступності необхідних інструментів для вивчення та використання, сфера розробки браузерних ігор та веб-додатків почала стрімко рости та популяризовувалась серед користувачів Інтернету.

В цей час були створені та активно поширювались площадки продажу різноманітних товарів Amazon та eBay, а також з'явилась нова тенденція серед ігрових студій та незалежних розробників ПЗ щодо продажу віртуальних копій своєї продукції.

На фоні цих подій у «Всесвітній мережі» почали зароджуватись площадки типу Newgrounds і Steam, які надавали користувачам можливість опублікувати власну гру чи повноцінну програму для ПК. Такого роду сайти отримали велику популярність серед користувачів, оскільки надавали останнім повну свободу творчого самовираження без обмежень зі сторони розробників площадки та необхідності платити за публікацію свого продукту. В подальшому ця подія зіграє значну роль в розвитку та становленні сучасного Інтернету.

Велика кількість нововведень викликала багато обговорень в Інтернет-мережі. Більшість користувачів хотіли донести свіжу інформацію до якомога більшої частини людей та обговорити поточні події з ними. Однак, поштові сервіси були доволі обмежені в своєму функціоналі та не давали можливостей людям організовуватись в групи для обговорень певних тем та новин. Форуми, хоч вже й існували на той час, але були не дуже зручними та вузькоспеціалізованими. Вирішення цієї проблеми з'явилося в середині 2000-х років, коли в Інтернеті зародився і різко став популярним сегмент соціальних мереж (див. рисунок 1.2).

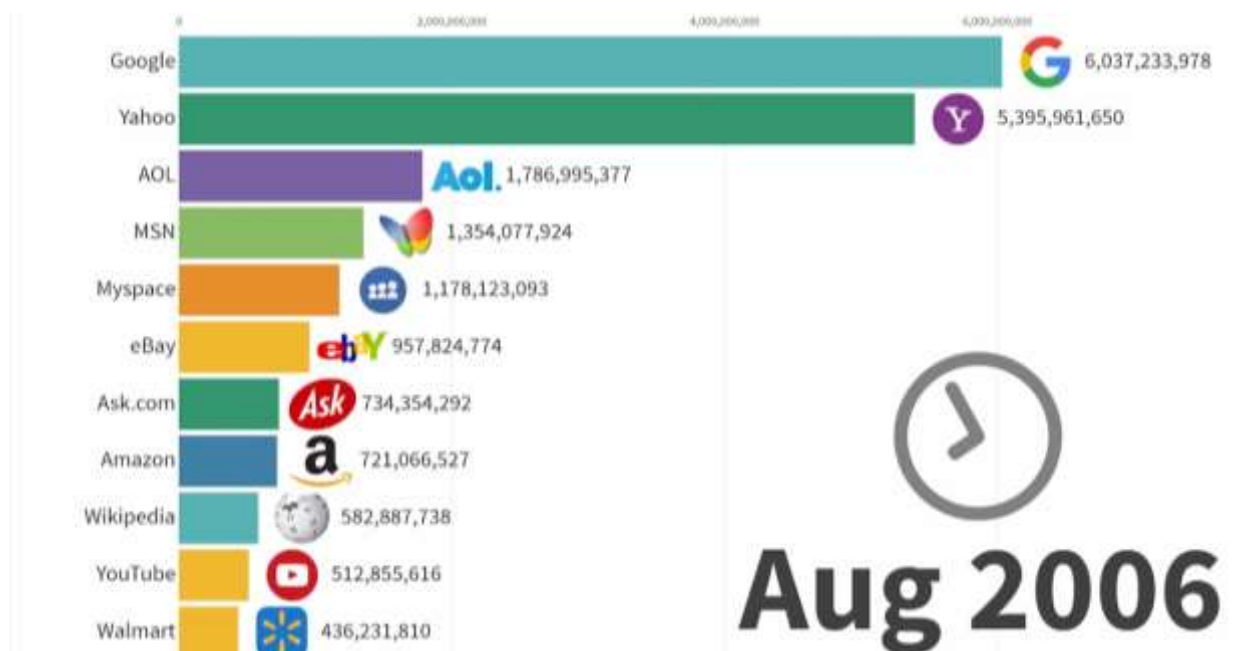


Рисунок 1.2 – Рейтинг популярності сайтів на 2006-й рік

My Space, LinkedIn, Facebook, Twitter стали першими глобальними соціальними сайтами, на яких користувачі отримували велику кількість функцій з оновленим та зручним, відносно поштових сервісів, дизайном. Головними особливостями таких сервісів були можливість публікацій різного роду файлів, зручність спілкування, відкритий пошук та об'єднання людей зі спільними інтересами в групи.

В цей же час на Інтернет-арені з'явився сайт, який мав доповнити ряди соціальних мереж, однак, завдяки своїй ідеї та активності користувачів, запустив новий повноцінний сегмент.

Відомий та популярний зараз як відеосервіс, YouTube напочатку задумувався сайтом, де люди могли б публікувати звичайні відео зі свого життя, вести публічні відео-блоги та спілкуватись між собою за допомогою відео-роликів в межах однієї платформи. Однак, користувачі на хвилі свободи творчого самовираження, яку надавали ігрові та соціальні сервіси, почали завантажувати на сайт різноманітні анімації, музикальні та комедійні відео, випуски новин на різну тематику та навіть документальні фільми. YouTube породив ціле покоління контент мейкерів, вивів творчі процеси на новий рівень, зіграв велику роль в розвитку реклами та популяризації окремих

розважальних, інформаційних, економічних, політичних та науково-технологічних сегментів. Також, сервіс виконав і свою початкову ціль, покращивши взаємодію між користувачами.

З того часу почався період домінації відео-сервісів, які на сьогоднішній день лише набирають популярності та актуальності. Станом на початок 2022-го року YouTube, за виключення пошукової системи Google, лідирує у списку найбільш відвідуваних сайтів з результатом у майже 409 мільярдів відвідувань. Для порівняння, на третьому після YouTube місці знаходиться Facebook з результатом у 265.7 мільярди переглядів. Також, серед відеохостингів, в поданому рейтингу можна виділити стрімінговий сервіс Netflix, який знаходиться на 14-му місці з охопленням у 29.9 мільярдів (див. рисунок 1.3).

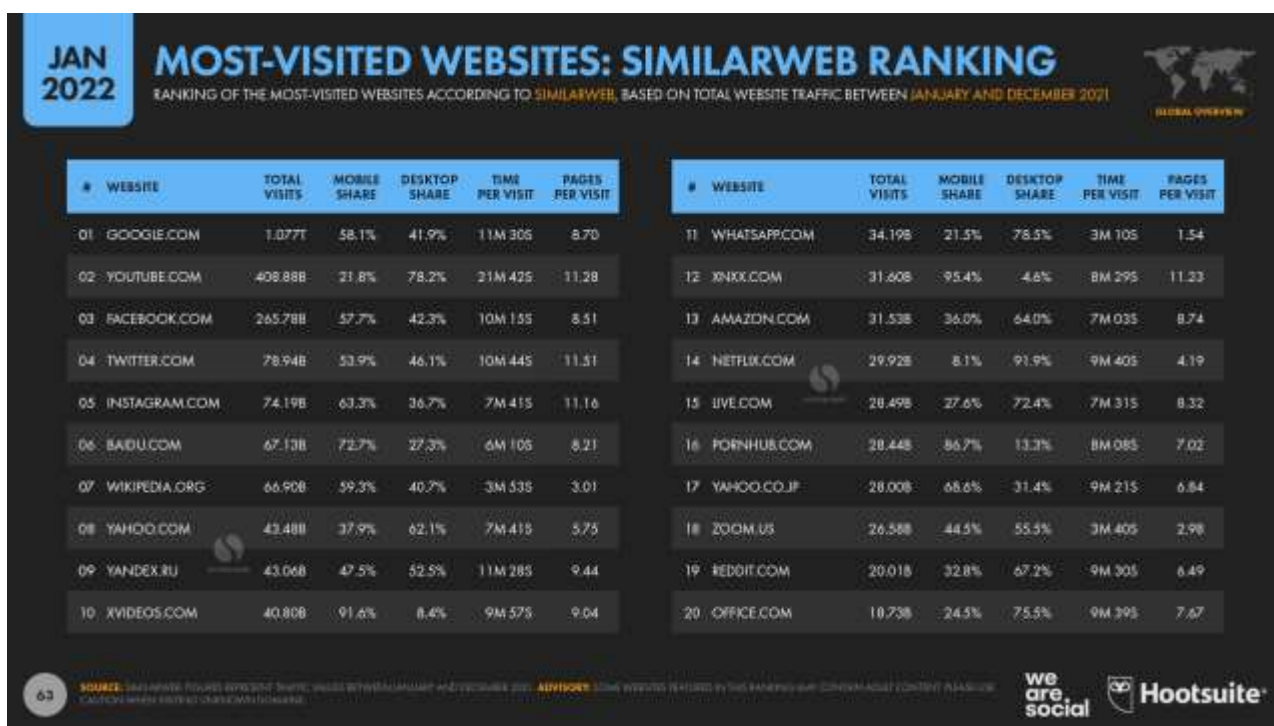


Рисунок 1.3 – Рейтинг популярності сайтів на січень 2022 року

Крім цифр на рейтингових порталах, актуальність цієї сфери відображають великі корпорації, які також активно відкривають свої стрімінгові платформи. Так, медіаконгломерат Walt Disney в 2019-му році запустив сервіс Disney+, Facebook почала роботу над покращенням та

просуванням відео-контенту на своїй площадці, та навіть техногігант Apple і компанія з продажу та доставки товарів Amazon запустили власні сервіси.

В умовах пандемії COVID-19 розробка відео-сервісів та онлайн-кінотеатрів стала ще більш актуальною, оскільки велика кількість користувачів мережі Інтернет була змушена проводити більше часу в своїх домівках. Кількість контенту на відеохостингах також збільшилась в рази, через що якісна інформація на конкретні теми почала губитись серед інших. У зв'язку з цим з'явилась необхідність виділення її в окремі групи. Розробка проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру, який буде поєднувати перегляд фільмів і відео на тему кіно, є спробою створення такого сервісу.

З отриманої інформації виходять висновки, що основною цільовою аудиторією майбутнього веб-сайту є споживачі та творці відеоконтенту, користувачі з необхідністю, або бажанням перегляду актуальних новинок кіно та відеоматеріалів на їх основі у вільному доступі в зручний час. При цьому, сам сервіс отримає виділяючу його на фоні конкурентів особливість, розробка якої і буде проводитись в рамках цієї роботи. Головними діючими сутностями в проекті стануть клієнт, інтерфейс сайту, сервер та база даних. Клієнт отримає можливість взаємодіяти з інтерфейсом для виконання конкретного ряду дій, в той час, як сервер буде приймати запити користувача та надсилати запити до бази даних з метою отримання необхідної інформації.

1.2 Вимоги до розробки проекту відеохостингу «R.S.Sh.»

Проаналізувавши сучасні тенденції росту ринку послуг в Інтернеті, розвиток потреб користувачів та проблеми сучасних сервісів, можна переходити на етап опису проекту та визначення необхідних інструментів для його програмної реалізації. Спершу визначимо ключові вимоги до сайту.

Функціональні вимоги для розробки проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру:

- можливість створення нового профілю користувача та авторизації до існуючого;

- редагування створеного профілю користувача;
- видалення профілю користувача;
- перегляд профілів інших користувачів;
- перегляд сторінок сайту та їх функціональних елементів;
- відтворення відео та фільмів;
- перегляд новин автора;
- можливість завантаження відео з серсеру на носій;
- можливість публікації відео, коментарів, новин автора;
- видалення відео, новин авора, профілю в цілому;
- спосіб оцінювання відео та фільмів з допомогою системи вподобань та коментарів;

коментарів;

- пошук відео, сторінок авторів та фільмів;
- фільтрація пошуку та виводу відео;
- фільтрація пошуку фільмів;
- публікація фільмів через адмінпанель;
- додавання нових відео-тегів через адмінпанель;
- перегляд та вибір дії щодо скарги на контент.

Вимоги до структури веб-сайту відеохостингу:

- реалізація сайту буде побудована на основі модульної структури функціональних елементів та елементів виводу інформації;

- ієрархія сторінок сайту включатиме довільну структуру переходу між ними;

- кожна сторінка повинна містити «шапку» сайту зі списком основних сторінок ресурсу, а також головний блок виводу інформації [2].

Вимоги до захисту та доступу:

- вся введена користувачем інформація в поля вводу у формі реєстрації перед збереженням повинна проходити перевірку регулярними виразами для визначення їх відповідності сформованим вимогам;

- введений при реєстрації користувачем пароль для доступу у профіль перед збереженням в БД повинен проходити обробку шифратором типу md5 та зберігатись у вигляді хешу;
 - обмеження можливості повторної реєстрації за адресою електронної пошти, вже збереженої в БД;
 - обмеження можливості повторної реєстрації нікнейму користувача, вже збереженого в БД;
 - для входу в систему повинні використовуватись унікальні адреса електронної пошти та пароль користувача;
 - обмеження доступу до головної сторінки сайту авторизованих користувачів та примусова переадресація їх у випадку спроби переходу з допомогою поля адреси браузера;
 - створення рангової системи, яка розділятиме користувачів на звичайних та адміністраторів і надаватиме останнім доступ до адміністративної панелі сайту;
 - обмеження для користувачів, які переглядають новину чи відео іншого автора, можливості їх видалення.
- Вимоги до інтерфейсу веб-сайту відеохостингу:
- обов'язкове використання CSS-стилів з ціллю відображення візуальних та функціональних елементів сторінок;
 - при створенні сторінок потрібно дотримуватись їх мінімалістичного наповнення з метою відображення найнеобхідніших функціональних можливостей та інформації;
 - слідувати принципу «все в одному» – тобто намагатись організувати вивід блоків інформації та контенту в рамках однієї сторінки таким чином, щоб користувачу не доводилось переходити на інші;
 - кольорова схема сторінок та елементів сайту повинна відповідати вимогам правильного комбінування кольорів , а також сучасним тенденціям їх використання.

Враховуючи вищевказані вимоги, з ціллю організації доступу до веб-сайту буде достатньо використання користувачем звичайного Інтернет-браузера та стандартного протоколу HTTP.

1.3 Вибір інструментів та середовища розробки

Практично всі, на даний момент, веб-сайти побудовані на основі гіпертекстової розмітки HTML, оскільки вона є стандартом, який підтримується розробниками та призначений саме для побудови елементів сторінок: «елементи HTML слугують елементами для будівництва сторінок HTML. За допомогою конструкцій HTML контент у вигляді зображень, інтерактивних форм та інший може бути вбудований у візуалізовану сторінку» [3].

В порівнянні зі своїми аналогами, наприклад XML, HTML має значну перевагу у вигляді логічних атрибутів та відсутньої чутливості до регістру, що спрощує написання коду, а також присутність тегів, які немає необхідності закривати, що дозволяє краще орієнтуватись в коді та зменшити його об'єм.

Мова каскадних таблиць CSS використовується, як основний та єдиний стандарт для налаштування параметрів стилю елементів сайту: «CSS є основною технологією мережі Інтернет, на рівні з HTML та JavaScript» [4]. Хоч в HTML і є можливість створення стилів прямо у розмітці, мова каскадних таблиць дозволяє задавати параметри зовнішнього вигляду кожного елемента всіх документів сторінок в окремому файлі, що дозволить не навантажувати розмітку зайвим кодом.

Однією з головних переваг CSS є наявність великого вибору фреймворків. Найпопулярнішим з них є Bootstrap. Він активно підтримується зі сторони розробників, про що говорять регулярні оновлення та доповнення бібліотек з великим вибором готових дизайнерських рішень для компонентів сайтів.

Головною ж перевагою вибору скриптової мови PHP в якості мови для розробки функціональної частини проекту є можливість інтеграції HTML-тегів

в код та навпаки. Генерація HTML-сторінок на стороні серверу підвищує безпеку ресурсу, оскільки браузер отримує лише код розмітки та не відображає PHP-код [5].

На сьогоднішній день одним з найпоширеніших та найзручніших методів керування реляційними БД є система MySQL. Для веб-сайту відеохостингу вона найбільш актуальна, оскільки, в першу чергу, використовується для обробки даних динамічних веб-сторінок, показує високі характеристики швидкості, стійкості та простоти використання.

До головних переваг використання технології MySQL можна віднести:

- простоту у встановленні та використанні;
- можливість одночасної роботи з БД необмеженої кількості користувачів;
- може містити до 50 млн. рядків у таблиці;
- виконання команд займає мінімум часу;
- досить ефективна та проста система безпеки.

Також, за рахунок того, що мова відноситься до класу інтерпретованих та має високий рівень продуктивності, швидкість обробки сценаріїв набагато вища, ніж у її аналогів [6].

Для розробки проекту необхідно підібрати спеціалізоване програмне забезпечення, яке надасть можливість швидко та зручно реалізувати всі поставлені завдання по створенню елементів веб-сайту відеохостингу згідно визначених методів вирішення та структури.

Не дивлячись на те, що в якості середовища розробки для створення гіпертекстової розмітки сторінок сайту, а також написання PHP-функцій та операцій достатньо лише простого текстового редактора, для більшої зручності та гнучкості в написанні коду необхідно, щоб цей редактор підтримував наступний ряд інструментів [7]:

- плагіни, наприклад, emmet для швидкого написання коду;
- тип кодування UTF-8;
- можливість розпізнавання та виділення HTML і PHP коду, що

дозволить краще орієнтуватись у розмітці та скриптах;

- можливість автоматичного доповнення написаної частини тексту з метою створення конкретного елемента, або конструкції коду;

- наявність інструменту для роботи з вмістом декількох файлів в одному вікні.

З допомогою аналізу ринку ПО вдалось виявити, що найкраще поставленим вимогам відповідає програма Sublime Text 3. Крім підтримки вищеперерахованих вимог, до її переваг також можна віднести інтуїтивно зрозумілий інтерфейс, доступність усіх функцій в безплатній версії та вміння автоматично закривати парні об'єкти HTML.

Під час розробки веб-сайту відеохостингу для створення бази даних та її підключення до проекту комбінації з вищезгаданих двох сервісів буде цілком достатньо.

Для перегляду сторінок сайту та перевірки функціонування його елементів в ході розробки найкращим варіантом є використання локального серверу Open Server. Такий підхід має багато переваг, оскільки не сприяє грошовим витратам на оплату послуг хостингу.

До переваг вибраного середовища можна віднести:

- містить в собі всі необхідні компоненти для роботи;
- найпростіша версія займає мінімум простору на жорсткому диску та містить такий самий функціонал, як і його старші версії, однак, не містить додаткового ПО, яке може і не знадобитись;

- мінімальне навантаження та використання ресурсів комп'ютера;
- не нав'язує установку зайвого стороннього софту;
- швидкий запуск та робота з базами даних, підтримка системи адміністрування «phpMyAdmin» та MySQL.

Цей програмний комплекс включає в себе ретельно підібраний набір серверного програмного забезпечення, а також неймовірно зручну і продуману керуючу утиліту, яка володіє потужними можливостями з адміністрування і налаштування всіх доступних компонентів [8].

1.4 Визначення варіантів використання проекту відеохостингу «R.S.Sh.»

Далі необхідно створити абстрактне бачення того, як повинна працювати система. Для цього використаємо можливість побудови діаграми прецедентів, або варіантів використання щоб відобразити, що незареєстрований користувач має можливість реєстрації на сайті в той час, коли авторизований та адміністратор мають можливість авторизації. (див. рисунок 1.4).

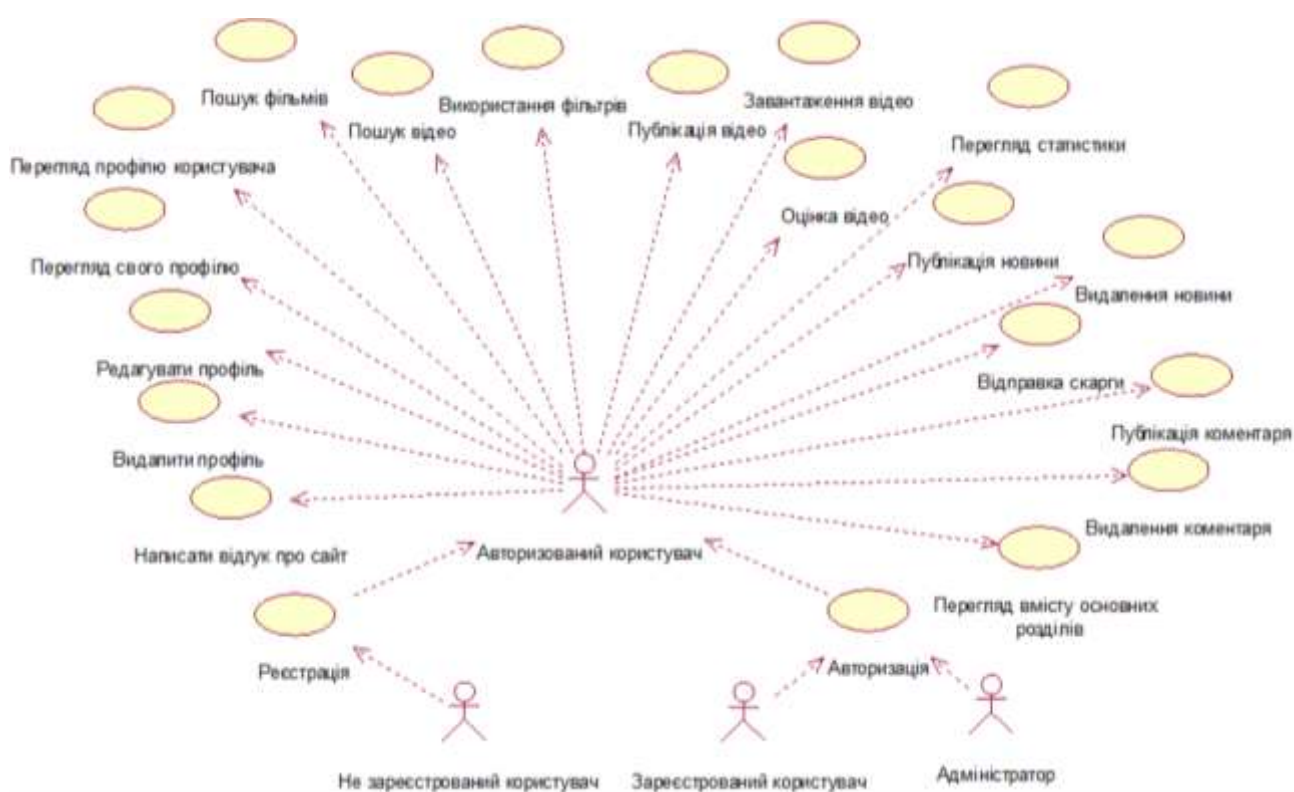


Рисунок 1.4 – Діаграма ВВ для різних типів користувачів

Суть діаграми ВВ полягає в тому, що створювана нами система зображується у вигляді множини сутностей чи акторів, які взаємодіють із системою за допомогою так званих варіантів використання. ВВ використовують для опиу функцій, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система

під час діалогу з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів із системою [9].

Оскільки кожен актор реагує на будь-яку дію за власним алгоритмом, необхідно більш детально розібрати ключові варіанти використання. Для початку будується UML діаграма ВВ реєстрації та авторизації користувачів, як показано на рисунку 1.5.

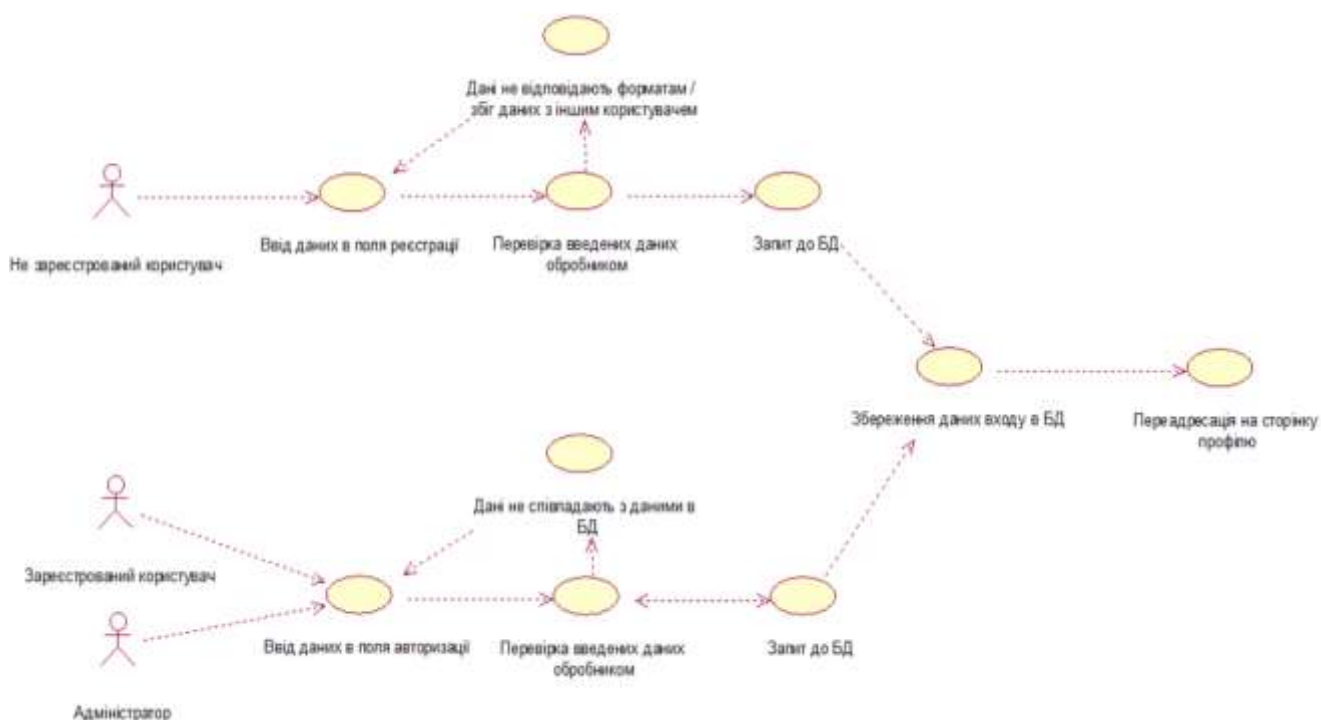


Рисунок 1.5 – Діаграма ВВ реєстрації та авторизації користувачів

На діаграмі зображено шлях, який проходять зареєстровані та не зареєстрований користувачі від вводу даних до переходу на сторінку профілю. В процесі реєстрації користувача після вводу всіх необхідних даних в текстові поля та їх відправки формується запит до обробника, який проводить перевірку інформації для визначення відповідності правилам вводу щодо спеціальних полів: паролів та адреси пошти. Якщо отриманий текст одного з полів не збігається з визначеним форматом, або ж при реєстрації паролі з обох форм не співпадають, обробник виведе відповідне повідомлення на індексній сторінці після перенаправлення. У випадку, коли перевірка проходить успішно, дані з

полів реєстрації відправляються до бази даних, де зберігаються. Після цього відбувається перенаправлення користувача на сторінку створеного профілю.

В свою чергу, дані з полів авторизації порівнюються обробником входу з інформацією, збереженою у таблиці зареєстрованих користувачів БД. У випадку збігу всіх значень відбувається перехід до сторінки профілю відповідно до вказаних в поля авторизації даних. При відсутності збігу обробник поверне відповідне повідомлення. Після завершення процесу реєстрації, або авторизації всі користувачі отримують доступ до базового функціоналу сайту, як продемонстровано на рисунку 1.6.

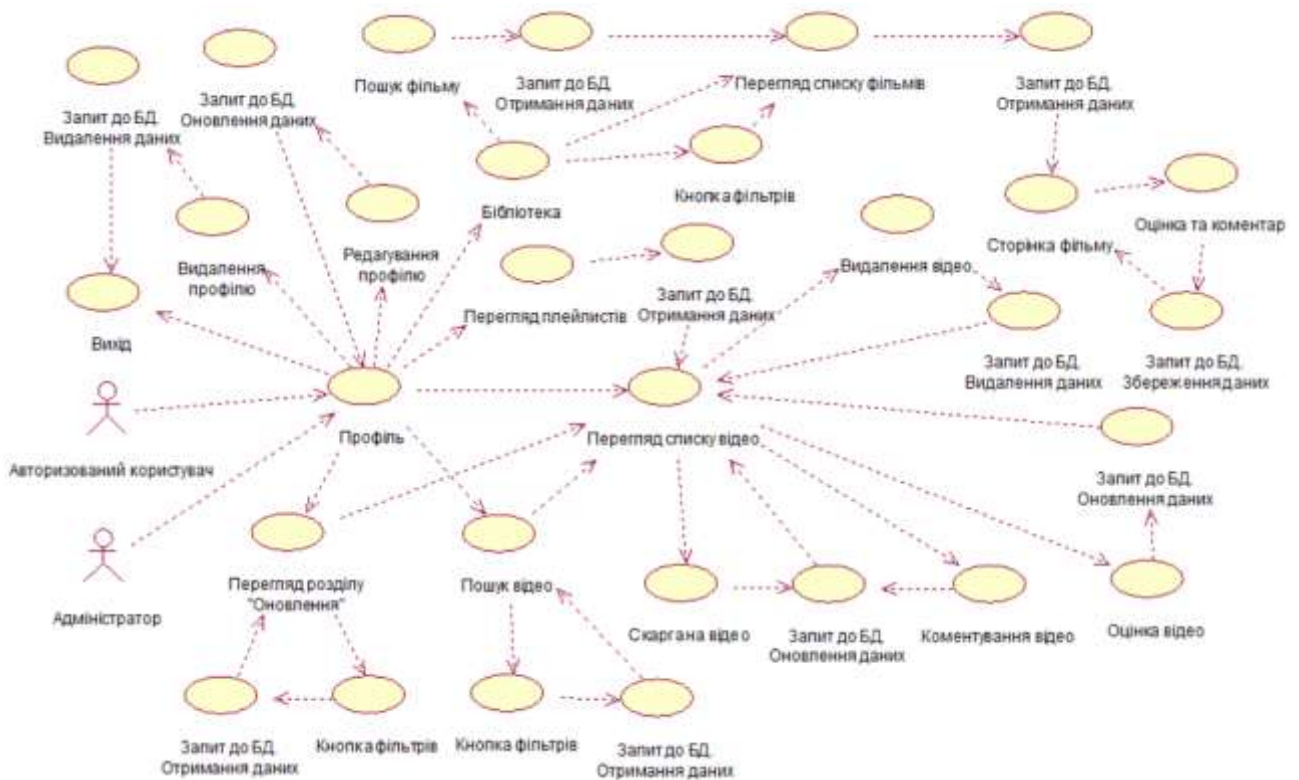


Рисунок 1.6 – Діаграма ВВ після авторизації

Згідно представленої UML діаграми ВВ, адміністратори та звичайні користувачі можуть взаємодіяти з базовими можливостями сайту з допомогою його сторінок, саме:

- перегляд контенту на всіх сторінках та розділах;
- публікація та видалення відеоматеріалів;

- завантаження відео;
- оцінювання контенту;
- скарга на контент;
- коментування відео;
- видалення коментарів;
- створення відгуку про сайт;
- видалення відгуку про сайт;
- чистий пошук відео;
- пошук відео за фільтрами;
- вивід найновіших відео у розділі «оновлення» з можливістю використання фільтрів;
- створення на видалення новин автора;
- пошук та перегляд профілів інших користувачів;
- перегляд статистики профілю та системних повідомлень;
- перегляд плейлистів;
- редагування профілю;
- видалення профілю;
- вихід з профілю;
- пошук та перегляд фільмів у розділі «Бібліотека»;
- оцінювання та коментування фільмів.

В залежності від рангу користувача, функціонал може відрізнятись. На даний момент, окрім стандартного статусу простого користувача, на сайті також може бути призначено окремий ранг адміністратора. Такий профіль отримує можливість переходу та користування розділом адмінпанелі для додавання в БД сайту нових тегів, фільмів, а також перевірки скарг користувачів на контент з подальшим його видаленням чи відправкою попередження.

Повну діаграму ВВ додаткових функцій, який отримує профіль з рангом адміністратора можна переглянути на рисунку 1.7.

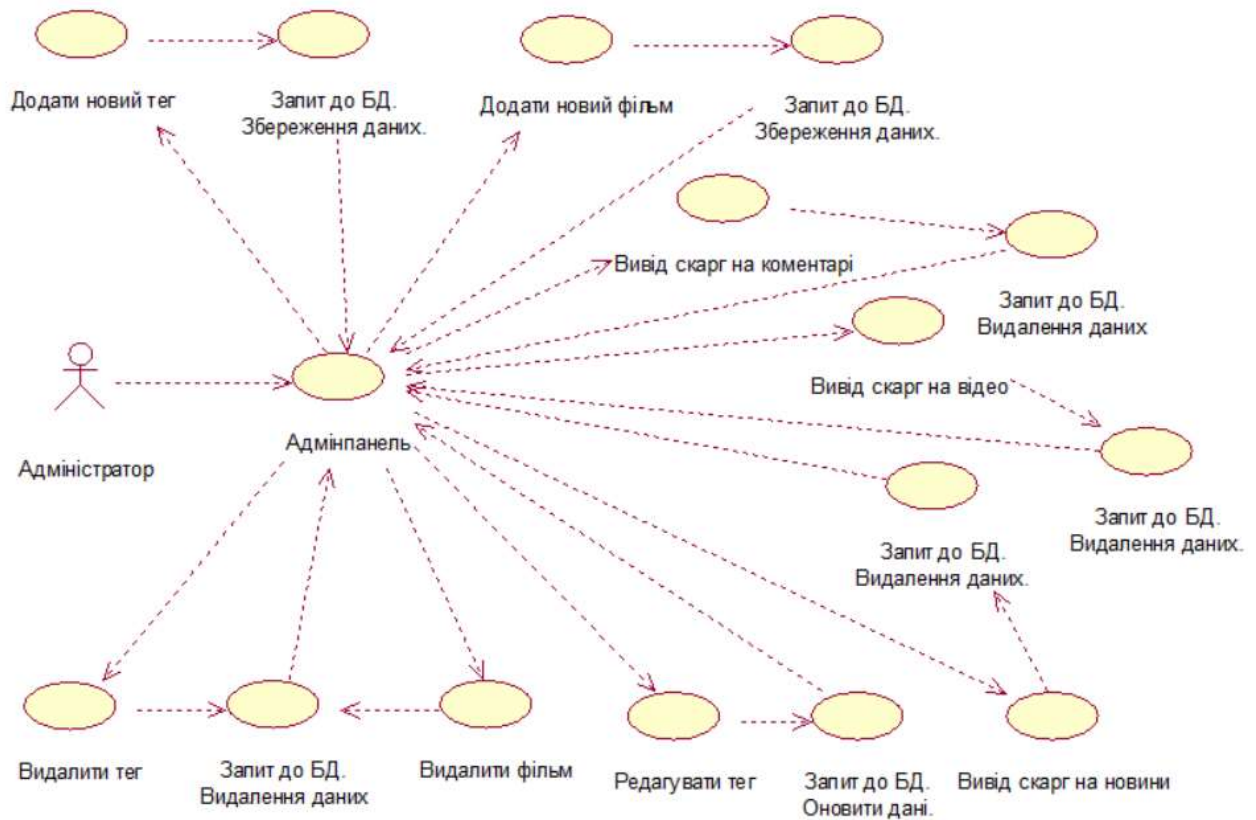


Рисунок 1.7 – Додаткові ВВ для адміністратора

На діаграмах продемонстровано, що практично всі операції виконуються з використанням запитів до БД. Це відбувається, оскільки більша частина функціоналу веб-сайту відеохостингу «R.S.Sh» потребує постійного зберігання та оновлення даних.

1.5 Моделювання архітектури проекту відеохостингу «R.S.Sh.»

Згідно структури проекту та отриманих результатів в ході визначення ключових компонентів його можна поділити за типом трірівневої архітектури, модель якої зображено на рисунку 1.8.

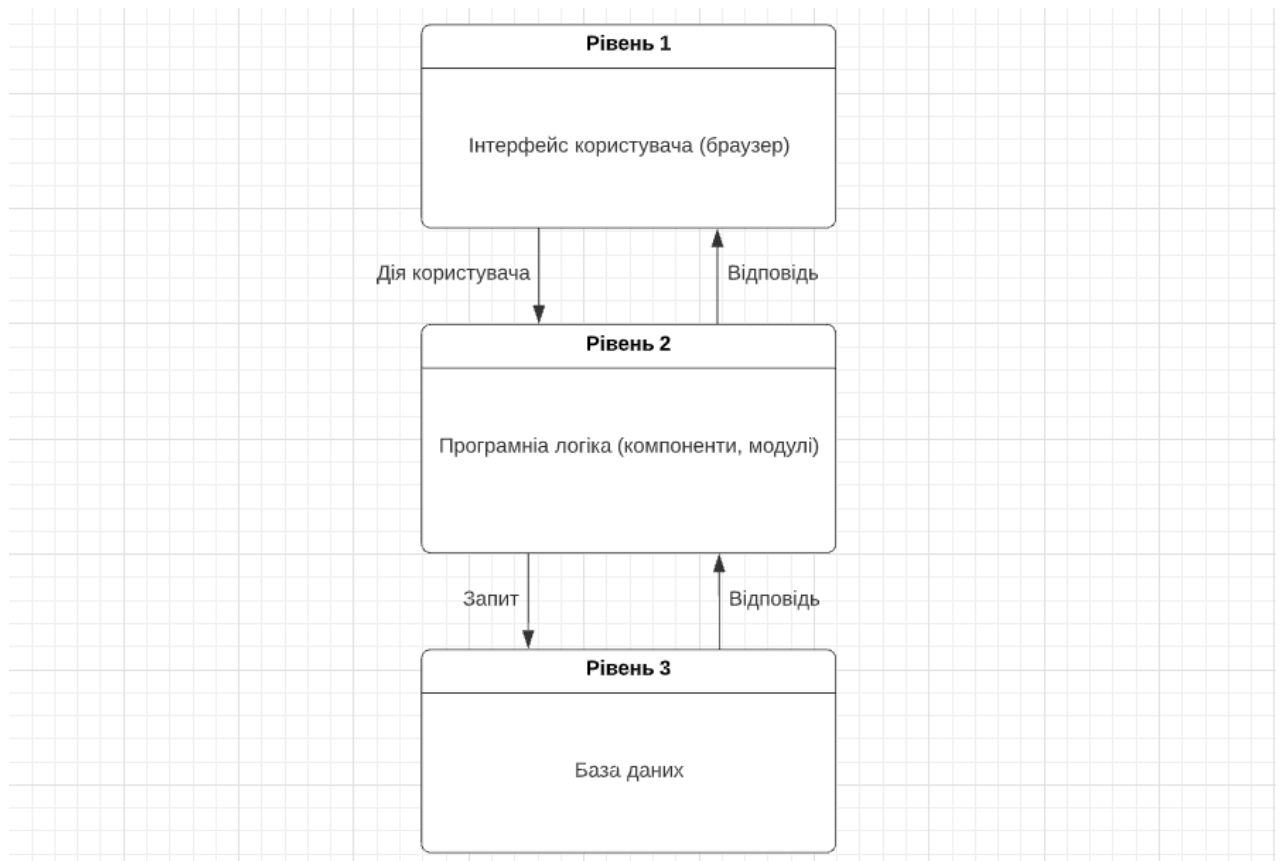


Рисунок 1.8 – Архітектура веб-сайту

Трирівнева архітектура – це тип архітектури «клієнт-сервер», за умовами якої функціональна логіка процесу, доступ до даних, зберігання даних у комп’ютері та користувацький інтерфейс розробляються та підтримуються як незалежні модулі на окремих платформах [10].

На першому рівні розміщений інтерфейс, в якому проводиться відображення інформації та вся взаємодія користувача веб-сторінки з її елементами. Цей рівень слугує простором між людиною та програмною логікою сервісу.

В свою чергу, рівень логіки виконує обробку виконуваних користувачем дій з інтерфейсом. В залежності від результату обробки програмний модуль надсилає відповідь для відображення.

Третій рівень визначається, як рівень даних, тобто БД. На цьому рівні відбувається зберігання інформації, яку відправляє рівень логіки, в розділах даних та її адміністрування в залежності від запитів рівня логіки [11].

Метод використання багаторівневої архітектури дозволяє підвищити безпеку та надійність сервісу, спростити роботу для розробників при підтримці працездатності проекту, його вдосконаленні, та пошуку несправностей [12].

1.6 Висновок до першого розділу

В результаті роботи над першим розділом кваліфікаційної роботи був проведений детальний аналіз тенденцій розвитку Інтернет-мережі з визначенням поточної ситуації та актуальності поточної теми. Після визначення перспективи створення веб-сайту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру виконано моделювання можливих варіантів використання і підбір моделі архітектури. Також були встановлені вимоги до розробки та підбору ПЗ з ціллю створення найкращих умов для реалізації проекту. В подальшому було виконано підбір необхідних технологій, ПЗ та інструментів згідно вищепоставлених параметрів.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВІДЕОХОСТИНГУ

2.1 Моделювання структури проекту відеохостингу «R.S.Sh.»

Метою моделювання організаційних структур є створення таких, які забезпечують надійне функціонування системи із заданими параметрами. Основними завданнями моделювання є формування загальної структурної схеми та регулювання організаційної структури [13].

Для розробки проекту необхідно спочатку описати його програмний комплекс, що буде відображати основні модулі веб-сайту, а також їх взаємодію зі сторінками і таблицями БД.

На сайті визначено 11 основних сторінок:

- index – головна сторінка, з якої відбувається вхід;
- profile – профіль користувача;
- author – профіль автора контенту;
- ratepage – сторінка відгуків;
- updates – виводить останні завантажені на сайт відео;
- search – сторінка пошуку відео;
- library – сторінка бібліотеки фільмів;
- administrator – сторінка для адміністрування сайту, доступна лише окремому колу користувачів;
- film – сторінка фільму;
- likeslist – вивід відео, які сподобались користувачу;
- dislikeslist – вивід відео, які сподобались користувачу.

З цими сторінками взаємодіють наступні модулі:

– модуль з'єднання «connect» – використовується для підключення до БД іншими модулями з ціллю проведення маніпуляцій зі збереження, оновлення чи вибірки даних іншими модулями та сторінками;

– модуль входу – проводить обробку отриманих від користувача даних для виконання входу у профіль. У випадку невірно введених даних

повертається до головної сторінки;

- модуль реєстрації – створює нові дані в БД та відкриває сторінку профілю;

- модуль виходу – виконує вихід з профілю користувача до головної сторінки;

- модуль відправки відгуку – надсилає дані зі сторінки rate page до БД через модуль з'єднання;

- модуль видалення профілю – виконує видалення даних, отриманих зі сторінки користувача через запит до БД;

- модуль перегляду чужого профілю – виводить інформацію та дизайн профілю користувача зі сторони гостя;

- модуль статистики – відображає розширені дані про профіль користувача;

- модуль сповіщень системи – виводить повідомлення про скарги на відео та заблокований контент;

- модуль новин – дозволяє створювати та виводити новини автора контенту;

- модуль редагування даних – виконує редагування даних сторінки користувача з використанням модуля з'єднання;

- модуль виводу відео – отримує дані з БД та виводить на сторінки профілю, пошуку, оновлення та інших список відео, визначений умовою запиту відповідної сторінки;

- модуль оцінок – проводить заміну чи доповнення оцінки для відео;

- модуль відображення списку фільмів – виконує запит до БД та виводить результат на сторінці «бібліотека»;

- модуль відправки коментарів – зберігає дані коментаря користувача для подальшого його виводу під відповідним відео чи фільмом;

- модуль фільтрації – використовує задані користувачем параметри для пошуку чи виводу відео користувачів;

- модулі скарг користувачів – зберігають та виводять адміністратору

скарги користувачів на контент;

- модуль додавання фільму – додає дані нового фільму в БД;
- модуль відео-тегів – організовує створення, редагування та видалення відео-тегів.

З опису схеми формується висновок щодо тісного взаємозв'язку між модулями та сторінками веб-сайту відеохостингу, та виконання кожним з них максимуму необхідних функцій.

2.2 Проектування моделей бази даних

Враховуючи, що розроблюваний веб-сайт відеохостингу буде постійно працювати з великою кількістю інформації, проводити її вибірку, обробку та інші можливі операції, необхідно визначити використовувані моделі даних, сутності та їх призначення, провести опис та моделювання логічної структури даних.

Логічна модель даних використовується у таких випадках, коли необхідно детально описати дані та взаємозв'язки на дуже високому рівні. Дані описуються на дуже абстрактному рівні без їх фізичної демонстрації в БД. В основному він включає сутності та взаємозв'язки між ними, а також атрибути кожної сутності.

Логічна модель даних включає первинні ключі кожної сутності, а також зовнішні ключі. При її створенні перші сутності та їх взаємозв'язки ідентифікуються за допомогою ключів. Потім ідентифікуються атрибути кожної сутності. Логічна модель даних не залежить від системи управління базами даних, оскільки не описує фізичну структуру реальної бази даних [14].

В побудованій моделі даних існують зв'язки між полями типу «один до багатьох». Таблиця users взаємодіє з усіма іншими таблицями, передаючи всім їм id користувача та для деяких – nickname.

Графічний варіант логічної моделі БД зображено на рисунку 2.1.

Специфікація таблиці включає такі деталі, як назва таблиці, номер стовпців, а специфікація стовпців включає ім'я та тип даних [14] (див. рисунок 2.2).

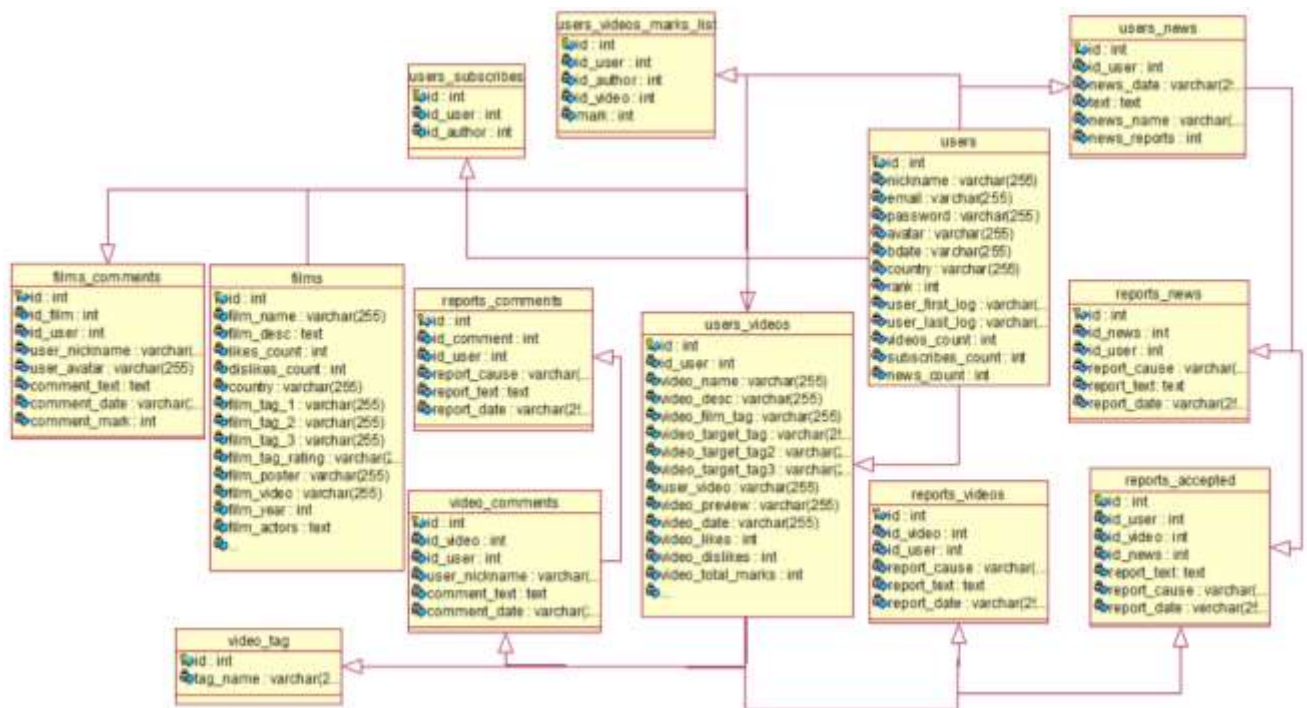


Рисунок 2.2 – Фізична модель бази даних

Основними типами даних, які використовуються для зберігання є int, що використовується для ідентифікаторів і числових значень, text – для текстових даних великої довжини та varchar(255) для текстових даних меншої довжини.

2.3 Проектування інтерфейсу відеохостингу «R.S.Sh.»

Перед безпосередньою розробкою проекту відеохостингу необхідно визначитись з його візуальним оформленням, після чого розробити структуру сторінок інтерфейсу для виділення функціональних можливостей кожної з них.

Як показує практика, за останні роки великої популярності серед розробників у різних сферах набирає темне оформлення інтерфейсу. Це не дивно, адже з появою «темної теми» в операційних системах такий тип оформлення швидко поширився серед користувачів завдяки приємному візуальному вигляду. Для створення інтерфейсу такого стилю набагато легше

проводити комбінації з різних темних відтінків кольорів, ніж зі світлих. Також, їх використання у елементах сайту допоможе користувачам з OLED типом матриць екранів економити рівень заряду батареї та навантаження на пікселі. Варто зазначити і те, що темно-сіре оформлення елементів буде створювати менше дискомфорту для очей людини в порівнянні зі світлими та яскравими кольорами.

Зручність інтерфейсу – один із найвагоміших показників якості сайту. Елементи інтерфейсу повинні бути логічно структуровані для розуміння користувачем та пов'язані між собою. Проектування інтерфейсу потрібно виконувати на основі законів про візуальний дизайн, іконографіки й типографіки. Використовувати актуальні постулати психології поведінки користувача та закономірності інформаційної архітектури та

При розробці веб-інтерфейсів бажано дотримуватися кількох правил:

- брати до уваги можливі цілі та поведінку потенційних користувачів;
- не вирішувати функціоналом сторінок сторонніх завдань;
- не ускладнювати логіку інтерфейсом веб-сайту;
- дотримуватись єдиного стилю інтерфейсу.

Грамотно побудована структура і зрозуміла навігація між сторінками набагато більше спростить життя користувачу, ніж ініціативність. Відвідувач сайту досягне потрібного для себе результату, витрачаючи мінімум зусиль і не отримуючи роздратування. Зручний сайт привабливий для користувача. Чим швидше людина знаходить для себе потрібну інформацію, тим швидше він може стати потенційним клієнтом [16].

У верхній частині кожної сторінки розміщується панель навігації з розділами сайту. Структура сайту буде довільною і він зможе переміщуватись між основними розділами сайту з будь-якої сторінки іншого розділу. Винятком є лише головна сторінка, оскільки вона є сторінкою входу та призначена для реєстрації і авторизації користувачів. При кліку на цій сторінці на будь-який розділ панелі буде відкриватись модальне вікно з формою авторизації та

реєстрації. Більш детально розглянути схему структури сайту сторінок та функціоналу можна на рисунку 2.3.

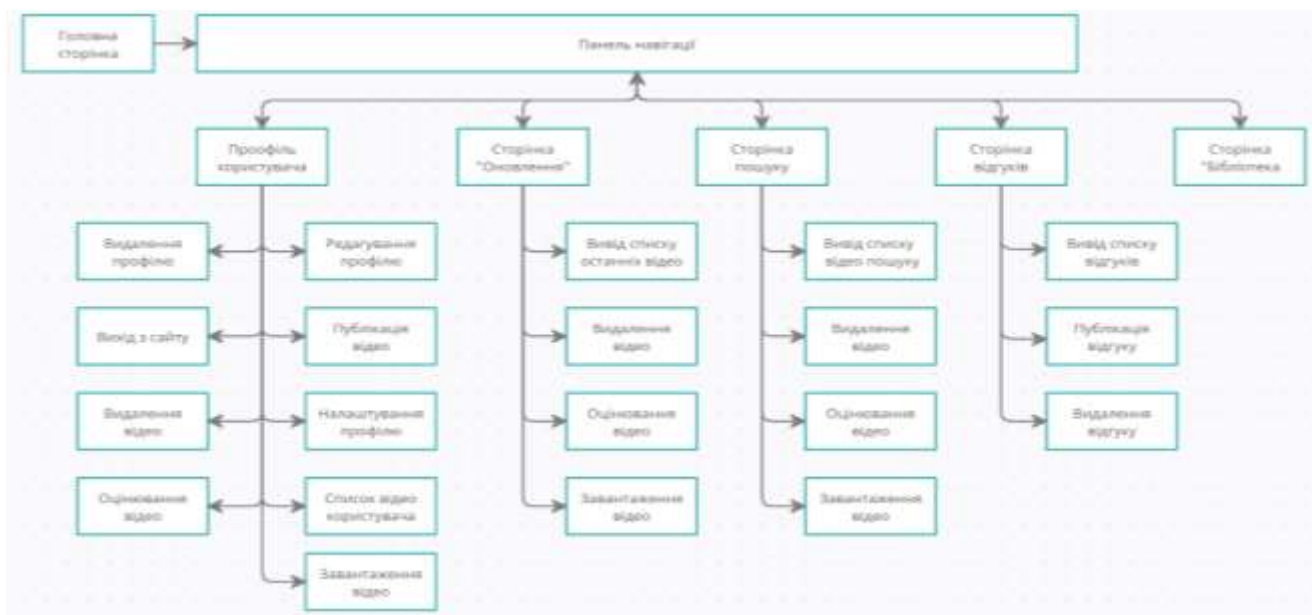


Рисунок 2.3 – Структура шаблону веб-сайту відеохостингу

Завдяки поданій схемі та раніше створеним діаграмам розробник має чітке поняття про вигляд сайту, його функціонал та можливості, що значно спростить реалізацію проекту.

2.4 Практична реалізація та тестування проекту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру

Після побудови всіх необхідних діаграм, підготовки програмного забезпечення та середовища розробки можна починати сам процес створення програмних компонентів та конструювання веб-сайту.

2.4.1 Реалізація програмної частини проекту

Програмна реалізація основних структурних елементів – один з найважливіших етапів розробки будь-якого веб-сайту, оскільки без

налагодженого виконавчого коду всі раніше створені схеми та діаграми не мають ніякого сенсу. Почнемо програмне відтворення елементів та функціональних модулів сайту з індексної сторінки.

Головна сторінка сайту – перша сторінка, яку бачить користувач. Від того, як вона оформлена, наскільки просто користувачеві орієнтуватися, чи є необхідна інформація – від цього залежить, чи залишиться клієнт на сайті або піде. Всі ресурси індивідуальні, але існують загальні принципи того, як повинна виглядати головна сторінка сайту [17].

Перше правило, яким ми скористаємось – це не навантажувати сторінку зайвою інформацією. Підбір даних для виводу виконується на основі психологічного аналізу: оскільки кожен користувач відвідує сервіс заради реалізації чи використання певних можливостей, необхідно в короткій формі описати, що може надати сервіс «R.S.Sh.» персонально йому, як це зображено на рисунку 2.4.



Рисунок 2.4 – Колаж головної сторінки сайту

На основі цього ж правила утримуємось від опису власників чи розробників відеохостингу прямо на сторінці. Цю інформацію, в короткій

формі, виносимо в окреме модальне вікно, яке користувач, при бажанні, зможе відкрити та ознайомитись з вмістом через розділ «Про нас» панелі навігації, як це зображено на рисунку 2.5.

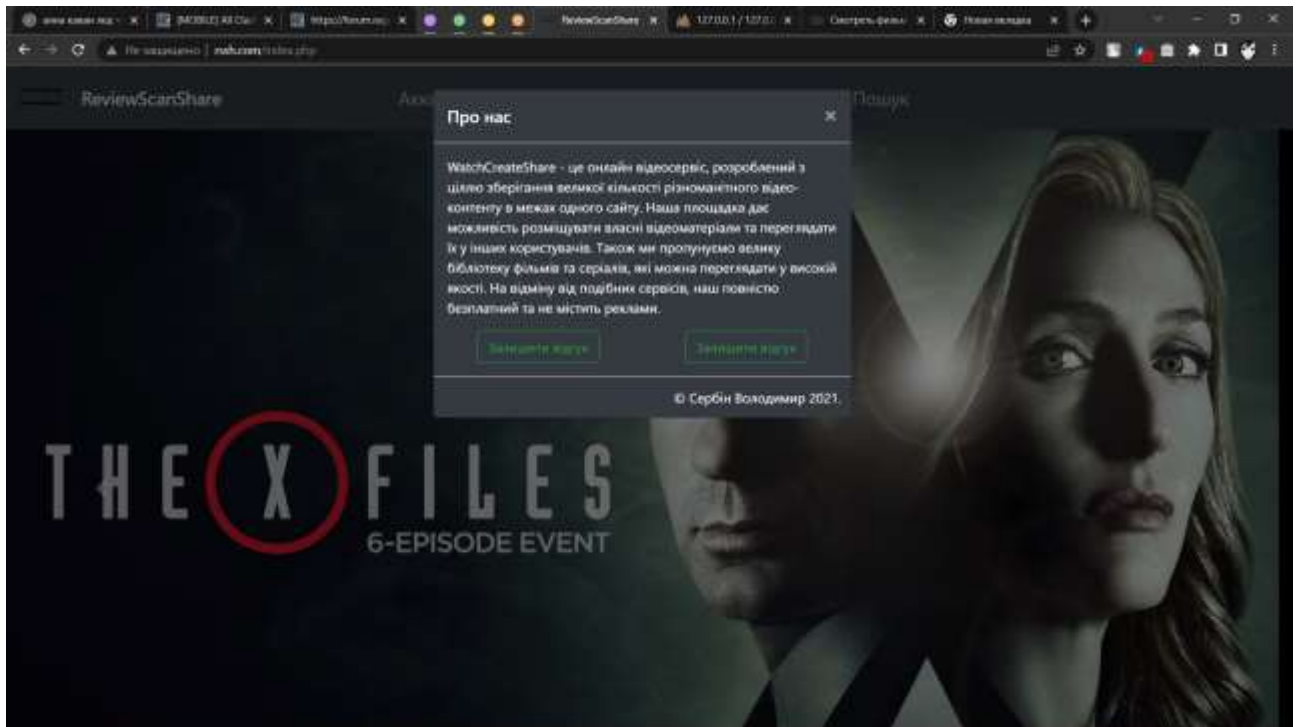


Рисунок 2.5 – Вміст модального вікна «Про нас»

При проектуванні структурного шаблону сайту було зазначено, що для взаємодії з основним функціоналом сервісу відвідувачу необхідний власний профіль користувача. Відповідно, на головній сторінці необхідно реалізувати форму авторизації та реєстрації.

Для цього через кнопку «Профіль» панелі навігації реалізовуємо, що при натисканні, під час знаходження на індексній сторінці, буде відкриватись модальне вікно з формами авторизації та реєстрації. Аналогічну дію повторюємо для інших розділів навігації. Програмний код реалізації зображено у лістингу 2.1.

Лістинг 2.1 – Фрагмент коду панелі навігації header.php:

```
<ul class="navbar-nav mr-auto">
<li class="nav-item active nav-fs" style="padding-right: 170px;">
```

```

<a href="index.php" class="nav-link">ReviewScanShare</a></li>
<li class="nav-item nav-fs" style="padding-left: 25px;">
<a href="#" class="nav-link" data-toggle="modal" data-
target="#AccountModal">Акаунт</a></li>
<li class="nav-item nav-fs" style="padding-left: 25px;">
<a href="#" class="nav-link" data-toggle="modal" data-
target="#AccountModal">Бібліотека</a></li>
<li class="nav-item nav-fs" style="padding-left: 25px;">
<a href="#" class="nav-link" data-toggle="modal" data-
target="#AccountModal">Оновлення</a>
</li>
<li class="nav-item nav-fs" style="padding-left: 25px;">
<a href="#" class="nav-link" data-toggle="modal" data-
target="#AboutUsModal">Про нас</a></li>
<li class="nav-item nav-fs" style="padding-left: 25px;">
<a href="#" class="nav-link" data-toggle="modal" data-
target="#AccountModal">Пошук</a></li></ul>

```

Формі авторизації присвоюємо два поля вводу даних input та кнопку відправки запиту. Форма реєстрації, в свою чергу, містить поля вводу всіх необхідних персональних даних користувача для збереження його профілю та кнопку відправки запиту (див. рисунок 2.6).

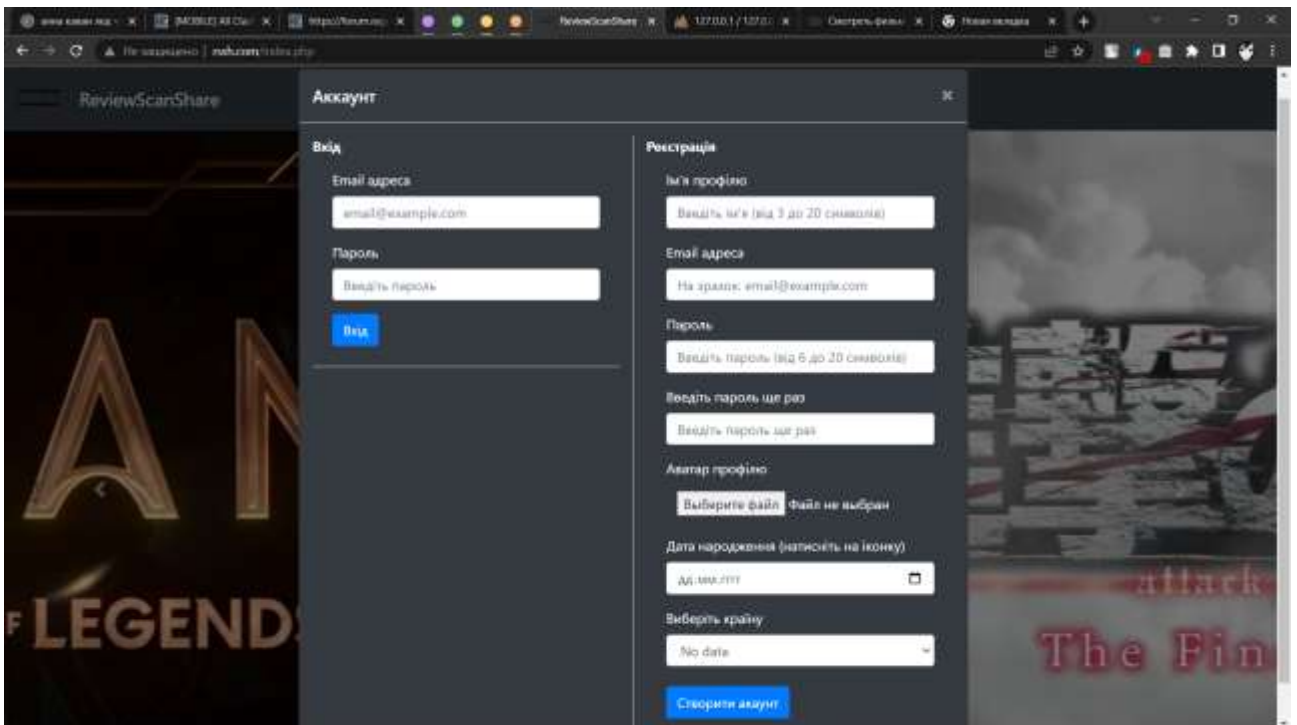


Рисунок 2.6 – Форма авторизації та реєстрації

В обох випадках, після заповнення даних та підтвердження дії, вся інформація надсилається відповідному обробнику для подальшої перевірки.

Для відправки даних обробникам використовуємо html-тег form. Форма призначена для обміну даними між користувачем та сервером. Область застосування форм не обмежена надсиланням даних на сервер, за допомогою клієнтських скриптів можна отримати доступ до будь-якого елементу форми, змінювати його та застосовувати на свій розсуд [18].

При реєстрації всі дані відправляються в файл singup.php, повний програмний код якого наведено в додатку А, де змінним присвоюються значення відправлених полів. Для всіх подальших задач обробника в плані використання найкраще підійде оператор if else. Конструкція if else є однією з найважливіших у практично всіх мовах програмування. Вона дає можливість згідно конкретної умови виконати потрібний фрагмент коду, як продемонстровано у лістингу 2.2 [19].

Лістинг 2.2 –– Перевірка відповідності паролів:

```
$regnickname = $_POST['regnickname'];//оголошення змінних полів
$regemail = $_POST['regemail'];
$regpassword = $_POST['regpassword'];
$confregpassword = $_POST['confregpassword'];
$regbdate = $_POST['regbdate'];
$regcountry = $_POST['regcountry'];
$regdate = date('Y-m-d H:i:s');
if ($regpassword === $confregpassword) {
    //продовження виконання коду
} else {
    $_SESSION['message'] = "<span style='color:red;font-weight:bold;'>Паролі не співпадають</span>";
    header('Location: ../index.php');
}
```

Згідно представленого коду спочатку оголошуються змінні, в які зберігаються дані, отримані з полів форми методом POST. Далі відбувається сама перевірка рядків з паролями. Якщо, в результаті перевірки, паролі не співпадатимуть, користувачу буде переадресовано та повернено відповідь про помилку вводу на головній сторінці.

У PHP регулярні вирази використовуються для синтаксичного аналізу тексту відповідно до визначеного шаблону. Їх використання зробить перевірку тексту на відеохостингу «R.S.Sh.» набагато швидшою та дозволить легше задавати умови вводу [21], тому далі виконується перевірка відповідності введених в поля даних форматам вводу з допомогою цих спеціальних шаблонів пошуку підрядків в тексті, що описано у лістингу 2.3 [20].

Лістинг 2.3 – Перевірка імені та аватара на :

```
if (preg_match("/^[\\w_-]{3,20}$/", $regnickname) &&
preg_match("/^[\\w_\\-\\.]+@[a-z]+[.]+[a-z]/", $regemail)) {
//виконання коду
} else {
    $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Введені в поля дані не відповідають вказаним
форматам!</span>";
    header('Location: ../index.php');
}
```

Для уникнення повторюваних адрес електронної пошти та імен користувачів створюється конструкція з запитом до БД для отримання всіх полів користувачів. Оператор if, в який буде обернено весь інший код, проводитиме їх порівняння з введеними в форму даними та, у випадку виявлення збігів, повертатиме відповідне повідомлення, а в іншому виконуватиме вищеописані частини коду.

В кінцевому результаті, якщо всі умови виконані, код надсилає БД запит на збереження в полях користувача нового запису та зберігає фото у папці серверу. Для з'єднання з БД використовується файл connect.php, який запитується усіма файлами з допомогою конструкції «require_once 'connect.php';» та містить у собі код, що наведено в додатку Б.

Для форми входу обробник виконаний простіше. Він, як і в кодї реєстрації, зберігає отримані дані в змінні та проводить запит до БД для їх порівняння. Якщо для введених логіну та паролю не знайдено збігів серед рядків користувачів, відвідувачу буде повернено відповідне повідомлення. Якщо ж поле з аналогічними даними знайдено, в ньому буде оновлено дату та

час входу на поточні і здійснено перехід на сторінку профілю зі збереженням сесії. Код обробника файлу `singin.php` подано у додатку В.

Після переадресації на сторінку профілю користувачу буде відкритий доступ до розділів панелі навігації. На самій сторінці розміщуватимуться завантажена фотографія користувача та фон на її основі, ім'я, дата останньої авторизації та три функціональні кнопки: «редагувати аккаунт», «завантажити відео» та «ваші дії».

На сторінці профілю кліком на кнопку «Редагувати аккаунт» викликається модальне вікно, у якому є можливість відредагувати збережені дані користувача, або змінити фотографію профілю, як показано на рисунку 2.7.

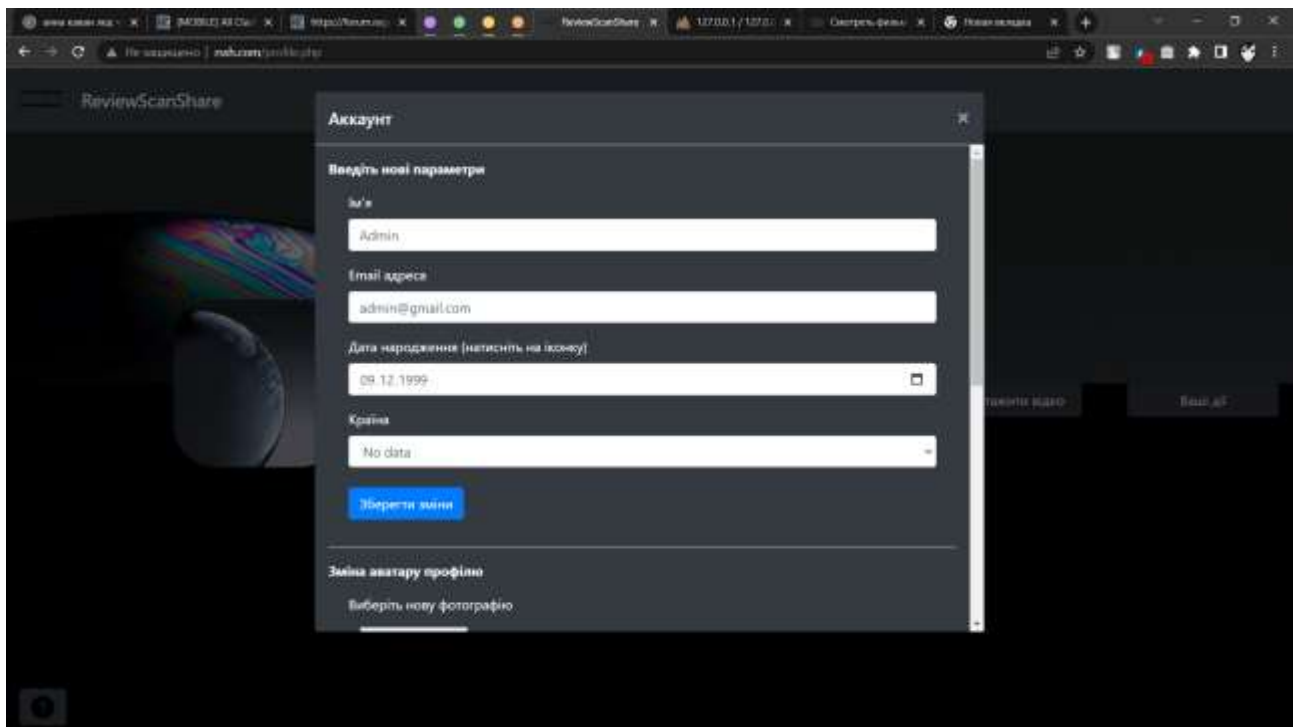


Рисунок 2.7 – Вікно редагування профілю

Продемонстровані, на рисунку вище, поля форми при відправці нових даних обробляються з допомогою файлу `edituserdata.php`, програмний код якого наведено у додатку Г, в той час, як фото та пароль знаходяться в інших формах і проходять обробку з допомогою файлу `editavatar.php` та `edithpass.php`, код та умови яких практично аналогічні коду реєстрації, однак, при запиті до БД

замість конструкції вибірки SELECT виконується конструкція оновлення UPDATE по відношенню до поля даних поточного користувача, як показано у лістингу 2.4.

Лістинг 2.4 – Код оновлення даних в БД:

```
mysql_query($connect, "UPDATE `users` SET `nickname`=
`$editnickname`, `email`= '$editemail', `bdate`= '$editbdate',
`country`= '$editcountry' WHERE `id`= {$_SESSION['user']['id']}");

$_SESSION['message'] = "<span style='color:green;font-
weight:bold;'>Дані успішно оновлено!</span>";
header('Location: ../profile.php');
```

У цьому ж модальному вікні у користувача є можливість повністю видалити свій аккаунт з БД сервісу (див. рисунок 2.8).

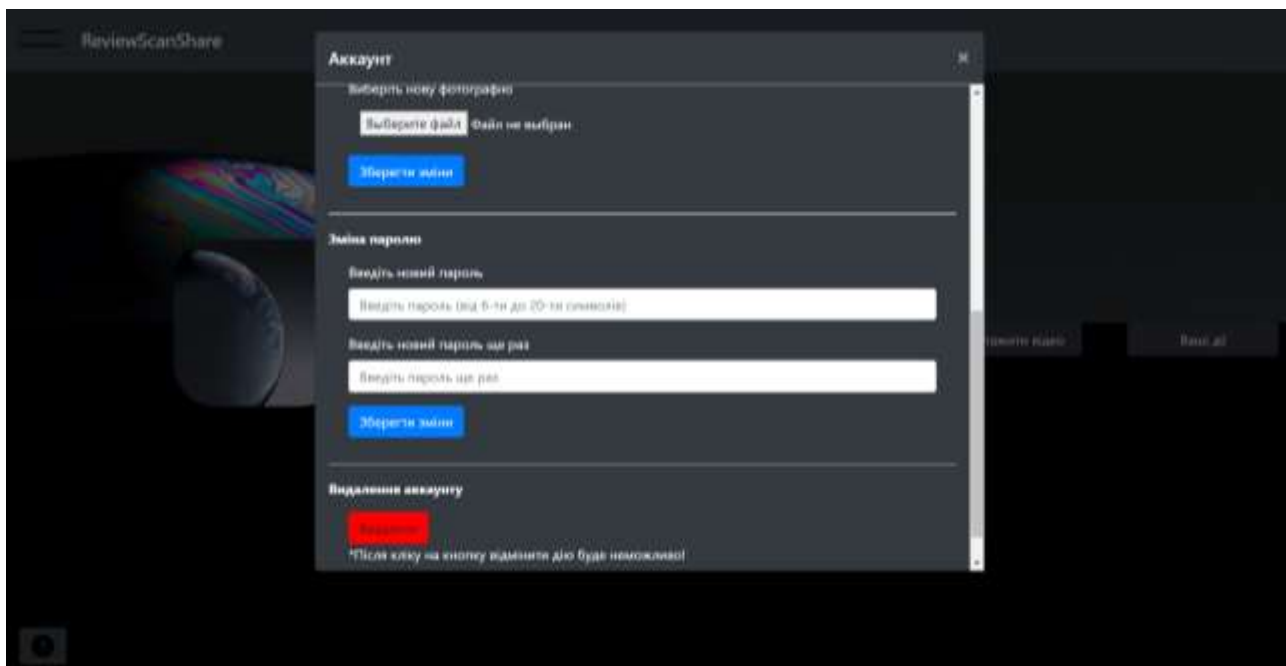


Рисунок 2.8 – Вікно редагування профілю

Після натиску на кнопку «Видалити» через обробник `accdelete.php`, програмний код якого наведено у додатку Д, до БД надсилається серія запитів на видалення всіх, пов'язаних з профілем, даних з усіх таблиць. По завершенню

видалення знищується поточна сесія користувача та відбувається перенаправлення його на індексну сторінку.

Кнопка «Завантажити відео» викликає модальне вікно, всередині якого є дві форми для вибору медіафайлів: відео та превью; декілька текстових форм: назва відео, опис відео (див. рисунок 2.9).



Рисунок 2.9 – Форма завантаження відео

Також, у формі є декілька випадаючих списків: цільовий фільм, ключові теги, вікова категорія.

Щоб запобігти використанню великої кількості однакових символів в ряд в якості опису для відео, проводиться перевірка довжини кожного слова тексту з допомогою конструкції `foreach`, яка надає простий спосіб перебору масивів, приклад якого зображено у фрагменті коду лістингу 2.5.

Лістинг 2.5 – Перевірка рядків:

```
$success = true;
foreach ($array as $word) {
    if(strlen($word) > 45) {
        $success = false;
        $_SESSION['message'] = "<span style='color:red;font-weight:bold;'>В звичайній розмовній лексиці слів довше 45-ти символів не буває! :)</span>";
        header('Location: ../profile.php');}}}
```

```
//виконання коду обробки відео}
```

В описаному кодї створена змінна success, значення якої стандартно є істиною. Далі, з допомогою функції if else виконується перевірка довжини слова з отриманого масиву тексту. Якщо якийсь значення буде вище умови, тоді змінна success отримає значення false та вудбудеться переадресація назад.

Foreach працює тільки з масивами та об'єктами, і генеруватиме помилку при спробі використання зі змінними інших типів або неініціалізованими змінними [22]. Повний код файлу описано у додатку Е.

Остання кнопка «Ваші дії» відкриває модальне вікно з широким інструментарієм для контролю профілю, розділеного на 3 частини: системні повідомлення, новини автора та розширені дані профілю.

В першому розділі виводяться повідомлення від адміністраторів та системи про ті чи інші події на аккаунті. Для прикладу, у випадку, коли адміністратор підтверджує скаргу на контент користувача, останній буде отримувати повідомлення про це, як можна бачити на рисунку 2.10.

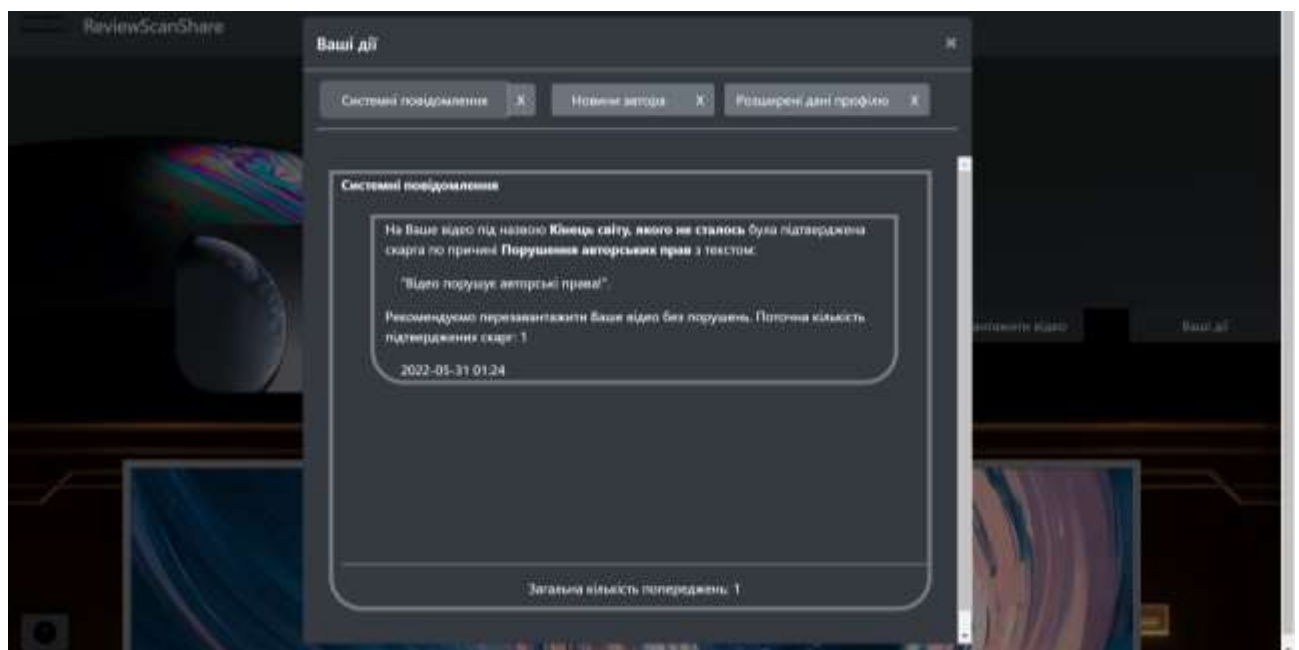


Рисунок 2.10 – Вивід системних повідомлень

Розділ «Новини автора», в основному, призначений для швидкого сповіщення глядачів важливою інформацією зі сторони власника сторінки. Однак, цей інструмент також може використовуватись і для письмового виразу власних думок до аудиторії на певні теми, ведення особистого блогу чи поширення інформації в рамках створюваного контенту.

Для публікації новини достатньо ввести її назву, текст та натиснути відповідну кнопку форми. Після перевірки запиту обробником news.php публікація буде збережена в БД та виведена в розділі для інших користувачів (див. рисунок 2.11).

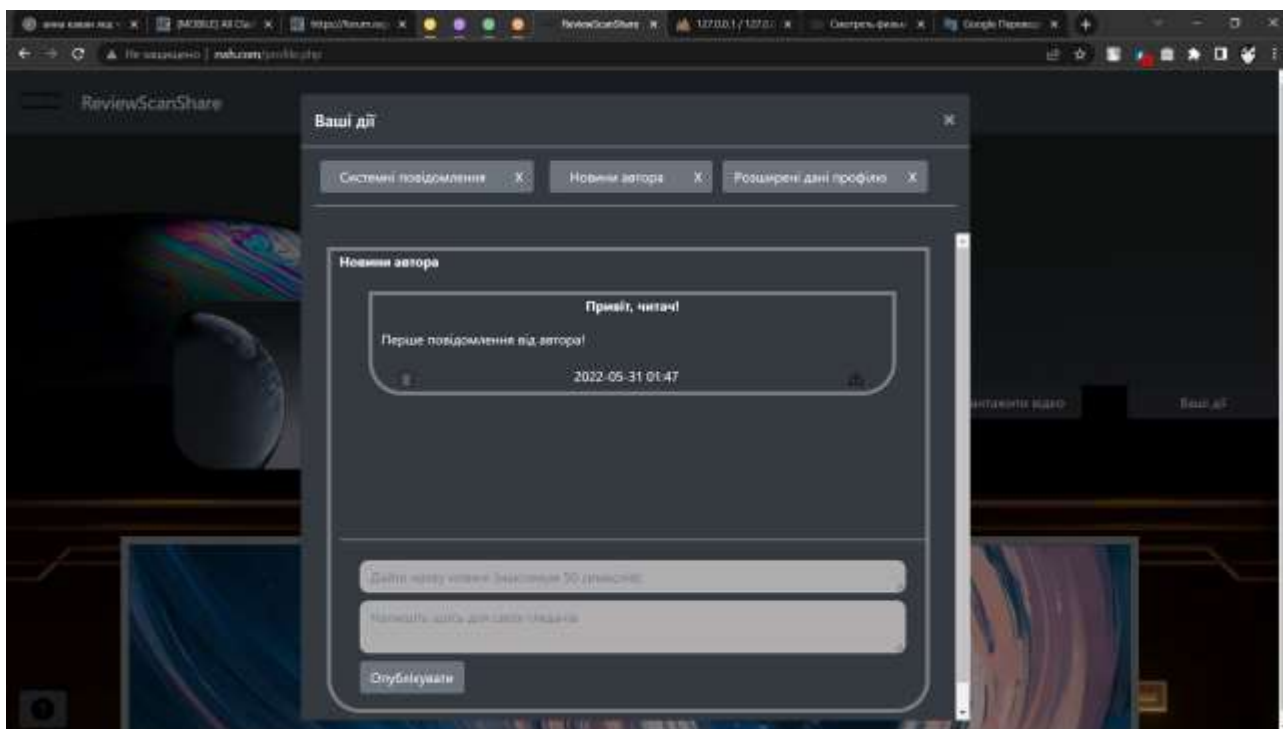


Рисунок 2.11 – Розділ новин автора

Окрім заголовку та тексту новини форма також відображає дату публікації, кнопку видалення та кнопку скарги.

Клік на кнопку скарги ініціалізує виклик модального вікна з випаданим списком, заповненим можливими причинами скарги на публікацію, полем вводу тексту для більш детального опису та кнопкою відправки. В якості обробника всіх типів скарг за форму відповідає файл

reportsend.php, код якого подано у додатку Ж. Методом визначення натиснутої кнопки в коді вибирається варіант зберігання даних ті їх передачі адміністратору. Переглянути вікно скарги можна на рисунку 2.12.

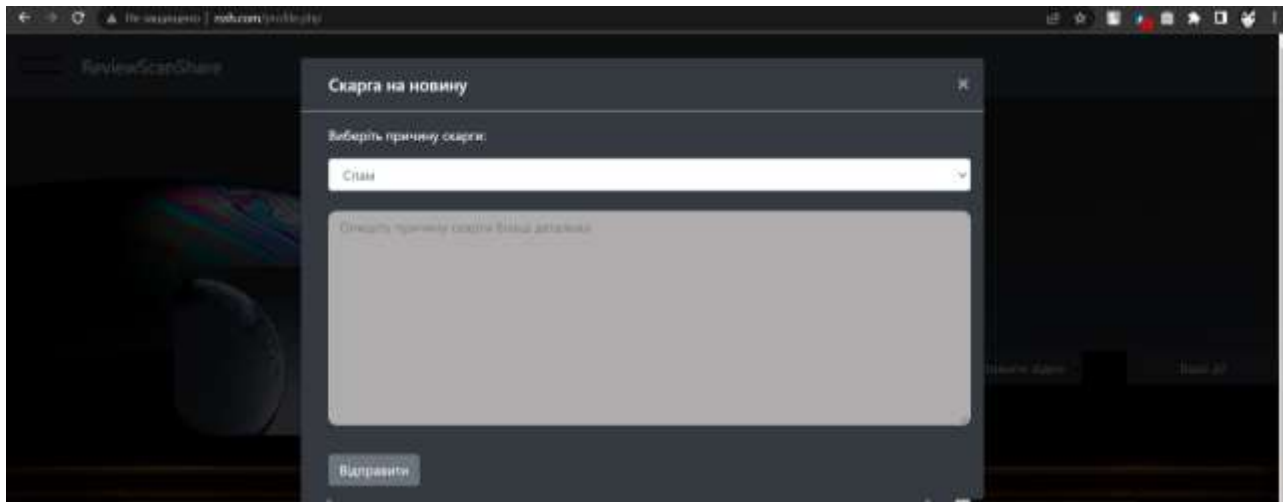


Рисунок 2.12 – Вікно скарги на контент

Останній розділ модального вікна «Ваші дії» призначений для виведення статистики по взаємодії аккаунту з іншими користувачами та контентом сервісу (див. рисунок 2.13).

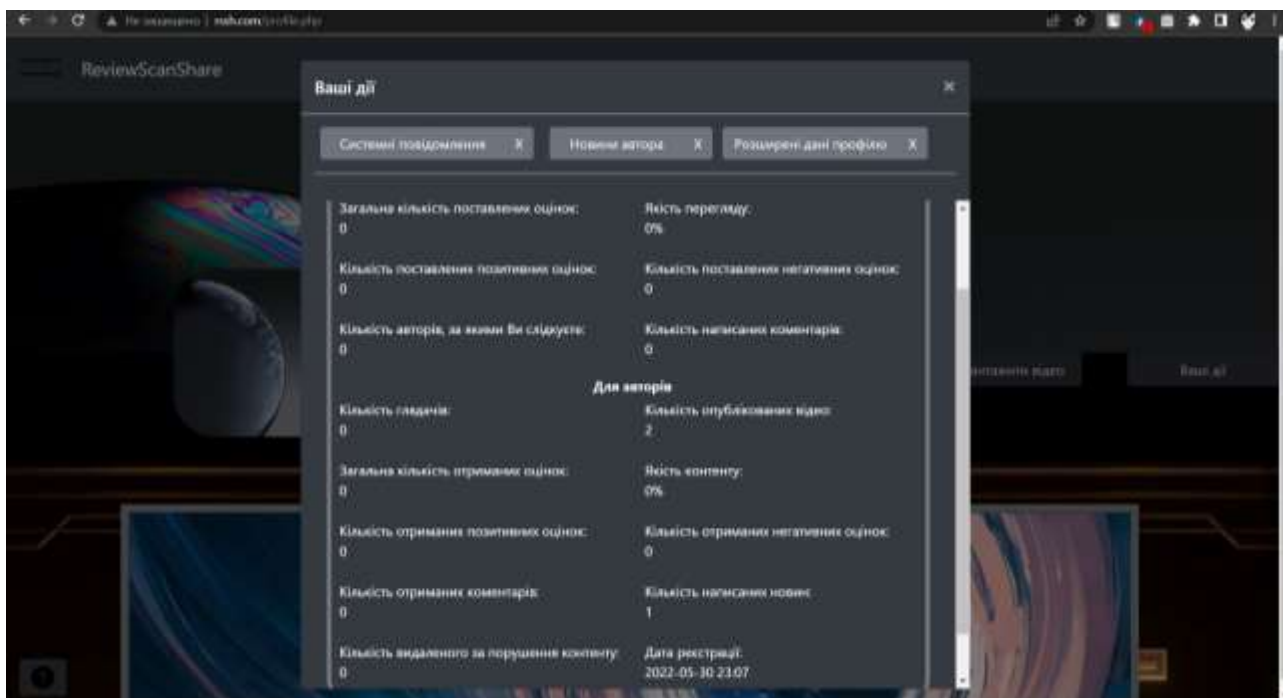


Рисунок 2.13 – Статистика профілю користувача

Інформація першого ряду під назвою «особисте» відображається лише для власника сторінки та показує дії з його сторони.

Другий ряд під назвою «Для авторів», в свою чергу, доступний для всіх користувачів та відображається, як основна інформація розділу «Про автора», за якою люди можуть оцінити загальну якість контенту та свою зацікавленість в його довготривалому перегляді.

Вивід відео на сторінці профілю відбувається нижче загальної інформації та функціональних кнопок, як зображено на рисунку 2.14.



Рисунок 2.14 – Вивід відео у профілі

Оскільки єдиним змінним значенням коду виводу є запит до БД сторінки, для загальної оптимізації сайту було прийнято рішення розмістити його в окремому файлі `uservideodisplay.php` та підключати в кожен файл використовуючи метод підключення `include`: «метод `include` підключає та виконує вказаний файл» [23]. Запити до БД на отримання інформації описуються безпосередньо у коді сторінок. В код обробника, поданого у

додатку И, повернеться вже його результат з повним виводом розмітки відео у порядку від останніх доданих в БД до перших.

В загальному, код працює, як зображено у лістингу 2.6.

Лістинг 2.6 – Запит сторінки та вивід даних:

```
//Запит поточного файлу profile.php до обробника
<!--Контент-->
    <div class="content" id="vidsdiv" style="background-image:
url(img/carousel11.jpg);">
        <div class="row" style="width:100%;">

            <?php

                $result = mysqli_query($connect,"SELECT * FROM
`users_videos` WHERE `id_user` = '$iduser' ORDER BY `id` DESC");

//Включення обробника в код поточного файлу
                include('uservideodisplay.php');
            ?>
        </div>
    </div>
//Обробник отримує запит
session_start();
require_once 'vendor/connect.php';

while($userid = mysqli_fetch_assoc($result)){

    //Вибірка даних з БД та вивід форми відео
}
```

На нижній панелі форми відео виконується вивід його назви, дати публікації, середньої оцінки користувачів у відсотковому співвідношенні, а також розміщуються кнопки оцінки відео, завантаження, відображення детальної інформації, видалення та скарги.

При натиску на кнопку позитивної або негативної оцінки дані з форми передаються до обробника videostats.php, де, в залежності від натиснутої кнопки, відбудеться запит до БД про оновлення інформації в рядку відео. Після виконання в формі детальної інформації зміниться кількість оцінок, а під відео відсоткове співвідношення оцінки. Також зовнішній вигляд натисненої кнопки буде змінено на заповнену іконку, як це зображено на рисунку 2.15.

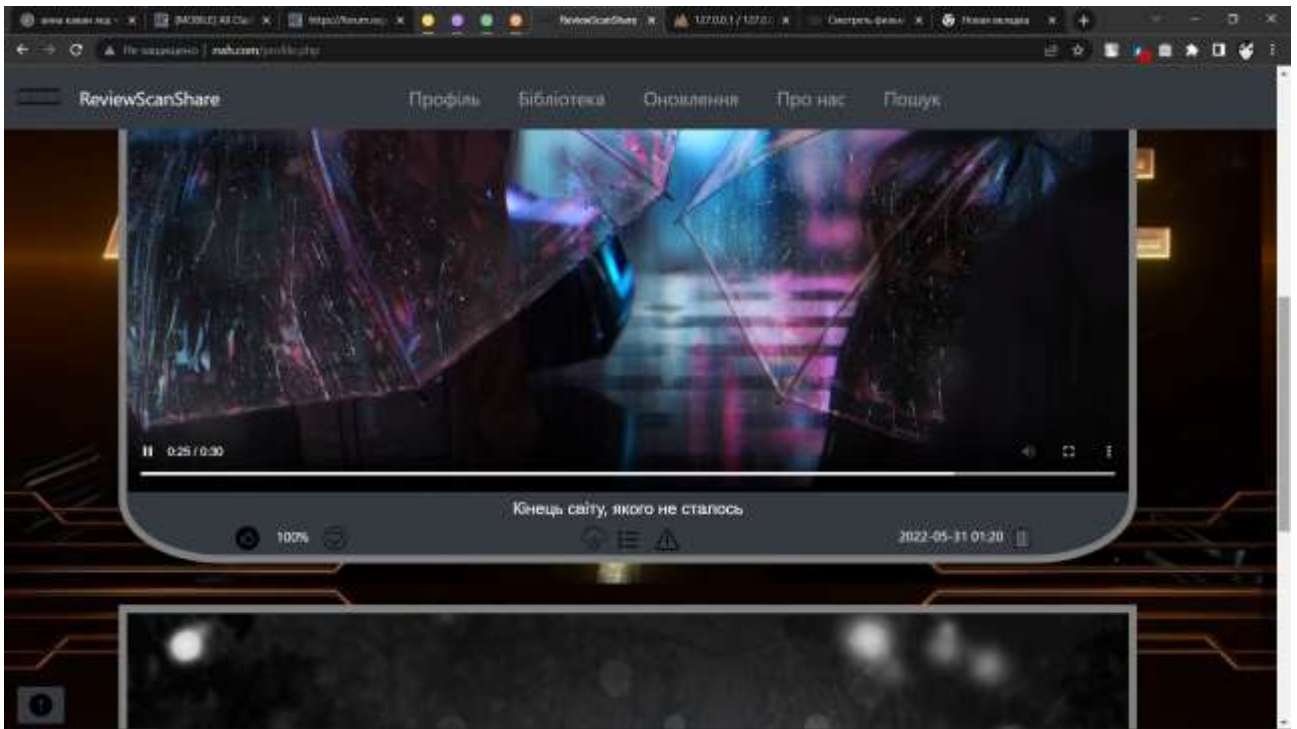


Рисунок 2.15 – Відображення оціненого відео

Після повторного кліку на кнопку оцінки вона, тобто оцінка, буде видалена з БД, а кількість оцінок зменшена.

Кнопка завантаження відео використовує html тег «a», призначений для створення посилань на сторінку чи файл, зі змінною в атрибуті src, яка містить в собі посилання на відео, отримане з БД, як описано у лістингу 2.7 [24].

Лістинг 2.7 – Вивід id відео в атрибуті посилання:

```
<div class="col">
  <div class="px-4 py-1">
    <a download="<?php echo $userid['user_video']; ?>"
href="<?php echo $userid['user_video']; ?>" style="font-
size:20px;font-family:Arial,sans-serif;color:#AEADAD;padding-
left:15%;">
    </a>
  </div>
</div>
```

Всі кнопки видалення та скарги працюють за однаковим принципом. Різниця є лише в типі запиту до БД. Однак, з ціллю обмеження можливості

простих користувачів видаляти опубліковані іншими аккаунтами відео, а також надати таку можливість адміністраторам, було прийнято рішення про використання в коді форми виводу змінної зі значенням «rank» в БД «1» простих користувачів та «2» – адміністраторів. Результат відображення кнопки для груп користувачів зображено на рисунку 2.16.

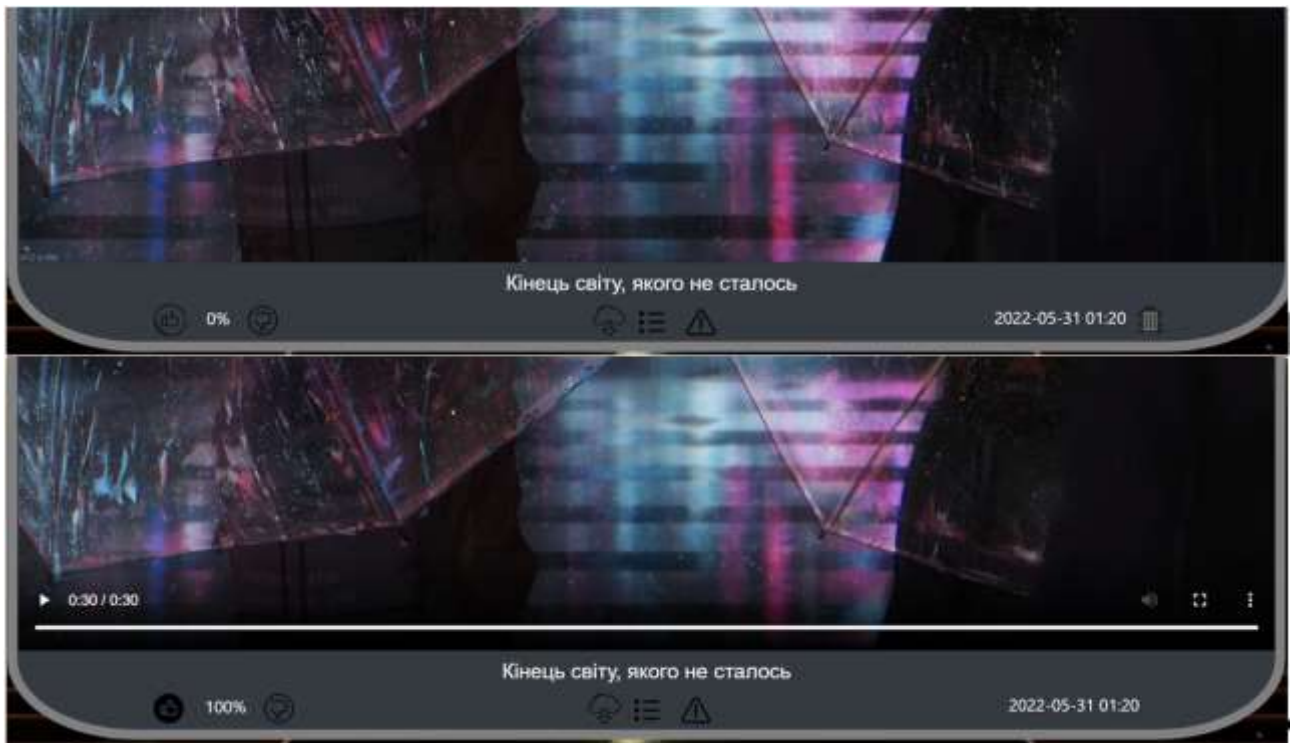


Рисунок 2.16 – Відображення кнопки видалення

Детальна інформація про відео включає в себе наступні дані:

- назву відео;
- опис відео;
- два відео-теги та вікову категорію;
- відношення відео до певного фільму;
- автора опублікованого відео;
- кількість оцінок.

Для виведення використовується javascript функція. Вона викликається натиском на кнопку та змінює параметр видимості блоку інформації. В тому ж блоці розміщена аналогічна кнопка для його приховування (див. рисунок 2.17).

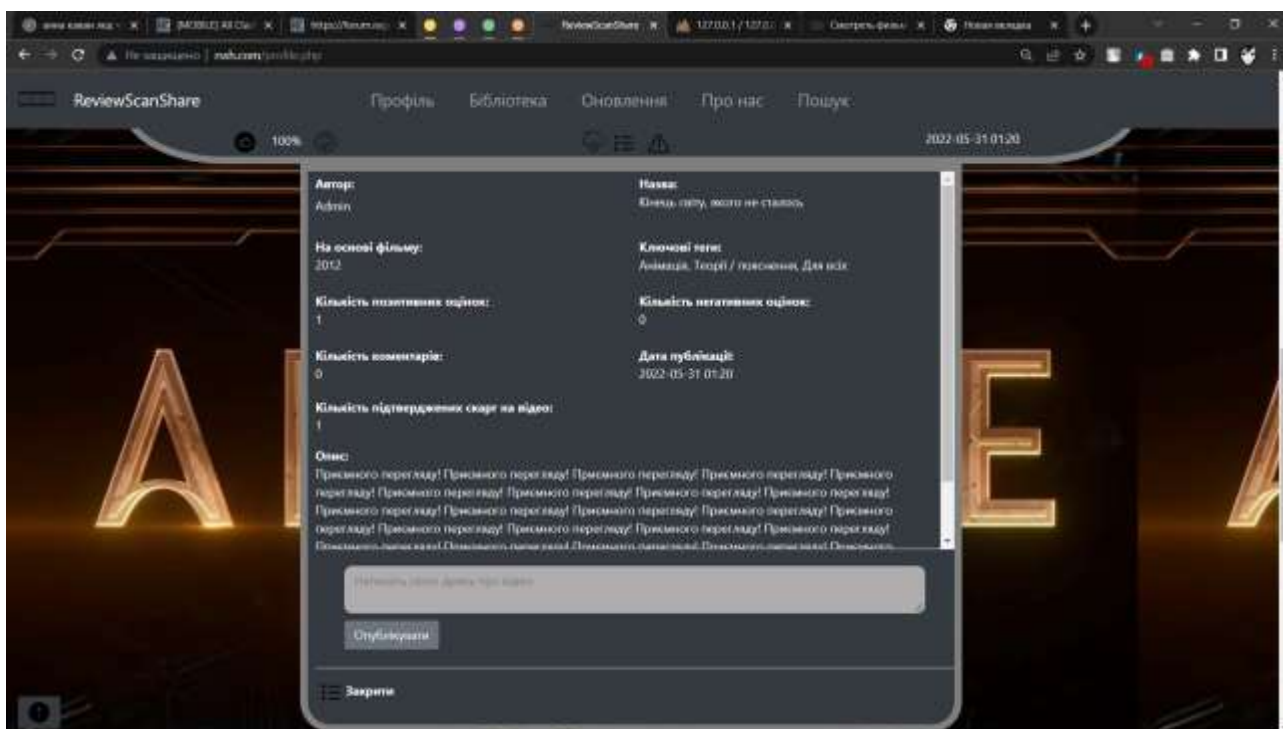


Рисунок 2.17 – Відображення блоку виводу детальної інформації

Вся інформація знаходиться в окремому блоці з можливістю прокрутки вниз. Нижче всього розміщений список коментарів. Кожен коментар виводиться з інформацією про його автора, датою публікації та кнопками видалення, скарги. Переглянути це можна на рисунку 2.18.

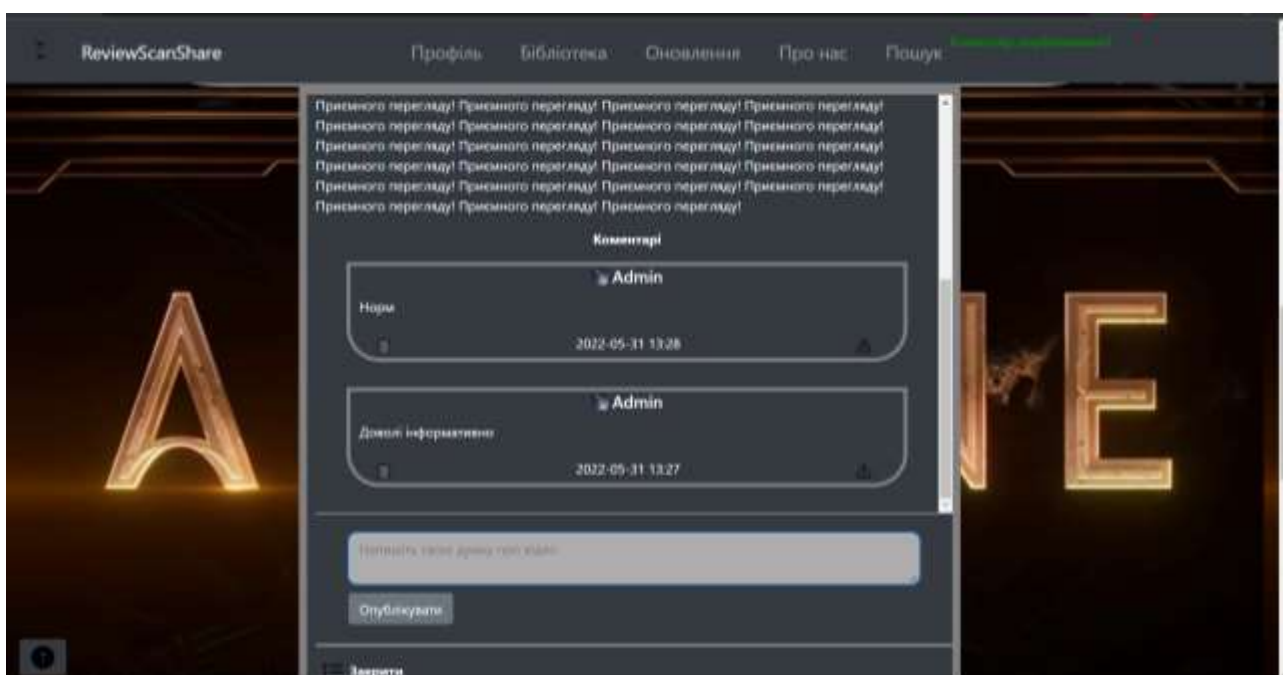


Рисунок 2.18 – Вивід коментарів.

Під блоком виводу інформації знаходиться текстове поле коментування та кнопка публікації, які можна використати для створення власного коментаря.

Повертаючись до панелі навігації слід повторити, що для авторизованого користувача доступні всі розділи, і першим з них є розділ «Профіль». При активації кнопки відбувається вивід модального вікна з короткою інформацією про поточний акаунт користувача: аватар, ім'я, поштова адреса реєстрації, останній вхід, дата реєстрації, дата народження та країна. Нижче цієї інформації знаходяться декілька кнопок: кнопки списків з відео, які сподобались чи не сподобались користувачу та кнопка, яка діє аналогічно кнопці виводу детальної інформації відео та відкриває список профілів, за якими слідкує користувач (див. рисунок 2.19).

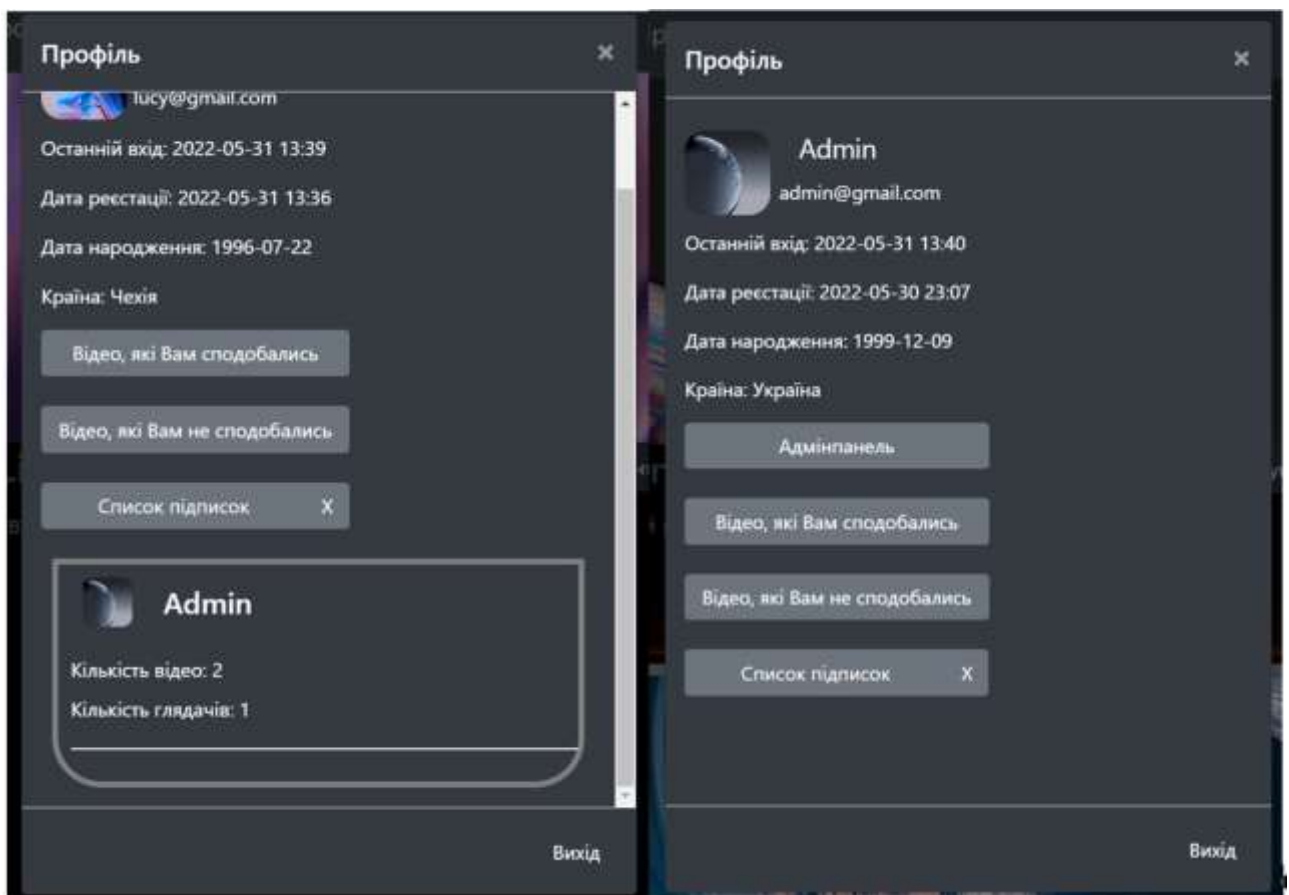


Рисунок 2.19 – Вікно профілю користувачів

Розміщення плейлистів відео та списку підписок в окремому вікні панелі навігації обумовлене тим, що користувачі досить часто переглядають ці розділи

на інших відеохостингах і мати швидкий доступ до них пріоритетно для більшості.

В залежності від рангу користувача, першою може відображатись кнопка адмінпанелі. Її виклик створює в новому вікні браузера сторінку, на якій розміщені інструменти адміністратора, зображені на рисунку 2.20.

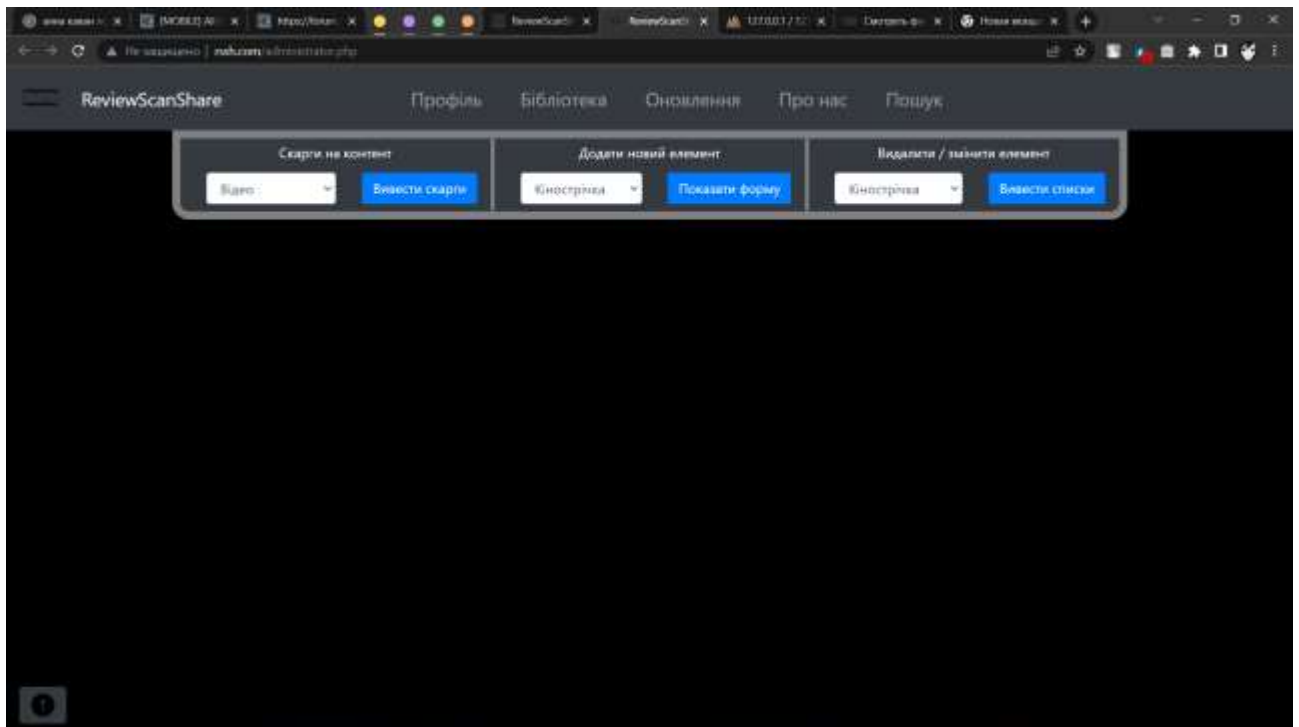


Рисунок 2.20 – Інструменти адміністратора

Інструменти розділені на 3 групи. Перша включає в себе вивід скарг користувачів на відео, новини та коментарі. При виборі необхідного пункту та виклику через відповідну кнопку на екрані будуть відображені всі скарги. У вигляді коду це описано у лістингу 2.8.

Лістинг 2.8 – Відображення форм:

```
<?php
//Якщо натиснута кнопка репортів
if (isset($_POST['btnreportsadm'])) {
    $reptype = $_POST['reportslistadm'];
    switch ($reptype) {
        //Репорти відео
        case 1:
```

```

$vidreportslist = mysqli_query($connect,"SELECT * FROM
`reports_videos` ORDER BY `id`");
while ($vidreports = mysqli_fetch_assoc($vidreportslist)) {
//Дані репорту
} break;
//Репорти новин
case 2:
$newsreportslist = mysqli_query($connect,"SELECT * FROM
`reports_news` ORDER BY `id`");
while ($newsreports = mysqli_fetch_assoc($newsreportslist)) {
//Дані репорту
} break;
//Репорти коментарів відео
case 3:
$commsreportslist = mysqli_query($connect,"SELECT * FROM
`reports_comments` ORDER BY `id`");
while ($commsreports = mysqli_fetch_assoc($commsreportslist)) {
//Дані репорту
} break;
}
} else if (isset($_POST['btnnewelementadm'])) {
switch ($addelements) {
//Якщо натиснута кнопка додавання елемента
}
//Якщо натиснута кнопка додавання елемента
}
} else if (isset($_POST['btndeletelementadm'])) {
$delements = $_POST['delementadm'];
switch ($delements) {
//Якщо натиснута кнопка видалення елемента
}
}
}

```

В загальному, при виводі першим буде відзначено причину скарги зі списку, її порядковий номер, що відповідає ідентифікатору в БД, та дата відправки адміністратору, сам контент, на який поскаржились і дві кнопки дії. В залежності від вибраного пункту форма виводу інформації буде відрізнятись лише об'єктом скарги і кнопкою підтвердження. Оскільки відео та новини – це тип контенту, для видалення якого замало однієї підтверженої скарги, адміністратор може лише підтвердити скаргу для її збереження в таблиці скарг БД та підрахунку в кількості. Якщо ж кількість буде вищою зазначеної в умові збереження, система автоматично видалить відео чи запис.

Форму виводу скарг для перегляду зображено на рисунку 2.21.

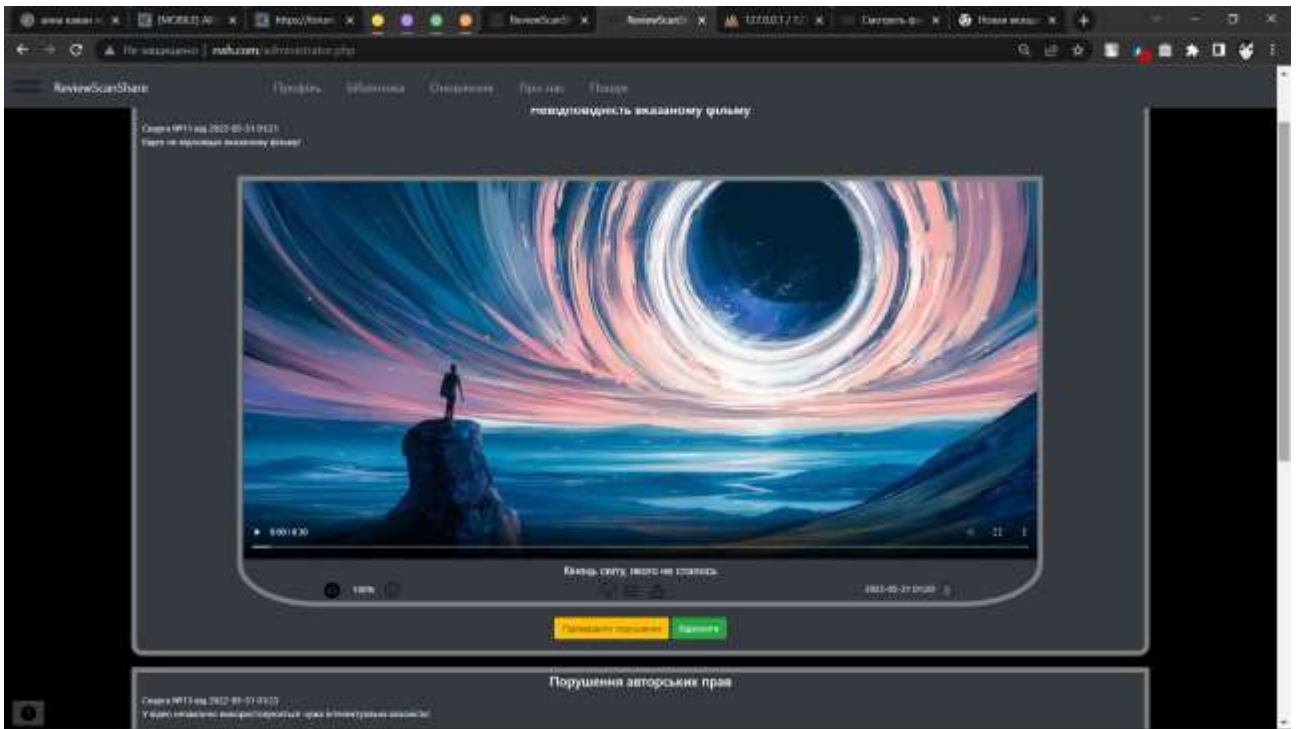


Рисунок 2.21 – Вивід списку скарг на відео

У випадку з коментарями адміністратору відразу надається можливість їх видалення, що зображено на рисунку 2.22.

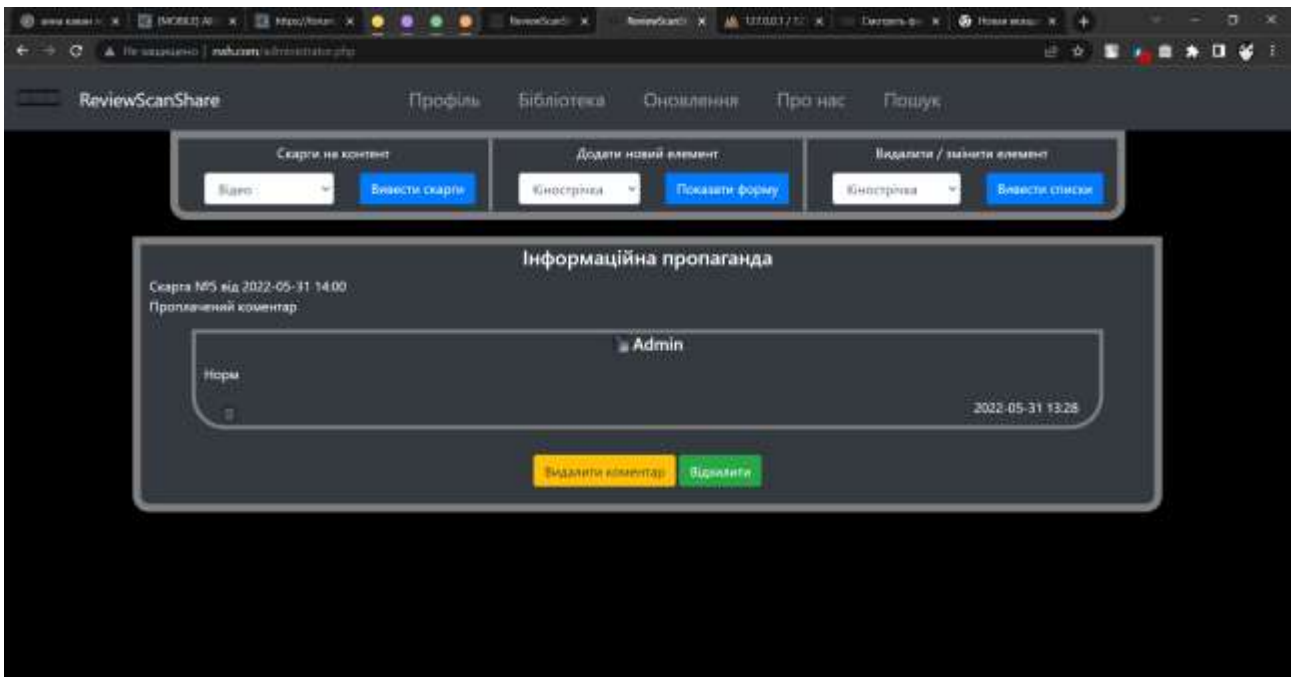


Рисунок 2.22 – Вивід списку скарг на коментарі

В другій групі інструментів адміністратор має можливість додати на сайт новий відео-тег або фільм до бібліотеки.

У першому випадку на сторінці створена одна текстова форма та кнопка відправки даних, як показано на рисунку 2.23.

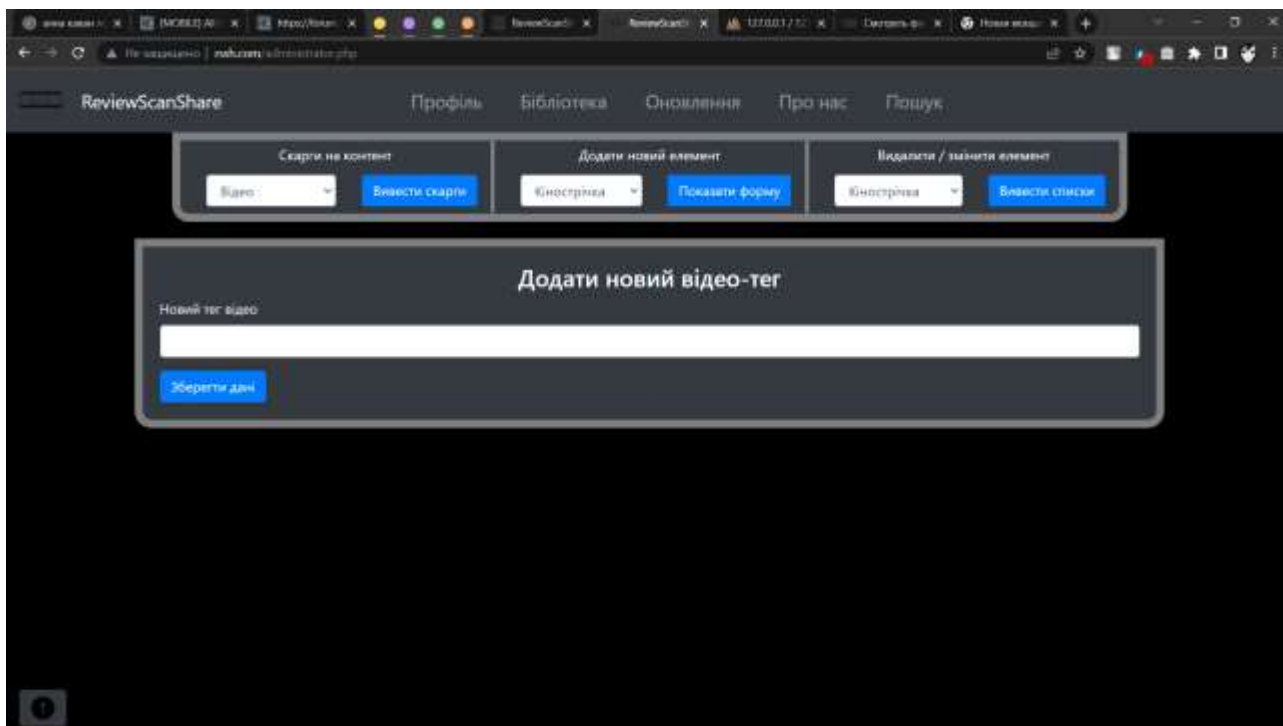


Рисунок 2.23 – Створення відео-тегу

Зберігання відео-тегів в БД через адміністратора обумовлене тим, що на всіх сайтах з можливістю задання тегу користувачами існує велика проблема пошуку контенту за ними. Значна кількість даних зберігається без практичного використання. Серед них чимало «клонів», тобто тегів з одним призначенням, але різними різними на один чи декілька символів назвами, збереженими так через помилку в написанні чи з ціллю відмінності від інших. На додачу, чітко виражений список тегів та обмеження в 2 на відео допоможе авторам контенту створювати його для більш конкретної аудиторії, а користувачам – швидше знаходити те, що їх цікавить.

Додавання нового фільму відбувається в окремому вікні та включає в себе параметри, зображені на рисунку 2.24.

The screenshot shows a web form titled "Додати новий фільм" (Add new film). The form is set against a dark background with white text and input fields. It includes the following elements:

- Назва фільму** (Film title): A text input field.
- Опис до фільму** (Film description): A text area.
- Головні актори** (Main actors): A text input field.
- Країна-виробник** (Country of production): A dropdown menu with "США" (USA) selected.
- Рік створення фільму** (Year of creation): A dropdown menu with "1921" selected.
- Повна дата випуску** (Full release date): A date picker field showing "дд.мм.гггг".
- Основні жанри** (Main genres): Three dropdown menus with "Трилер" (Thriller), "Катастрофа" (Disaster), and "Екшн" (Action) selected.
- Вікова категорія** (Age category): A dropdown menu with "Для всіх" (For all) selected.
- Обкладинка фільму (вертикальна)** (Vertical film cover): A file upload field with the text "Виберіть файл" (Choose file) and "Файл не вибран" (File not selected).
- Відео фільму** (Video film): A file upload field with the text "Виберіть файл" (Choose file) and "Файл не вибран" (File not selected).
- Зберегти дані** (Save data): A blue button at the bottom of the form.

Рисунок 2.24 – Вікно додавання нового фільму

Після заповнення всієї необхідної інформації, вибору відповідних значень випадальних списків, завантаження фото- та відео-файлів необхідно натиснути на кнопку «Зберегти дані». Це викличе дію відправки всієї інформації до обробника `film.php`, який проведе перевірку всіх полів на відповідність поставленим вимогам. Програмний код обробника `film.php` представлений в додатку К та дуже схожий з обробниками, що раніше використовувались.

Після підтвердження виконання всіх умов в ньому дані будуть збережені до таблиці БД, а сам фільм – відображений у бібліотеці.

Третя група інструментів надає можливість видалення фільму та відео-тегу, а також редагування останнього при допущенні помилки, що зображено на рисунку 2.25.

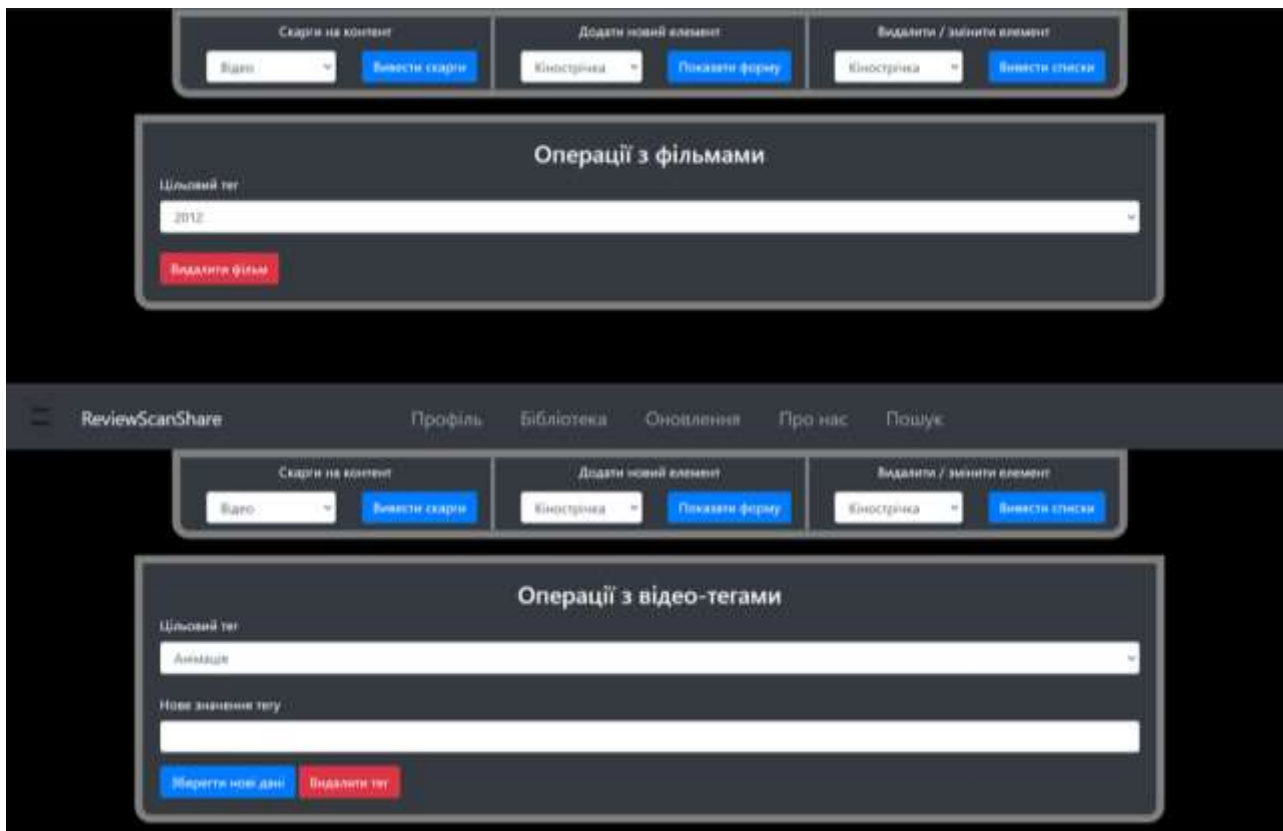


Рисунок 2.25 – Третя група інструментів

Вивід відео-тегів для адміністратора та у вікні завантаження профілю відбувається на основі списку, збереженого в окремій таблиці БД за запитом, описаним у лістингу 2.9.

Лістинг 2.10 – Вивід списку відео-тегів:

```
<div style="border:10px solid gray;border-bottom-left-
radius:20px;border-bottom-right-radius:20px;background-
color:#343A40;color:white;width:80%;margin-left:10%;margin-
top:20px;padding:20px 20px 20px;">
<h3 style="text-align:center;">Операції з відео-тегами</h3>
<label for="tagselect">Цільовий тег</label>
<form action="vendor/film.php" method="post"
enctype="multipart/form-data">
<select name="tagselect" class="form-control" id="tagselect">
<?php
$tagslist = mysqli_query($connect, "SELECT * FROM `video_tags`
ORDER BY `tag_name`");
while ($tagslistd = mysqli_fetch_assoc($tagslist)) {
?>
<option value="<?php echo $tagslistd['id'];?>"><?php echo
$tagslistd['tag_name'];?></option>
<?php
```

```

}??>
</select><br>
<div class="form-group">
<label for="regtagname">Нове значення теґу</label>
<input type="text" class="form-control" name="regtagname"
id="regtagname">
</div>
<input type="submit" class="btn btn-primary" name="btnregvideotag"
value="Зберегти нові дані">
<input type="submit" class="btn btn-danger" name="btndelvideotag"
value="Видалити теґ">
</form>
</div>

```

Наступним розділом панелі навігації є сторінка бібліотеки. При відкритті відбувається вивід списку фільмів від найновіших (див. рисунок 2.26).



Рисунок 2.26 – Вивід списку фільмів

Вище списку результатів знаходиться поле пошуку та кнопка фільтрів, з допомогою якої відкривається модальне вікно налаштувань параметрів пошуку. Ці параметри допоможуть зробити результати більш точними (див. рисунок 2.27).

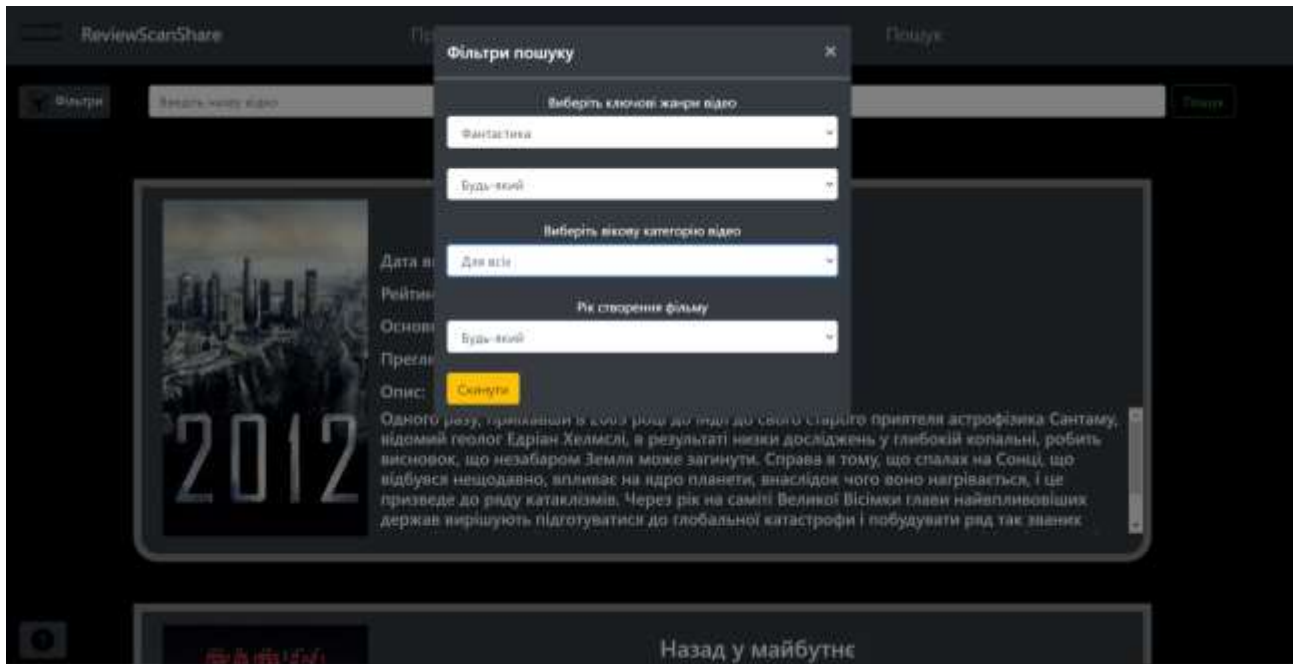


Рисунок 2.27 – Фільтри пошуку

Принцип роботи кнопки фільтрів окремо відображено у додатку Л. Можна додати, що працює код обробки цих параметрів на основі оператора умови if else, який виконує відповідний запит до включеного модулю відео.

Натиснувши на назву фільму в окремій вкладці браузера відкриється його сторінка, зображена на рисунку 2.28.

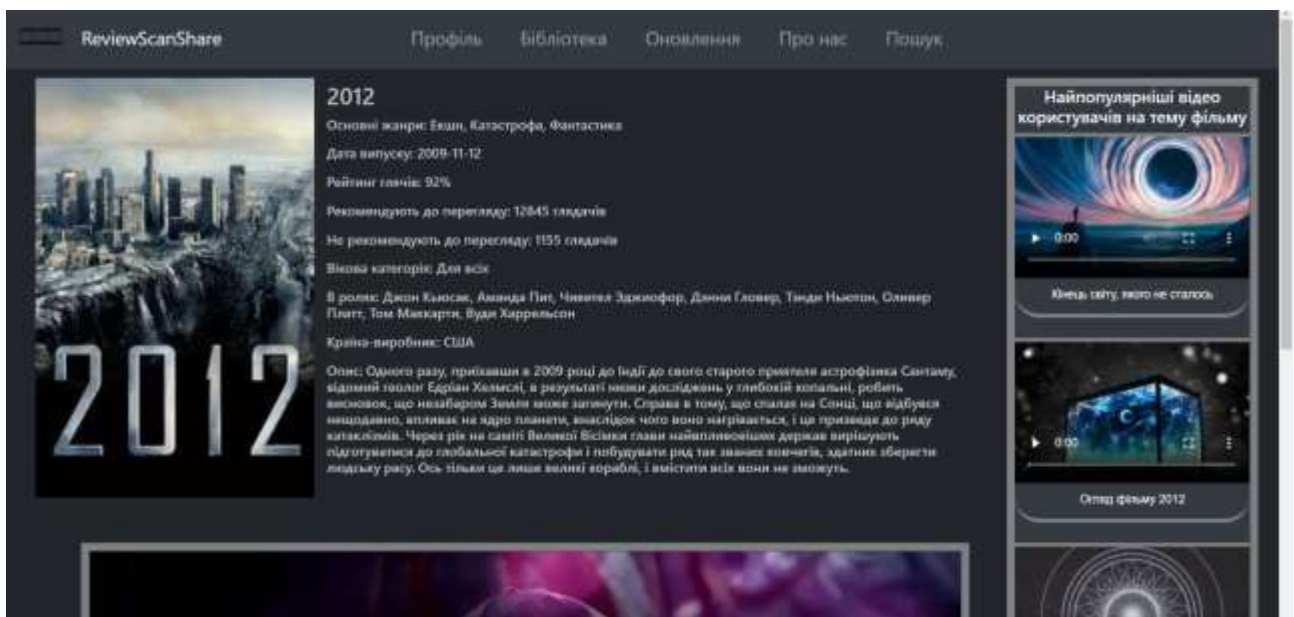


Рисунок 2.28 – Сторінка фільму

Сторінка поділена на дві частини. В першій відбувається відображення постера та детальної інформації про фільм. Нижче знаходиться саме відео фільму та коментарі користувачів до нього (див. рисунок 2.29).

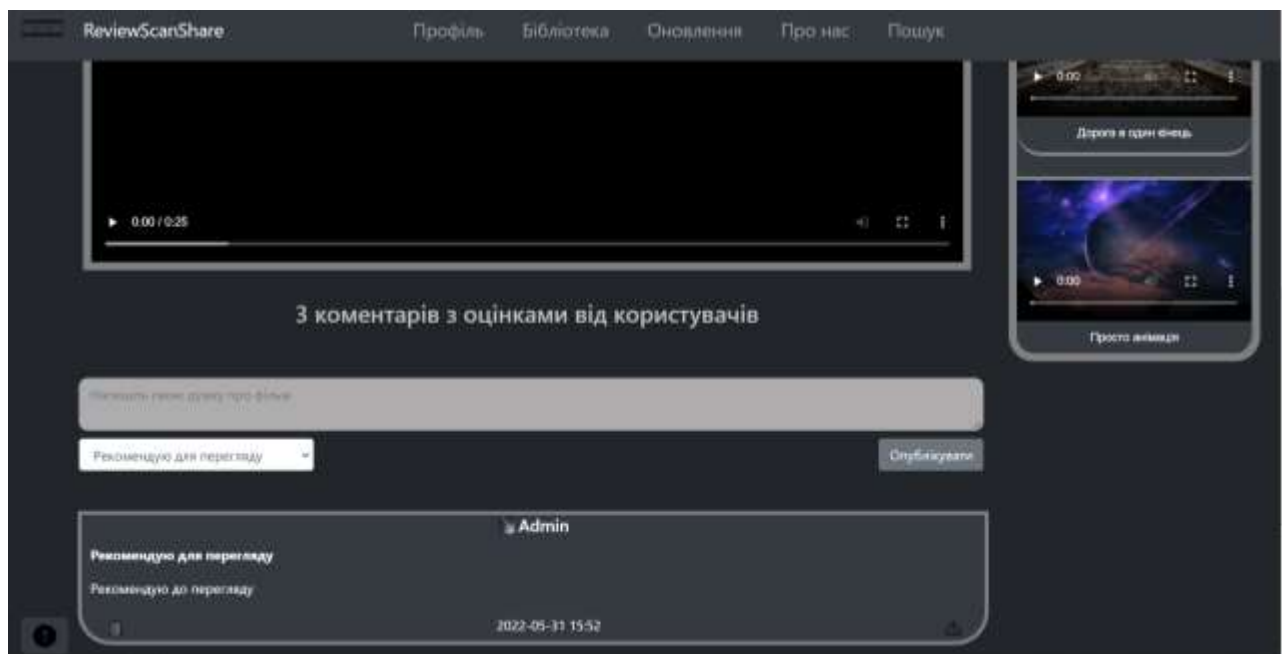


Рисунок 2.29 – Блок коментарів

Перед тим, як опублікувати коментар, кожен користувач у випадяючому списку вибирає один з варіантів: чи рекомендує він фільм до перегляду, чи не рекомендує. Це дозволить користувачам, які читають коментар, зрозуміти оцінку його автора. Також, на основі цих оцінок задоволеності переглядом формується загальний рейтинг фільму, що більш практично, оскільки користувачі можуть самі переконатись в чесності оцінки, виставленої на сайті.

В другій частині розміщено блок, який виводить 5 найпопулярніших відео на тему фільму. Перегляд відео можна здійснити просто натиснувши на відповідні кнопки управління. Клік по назві відео відкриє сторінку оновлень.

Аналогічно «Бібліотеці», сторінка «Оновлення» стандартно виводить найновіші опубліковані авторами відео. Однак, на сторінці також присутня форма фільтрації, яка допоможе вивести більш близький до бажаного контент і без необхідності самостійно щось шукати, що зображено на рисунку 2.30.

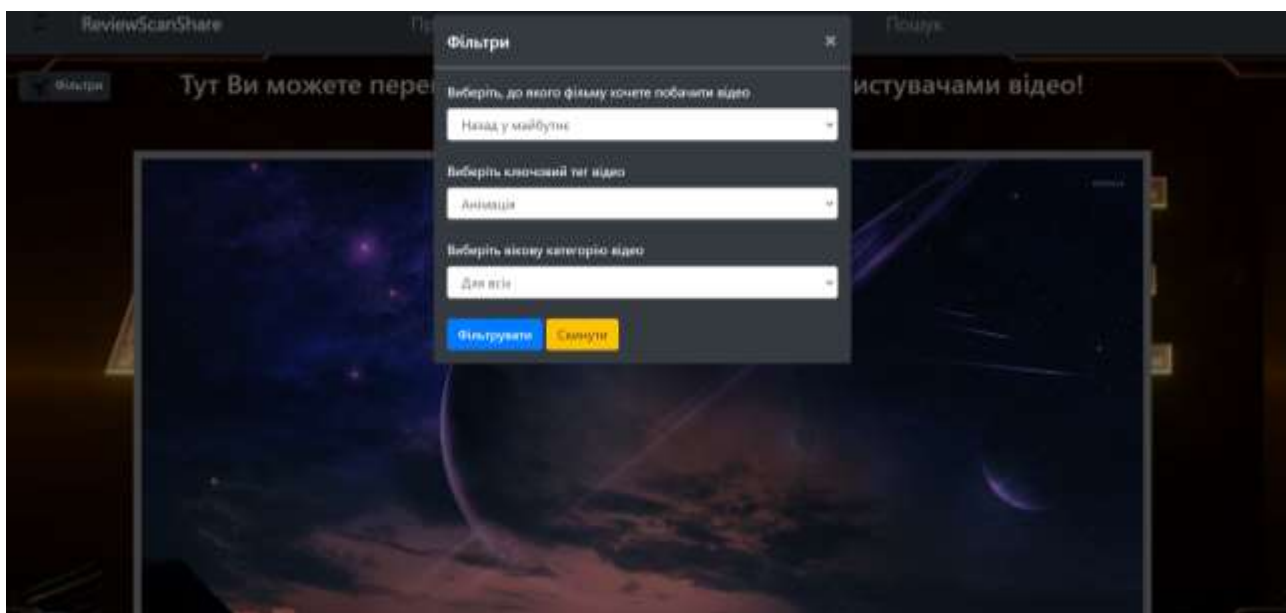


Рисунок 2.30 – Фільтрація виводу відео

Якщо ж користувач хоче знайти конкретне відео, він може скористатись розділом «Пошук», форми якого працюють подібно пошуку на сторінці бібліотеки (див. рисунок 2.31).



Рисунок 2.31 – Сторінка пошуку

Окрім пошуку відео за фільтрами користувач також може здійснити пошук профілю іншого користувача, як це зображено на рисунку 2.32.

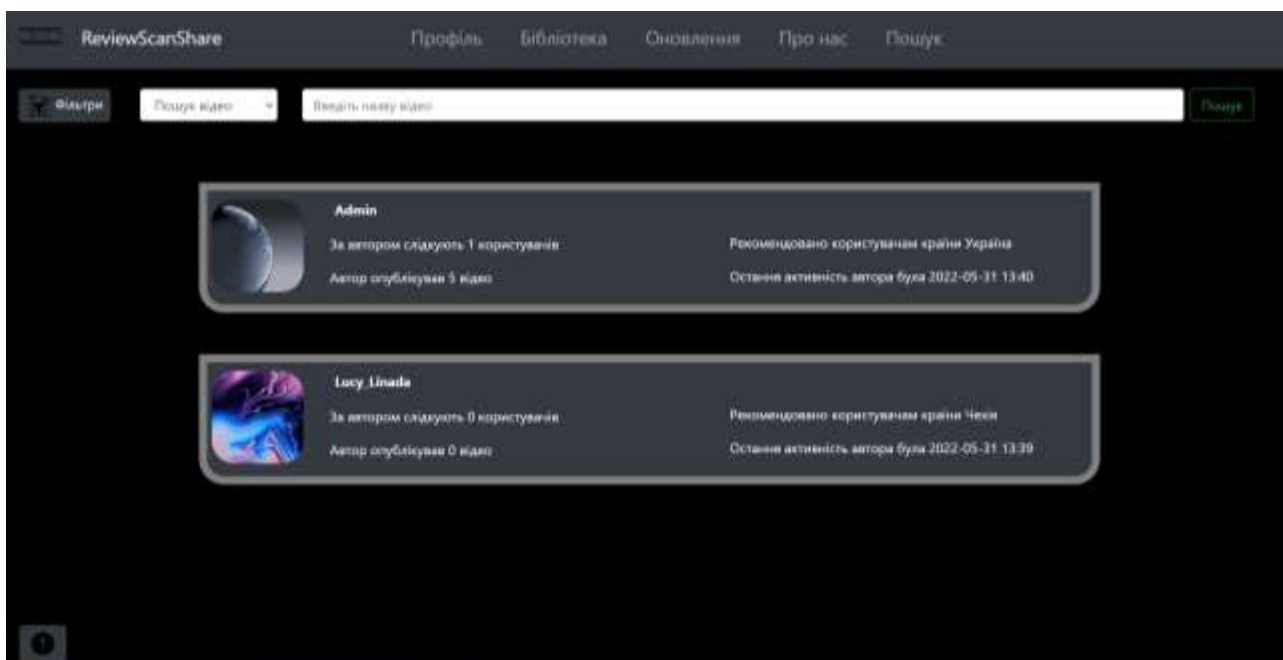


Рисунок 2.32 – Сторінка пошуку

Серед результатів пошуку також можна знайти власний профіль та переглянути його. Про перегляд власного профілю зі сторони гостя буде повідомляти відповідний напис під панеллю навігації, а замість кнопки підписки на автора буде відображено кнопку переходу на сторінку профілю, що зображено на рисунку 2.33.

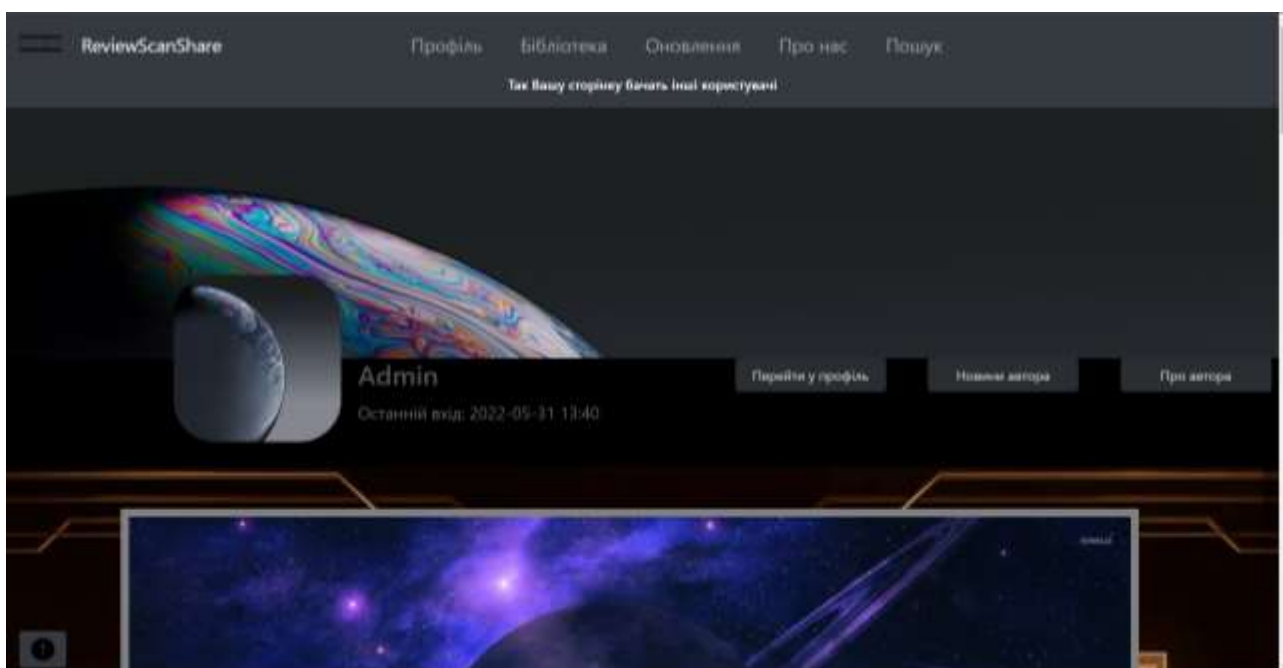


Рисунок 2.33 – Сторінка користувача

Якщо користувач площадки має бажання оцінити сайт чи запропонувати певні покращення, доповнення, він може скористатись кнопкою «Залишити відгук» модального віна «Про нас». Після кліку на кнопку відбудеться перехід на сторінку відгуків, де можна написати та відправити текст відгуку, видалити написаний, а також переглянути інші, як на рисунку 2.34.

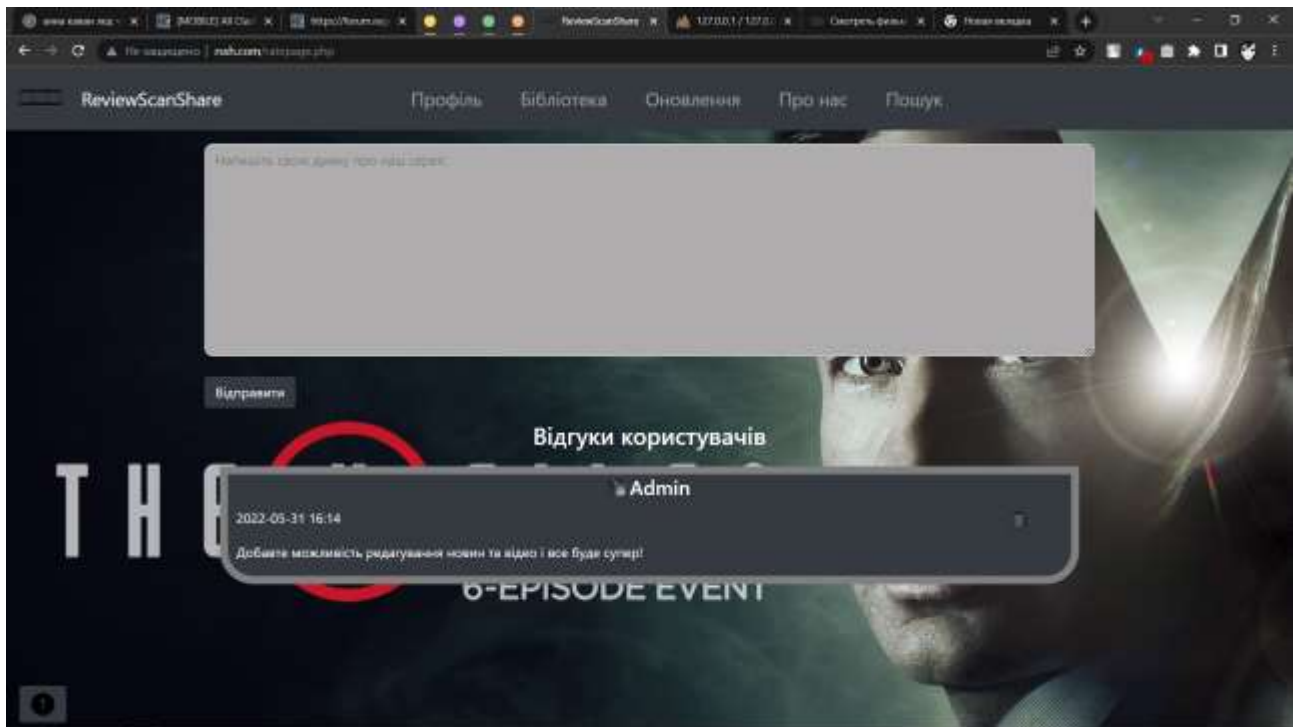


Рисунок 2.34 – Сторінка відгуків

На цій сторінці кожен може переглянути текстові оцінки сайту від його користувачів та зробити власні висновки щодо його якості.

2.4.2 Розгортання та тестування інфраструктури відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру

Після завершення розробки візуальної та функціональної частини коду веб-сайту відеохостингу необхідно провести повну перевірку коректності виконання та відображення елементів в браузері.

Для тестування сайту необхідно виконати наступні дії по його встановленню: провести інсталяцію локального серверу, встановити файли в директорію встановленого ПЗ та імпортувати БД в систему адміністрування.

Для тестування розробленого сайту його необхідно запусити з допомогою локального серверу. Для установки сервісу спочатку завантажується інсталятор локального серверу «Open Server» за наступним посиланням: <https://ospanel.io/news/>. По завершенню завантаження проводиться установка середовища. Після інсталяції необхідно запусити програму. В треї Windows з'явиться іконка прапорця, натиснувши на яку можна викликати контекстне меню програми та запусити сервер, або провести налаштування системи, як показано на рисунку 2.35.

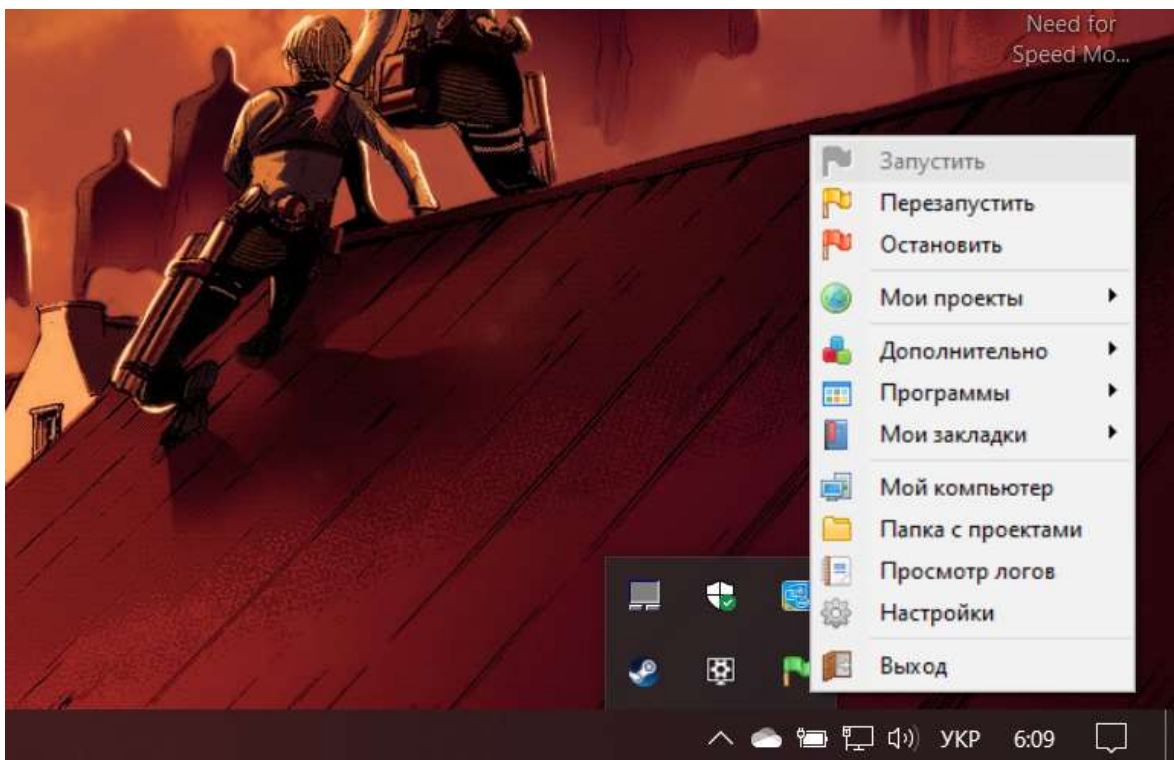


Рисунок 2.35 – Запуск Open Server

Далі необхідно завантажити файли сайту в директорію за шляхом: `C:\openserver\domains` та, якщо сервер запущений, перезавантажити його, що продемонстровано на рисунку 2.36.

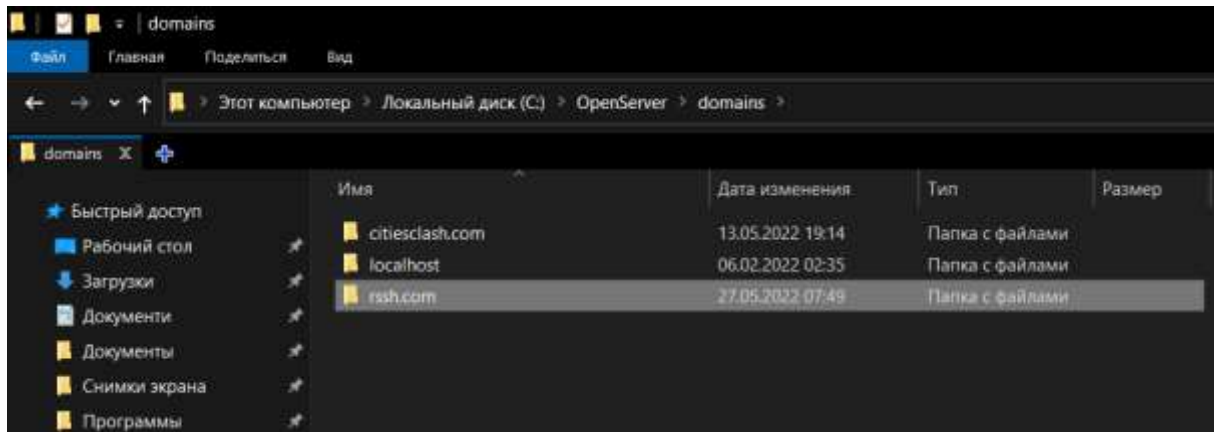


Рисунок 2.36 – Шлях розміщення файлів веб-сайту

Після цього потрібно увійти в акаунт phpMyAdmin та імпортувати базу даних «rsshdb.sql». За шляхом: C:\opensever\domains\wssh.com\sql розміщені sql-файли БД, як показано на рисунку 2.37.

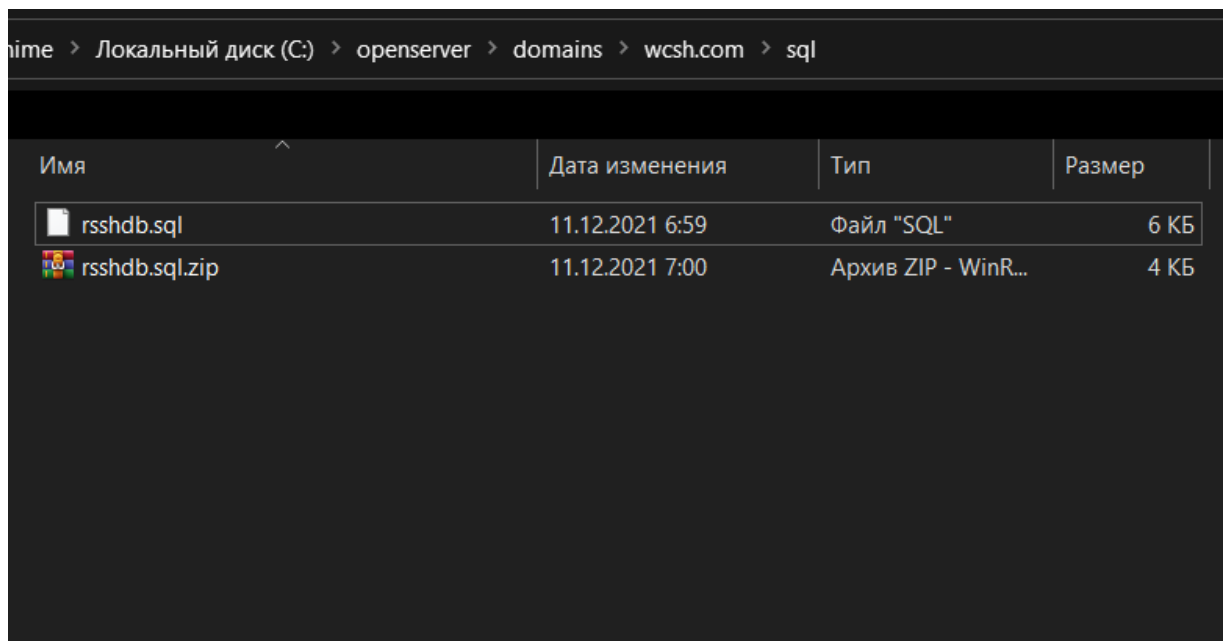


Рисунок 2.37 – Шлях розміщення файлів бази даних

Далі через пункт меню OpenServer «Додатково» відкривається система адміністрування бази даних phpMyAdmin.

Після переходу у систему необхідно натиснути кнопку «Імпорт» та у відкритому вікні файлового менеджера вибрати один з вищевказаних файлів, що продемонстровано на рисунку 2.38.

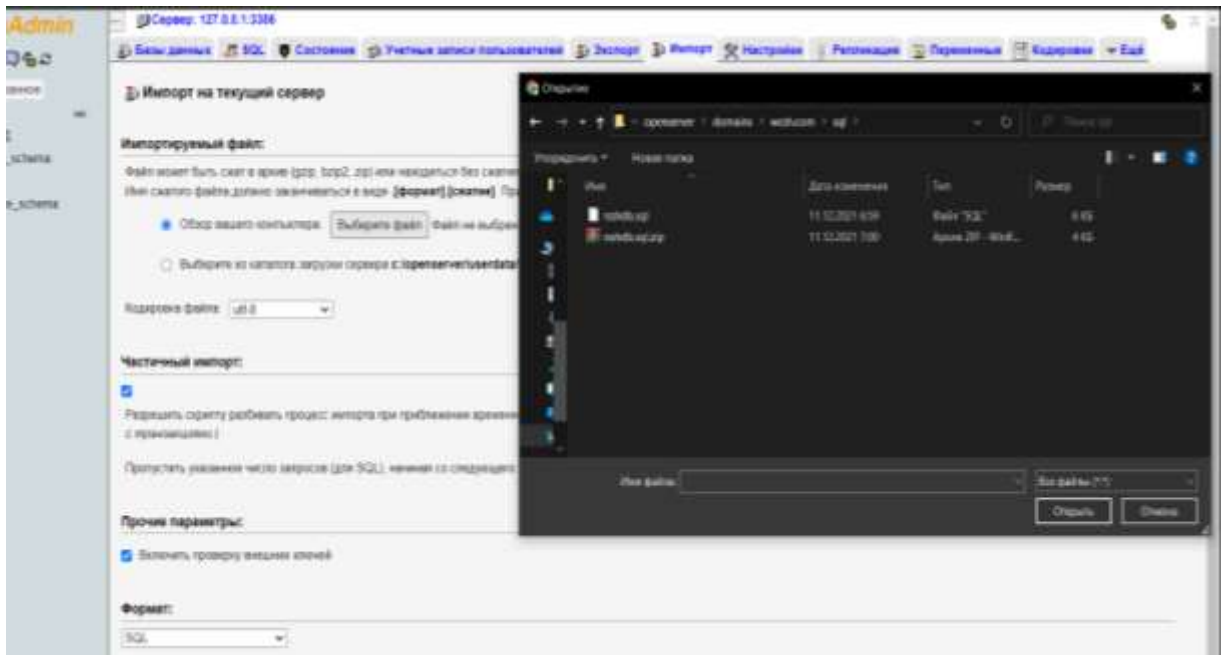


Рисунок 2.38– Імпорт БД

Після підтвердження дії таблиці будуть завантажені в БД та відобразяться на панелі зліва, як на рисунку 2.39.

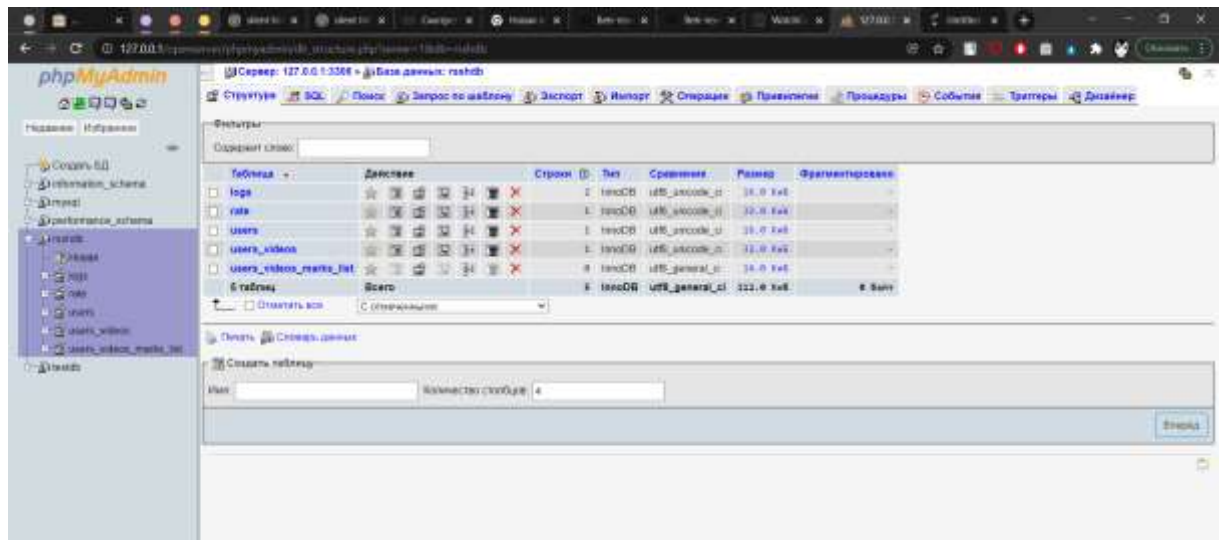


Рисунок 2.39 – Результат імпорту БД

Останнім кроком буде перевірка файлу connest.php. Якщо логін та пароль не співпадають з Вашими в phpMyAdmin, необхідно переписати дані, як показано на рисунку 2.40.

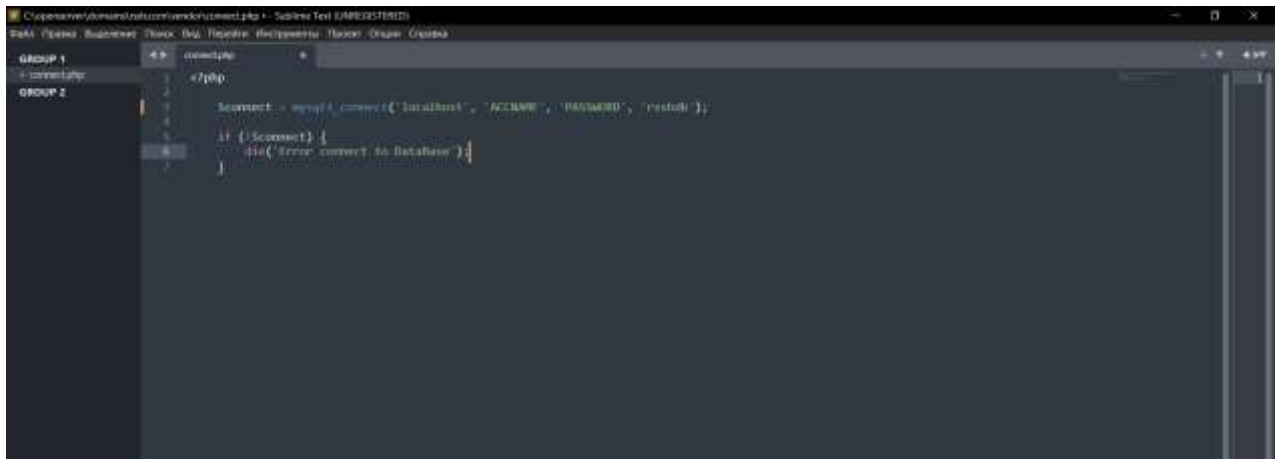


Рисунок 2.40 – Вміст файлу «connect.php»

Фінальним етапом тестування будь-якого проекту є перевірка валідації коду. Це необхідно для виявлення зайвих елементів, змінних, коду, який повтрюється чи властивостей. Також метод перевірки коду також допомагає уникнути можливих помилок та виявити вже створені.

Для тестування коду файлів веб-сайту відеохостингу було використано ресурс «PHP Tester», інтерфейс якого зображений на рисунку 2.41.



Рисунок 2.41 – Перевірка валідності програмного коду у файлі

Для перевірки валідності коду достатньо вставити його в поле вводу та натиснути на кнопку перевірки. Результат буде виведено у вікні зліва.

Для валідації HTML розмітки використовується сервіс «Involta». Головною перевагою цього сайту є візуальне відображення введеного коду з виділенням усіх помилок та їх детальний опис, як показано на рисунку 2.42.



Рисунок 2.42 – Перевірка валідності html-коду

В результаті виконання перевірки валідності коду в html розмітці були виявлені елементи, рекомендовані для виправлення сервісом.

2.5 Висновок до другого розділу

В результаті роботи над другим розділом кваліфікаційної роботи шляхом побудови UML-діаграм та моделей, була проведена робота зі створення загальної структури розроблюваного веб-сайту та абстрактного відображення окремих його елементів.

На основі діаграм та раніше визначених вимог, програмного забезпечення, інструментів була розроблено програмну частину веб-сайту відеохостингу «R.S.Sh» з функцією онлайн-кінотеатру.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Вимоги ергономіки до організації робочого місця оператора ПК

Робота з комп'ютером характеризується значною розумовим та нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи та досить великим навантаженням на м'язи рук при роботі з клавіатурою ПК. Велике значення має раціональна конструкція та розташування елементів робочого місця, що важливо для підтримки оптимальної робочої пози людини-оператора.

У процесі роботи з комп'ютером необхідно дотримуватися правильного режиму праці та відпочинку. В іншому випадку у персоналу відзначаються значне напруження зорового апарату з появою скарг на незадоволеність роботою, головний біль, дратівливість, порушення сну, втому і хворобливі відчуття в очах, в попереку, в області шиї та руках.

Вимоги щодо організації та обладнання робочих місць при розробці програмного забезпечення включає в себе площу, відведену на одне робоче місце не менше 6 м². Конструкція робочого місця повинна забезпечувати підтримання оптимальної робочої пози, тобто такої, яка дозволяє працівникові при написанні коду на ПК виконувати роботу з мінімальним напруженням тіла, і яка дозволяє уникнути перевтоми в ході і після закінчення робочого процесу.

Раціональна робоча поза має важливе значення для збереження здоров'я працівника, оскільки тривале перебування його в незручній і напруженій позі може призвести до таких захворювань, як сколіоз, варикозне розширення вен, плоскостопість тощо. Установлено, що робота в зігнутому положенні збільшує затрати енергії на 20%, а при значному нахиленні — на 45% порівняно з прямим положенням корпусу.

За потреби особливої концентрації уваги під час виконання робіт з написання коду для проекту суміжні робочі місця необхідно відділяти одне від одного перегородками висотою 1,5 – 2 м [25].

Забарвлення приміщень та меблів має сприяти створенню сприятливих умов для зорового сприйняття та позитивного настрою.

Джерела світла, такі як світильники та вікна, які дають відображення від поверхні екрана, значно погіршують точність знаків і спричиняють перешкоди фізіологічного характеру, які можуть виразитися у значній напрузі, особливо при тривалій роботі.

Недостатність освітлення призводить до напруги зору, послаблює увагу, призводить до настання передчасної втоми. Надмірно яскраве освітлення викликає засліплення, роздратування та різь в очах. Неправильний напрямок світла робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань, тому настільки важливим є правильний розрахунок освітленості.

Відображення, включаючи відображення від вторинних джерел світла, має бути мінімальним. Для захисту від надмірної яскравості вікон можуть бути використані штори та екрани.

Робочі місця слід розташовувати відносно джерела природного світла, тобто вікон, таким чином, щоб світло падало на клавіатуру програміста збоку, переважно зліва. Також робоче місце для роботи на ПК має відповідати сучасним вимогам ергономіки [26]:

- стіл повинен мати висоту поверхні 680 - 800 мм, ширину 600 - 1400 мм і глибину 800 - 1000 мм. Такі параметри забезпечують можливість виконання операцій в зоні досяжності працівника;

- робочий стілець робочий стілець має бути підйомно-поворотним, з можливістю регулювання висоти, бажано зі стаціонарними або змінними підлікотниками і напівм'якою нековзкою поверхнею сидіння, що легко чиститься і не електризується;

- екран комп'ютера має розташовуватися на оптимальній відстані від користувача з урахуванням літерно-цифрових знаків і символів. Стандартне значення становить 600 – 700 мм.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість монітору, зручність ручного керування пристроєм введення-виведення інформації.

3.2 Вимоги до безпеки з охорони праці та профілактичних медичних оглядів для працівників ПК

Техніка безпеки для програміста – це запорука довготривалої та безпроблемної роботи такого фахівця. Техніка безпеки програмістів регулюється «Інструкцією з охорони праці», де все розкладено за пунктами та дуже докладно описано. Знати її потрібно, якщо програміст працює у великій офісній будівлі, де до комп'ютера мають непрямий доступ кілька людей. У цьому випадку він зобов'язаний слідувати інструкціям техніки безпеки, щоб не наражати на небезпеку своє здоров'я і здоров'я оточуючих його колег. Плюс програміст просто повинен знати, як поводитися під час надзвичайних ситуацій.

Зазвичай техніка безпеки програміста зачитується фахівцями з безпеки праці кожної організації, де працюють програмісти, чи вона доступна прочитання кожним співробітником. А щорічно, іноді й частіше, всі співробітники розписуються, що ознайомлені з технікою безпеки. Фактично техніку безпеки мало хто читає, мало хто знає і мало хто дотримується, тому що все обмежується тим, що програмісти просто розписуються в журналі, ніби вони «ознайомлені» і спокійно працюють далі.

Коли програміст працює віддалено з дому, вся відповідальність за його безпеку лежить на його плечах. Коли програміст працює в офісі, то відповідальність за його безпеку лежить на плечах програміста та окремого спеціаліста, який має слідкувати за дотриманням техніки безпеки. І в тому, і в іншому випадку програміст повинен знати основи охорони праці, описані нижче.

Вся техніка безпеки для програміста поділяється на кілька етапів: до початку роботи, під час роботи, після закінчення роботи; у разі аварійної ситуації.

Програміст перед стартом своєї роботи повинен:

- провести огляд свого робочого місця;
- провести регулювання освітлення, щоб екран був добре видно і не відображав світло;
- проконтролювати коректне підключення електричних частин комп'ютера до мережі;
- проконтролювати відсутність оголених частин проведення на електричних проводах комп'ютера;
- провести перевірку цілісності столу, стільця, підставки для ніг, висувної частини столу для клавіатури тощо; якщо потрібно, програміст повинен відрегулювати всі ці моменти.

Під час своєї роботи програміст повинен:

- стежити за чистотою свого робочого місця;
- не закривати вентиляційні вікна комп'ютера;
- коректно припиняти роботу комп'ютера, коли це необхідно;
- стежити за дотриманням свого графіка роботи та відпочинку;
- правильно та за призначенням експлуатувати комп'ютер та всі його частини;
- вчасно виконувати фізичні вправи для очей, шиї, рук та тулуба;
- стежити за своїм розташуванням на робочому місці: правильна постава, відстань до та екрану тощо.

Після закінчення робочого дня або свого робочого часу програміст повинен:

- правильно завершити роботу всіх запущених програм та пристроїв;
- перевірити відсутність у дисководі дисків чи дискет;
- відключити системний блок від електромережі;
- вимкнути додаткові пристрої від електромережі;

– оглянути своє робоче місце і привести його до ладу, якщо це необхідно.

Перш за все, відповідно до ст. 169 Кодексу законів про працю України та ст. 17 Закону України «Про охорону праці» від 2002 року роботодавець зобов'язаний за свої кошти організувати проведення попереднього (при прийнятті на роботу) і періодичних (протягом трудової діяльності) медичних оглядів працівників, зайнятих на важких роботах, роботах зі шкідливими чи небезпечними умовами праці або таких, де є потреба у професійному доборі, а також щорічного обов'язкового медичного огляду осіб віком до 21 року.

Метою проведення обов'язкових профілактичних медичних оглядів є запобігання розповсюдженню інфекційних та небезпечних захворювань, динамічне спостереження за станом здоров'я працюючого населення.

Такий вид оглядів передбачений статтею 21 Закону України «Про захист населення від інфекційних хвороб» та статтею 26 Закону України «Про забезпечення санітарного та епідемічного благополуччя населення».

Таким чином працівники окремих професій, виробництв та організацій, діяльність яких пов'язана з обслуговуванням населення і може призвести до поширення інфекційних хвороб, повинні проходити обов'язкові попередні (до прийняття на роботу) і періодичні профілактичні медичні огляди.

3.3 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано вимоги ергономіки робочого місця, оформлення робочого кабінету та необхідних умов для роботи за ПК. Описані фактори та умови, які позитивно та негативно впливають на людину при роботі. Також висвітлені теми вимог безпеки та проведення профілактичних медичних оглядів для операторів ПК.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи був спроектований та реалізований веб-сайт відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру. Повний шлях розробки від планування до реалізації було розділено на два розділи роботи.

В першому розділі проведено роботу по дослідженню предметної області та ринку послуг, розгляд актуальності теми розробки, потенційної аудиторії користувачів. Також виконано встановлення вимог до веб-сайту у відповідності з потребами замовника та отриманою інформацією. Були розглянуті та підібрані варіанти використовуваного програмного забезпечення та вибір оптимального середовища розробки.

В другому розділі, шляхом побудови UML-діаграм та моделей, була проведена робота зі створення загальної структури розроблюваного веб-сайту та абстрактного відображення окремих його елементів. Написано програмний код проекту. Детально описано інтерфейс та виконання функціональних модулів реалізованого веб-сайту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру.

Третій розділ роботи присвячений вимогам ергономіки при роботі з комп'ютером, а також вимогам безпеки та проведенню профілактичних заходів.

Мета та завдання, встановлені для студента, були виконані в повному обсязі, а саме:

- Аналіз предметної області та сфери тематики розроблюваного веб-сайту.
- Визначення використовуваних технологій.
- Аналіз варіантів використання розроблюваного веб-сайту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру.
- Розробка моделей даних та їх структури.
- Розробка структури сайту та його елементів.
- Реалізація коду з допомогою визначених технологій розробки.

– Виконання перевірки валідності коду розробленого веб-сайту відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру.

Результатом виконання процесу розробки став реалізований веб-сайт відеохостингу «R.S.Sh.» з функцією онлайн-кінотеатру. Створений проект повністю задовільняє вимоги, поставлені на етапі планування.

При розробці на практиці були використані та розширені набуті за час вивчення курсу знання. В подальшому веб-сайт буде вдосконалюватись та доповнюватись функціонально.

ПЕРЕЛІК ДЖЕРЕЛ

1. Історія Інтернету від ARPANET до сьогодні [Електронний ресурс] // Ucloud. – 2019. – Режим доступу до ресурсу: <https://ucloud.ua/istoriya-internetu-vid-arpanet-do-sogodni/>.
2. Правильна структура веб-сайту під SEO: приклади, види і рекомендації з розробки структури [Електронний ресурс] // БМБ. – 2021. – Режим доступу до ресурсу: <https://www.bmb.com.ua/2021/02/seo.html>.
3. HTML [Електронний ресурс] // Wikipedia. – 2022. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>.
4. CSS [Електронний ресурс] // Wikipedia. – 2022. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/CSS>.
5. PHP проти HTML – у чому різниця між PHP і HTML [Електронний ресурс] // Myservername. – 2020. – Режим доступу до ресурсу: <https://uk.myservername.com/php-vs-html-what-is-difference-between-php>.
6. Основні переваги СУБД MySQL [Електронний ресурс] // НТУ. – 2018. – Режим доступу до ресурсу: <https://studfile.net/preview/5607354/page:3/>.
7. Обґрунтування вибору технології розробки, програмного середовища та мови програмування [Електронний ресурс] // Студіопедія. – 2020. – Режим доступу до ресурсу: https://studopedia.ru/21_90335_obruntuvannya-viboru-tehnologii-rozrobki-programnogo-seredovishcha-ta-movi-programuvannya.html.
8. Москаленко А. В. Програмне середовище для веб розробки. [Електронний ресурс] / А. В. Москаленко // OSPanel.io. – 2022. – Режим доступу до ресурсу: <https://ospanel.io/>.
9. Діаграма прецедентів. [Електронний ресурс] // Wikipedia. – 2021. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2.

10. Що таке трирівнева архітектура? [Електронний ресурс] // Розвиток. – 2022. – Режим доступу до ресурсу: <https://uk.theastrologypage.com/three-tier-architecture>.
11. Павлов А. В. Огляд підходів до організації інтелектуального інтерфейсу користувача [Електронний ресурс] / А. В. Павлов // МАН. – 2019. – Режим доступу до ресурсу: <http://www.mgua.irtc.org.ua/attach/IMCS/2019/10.pdf>.
12. Лекція 3. GRID і бази даних [Електронний ресурс] // ShporaMe. – 2022. – Режим доступу до ресурсу: <https://shpora.me/mike/db2>.
13. Трач Р. В. Моделювання організаційної структури проекту [Електронний ресурс] / Р. В. Трач // Вісник НУВГП. – 2019. – Режим доступу до ресурсу: <http://ep3.nuwm.edu.ua/16624/>.
14. Різниця між логічною та фізичною моделлю даних [Електронний ресурс] // Sawakinome. – 2021. – Режим доступу до ресурсу: <https://ua.sawakinome.com/articles/technology/difference-between-logical-and-physical-data-model-2.html>.
15. Перехід до фізичної моделі бази даних [Електронний ресурс] // Підручники для студентів онлайн. – 2022. – Режим доступу до ресурсу: https://stud.com.ua/77259/informatika/perehid_fizichnoyi_modeli_bazi_danih.
16. Проектування інтерфейсу для сайту та онлайн сервісу [Електронний ресурс] // D.Logic. – 2022. – Режим доступу до ресурсу: <https://dlogic.com.ua/ua/service/proektuvannya-interfeisiv/>.
17. Як правильно оформити головну сторінку сайту [Електронний ресурс] // HOSTiQ. – 2021. – Режим доступу до ресурсу: <https://hostiq.ua/blog/ukr/main-page/>.
18. Тег «form» [Електронний ресурс] // htmlbook. – 2018. – Режим доступу до ресурсу: <http://htmlbook.ru/html/form>.
19. Оператор if [Електронний ресурс] // The PHP Group. – 2021. – Режим доступу до ресурсу: <https://www.php.net/manual/ru/control-structures.if.php>.

20. Регулярні вирази [Електронний ресурс] // Codedokode. – 2018. – Режим доступу до ресурсу: <http://archive-ipq-co.narod.ru/11/regexp.html>.
21. Уроки PHP – регулярні вирази з прикладами [Електронний ресурс] // Sebweo. – 2019. – Режим доступу до ресурсу: <https://sebweo.com/uroki-php-regulyarni-virazi-z-prikladami/>.
22. Оператор foreach [Електронний ресурс] // The PHP Group. – 2021. – Режим доступу до ресурсу: <https://www.php.net/manual/ru/control-structures.foreach.php>.
23. Функція include [Електронний ресурс] // The PHP Group. – 2021. – Режим доступу до ресурсу: <https://www.php.net/manual/ru/function.include.php>.
24. Тег «a» [Електронний ресурс] // htmlbook. – 2020. – Режим доступу до ресурсу: <http://htmlbook.ru/html/a>.
25. Організація робочого місця оператора ПК [Електронний ресурс] // Studcon. – 2021. – Режим доступу до ресурсу: <http://studcon.org/organizaciya-robochogo-miscya-operatora-pk>.
26. Охорона праці в офісі. Вимоги до робочого місця офісного працівника [Електронний ресурс] // AGN International. – 2021. – Режим доступу до ресурсу: <https://gc.ua/uk/oxorona-praci-v-ofisi-vimogi-do-robochogo-miscya-ofisnogo-pracivnika/>.
27. Скляр Д. Вивчаємо PHP 7. Керівництво по створенню інтерактивних веб-сайтів / Девід Скляр. – Бостон: Бейджин, 2017. – 464 с.
28. Зандстра М. PHP. Об'єкти, шаблони, методики програмування. / Метт Зандстра. – Нью-Йорк: Апрес, 2019. – 577 с.
29. Тейлор А. SQL для чайників / Аллен Тейлор. – Нью-Джерсі: Хобокен, 2019. – 515 с.
30. Ніксон Р. Створюємо динамічні веб-сайти з допомогою PHP, MySQL, JavaScript, CSS, HTML 5 / Робін Ніксон. – Ньютон: Орейлі Медіа, 2018. – 528 с.

ДОДАТКИ

Програмний код файлу singup.php

```

<?php

session_start();
require_once 'connect.php';

$regnickname = $_POST['regnickname'];
$regemail = $_POST['regemail'];
$regpassword = $_POST['regpassword'];
$confregpassword = $_POST['confregpassword'];
$regbdate = $_POST['regbdate'];
$regcountry = $_POST['regcountry'];

$regdate = date('Y-m-d H:i');

$usersDataN = mysqli_query($connect, "SELECT * FROM `users`
WHERE `nickname` = '$regnickname'");
$usersDataE = mysqli_query($connect, "SELECT * FROM `users`
WHERE `email` = '$regemail'");

$usersCheckN = mysqli_fetch_assoc($usersDataN);
$usersCheckE = mysqli_fetch_assoc($usersDataE);

if ($usersCheckN['nickname'] == $regnickname &&
$usersCheckE['email'] == $regemail) {

    $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Ім'я користувача та адреса електронної пошти вже
zareestrovani!</span>";
    header('Location: ../index.php');

} else if ($usersCheckN['nickname'] == $regnickname) {

    $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Ім'я користувача вже zareestrovano!</span>";
    header('Location: ../index.php');

} else if ($usersCheckE['email'] == $regemail) {

    $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Адреса електронної пошти вже zareestrovana!</span>";
    header('Location: ../index.php');
} else {

    if ($regpassword === $confregpassword) {

        if (preg_match("/^[\\w_-]{3,20}$/", $regnickname) &&
preg_match("/^[\\w\\-\\.]+@[a-z]+[.]+[a-z]/", $regemail)) {
            if ($_FILES['regavatar']['size'] > 0) {

$spath = 'uploads/' . time() . $_FILES['regavatar']['name'];

```

```

        if
(!move_uploaded_file($_FILES['regavatar']['tmp_name'], '../' .
$path)) {
            $_SESSION['message'] = "<span
style='color:red;font-weight:bold;'>Помилка завантаження
фотографії!</span>";
            header('Location: ../index.php');
        }

        $regpassword = md5($regpassword);

        mysqli_query($connect, "INSERT INTO `users` (`id`,
`nickname`, `email`, `password`, `avatar`, `bdate`, `country`,
`user_first_log`, `user_last_log`) VALUES (NULL, '$regnickname',
'$regemail', '$regpassword', '$path', '$regbdate', '$regcountry',
'$regdate', '$regdate')");

        $_SESSION['message'] = "<span
style='color:green;font-weight:bold;'>Профіль успішно
створено!</span>";
        header('Location: ../index.php');
    } else {
        $_SESSION['message'] = "<span
style='color:red;font-weight:bold;'>Фото не вибрано!</span>";
        header('Location: ../index.php');
    }

    } else {
        $_SESSION['message'] = "<span
style='color:red;font-weight:bold;'>Введені в поля дані не
відповідають вказаним форматам!</span>";
        header('Location: ../index.php');
    }

    } else {
        $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Паролі не співпадають</span>";
        header('Location: ../index.php');
    }
}
?>

```

Програмний код файлу connect.php

```
<?php

    $connect = mysqli_connect('localhost', 'root', 'root',
    'rsshdb');

    if (!$connect) {
        die('Error connect to DataBase');
    }
```

Програмний код файлу `singin.php`

```
<?php

    session_start();
    require_once 'connect.php';

    $email = $_POST['email'];
    $password = md5($_POST['password']);
    $indate = date('Y-m-d H:i');

    $check_user = mysqli_query($connect, "SELECT * FROM `users`
WHERE `email` = '$email' AND `password` = '$password'");
    if (mysqli_num_rows($check_user) > 0) {

        $user = mysqli_fetch_assoc($check_user);

        $_SESSION['user'] = [
            "id" => $user['id'],
            "rank" => $user['rank']
        ];

        mysqli_query($connect, "UPDATE `users` SET
`user_last_log` = '$indate' WHERE `id` =
{$_SESSION['user']['id']}");

        header('Location: ../profile.php');

    } else {
        $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Невірний логін або пароль!</span>";
        header('Location: ../index.php');
    }
?>

<pre>
    <?php
        print_r($check_user);
        print_r($user);
    ?>
</pre>
```

Програмний код файлу edithuser.php

```
<?php

    session_start();
    require_once 'connect.php';

    $editnickname = $_POST['editnickname'];
    $editemail = $_POST['editemail'];
    $editbdate = $_POST['editbdate'];
    $editcountry = $_POST['editcountry'];

    $usersDataN = mysqli_query($connect, "SELECT * FROM `users`
WHERE `nickname` = '$editnickname'");
    $usersDataE = mysqli_query($connect, "SELECT * FROM `users`
WHERE `email` = '$editemail'");

    $usersCheckN = mysqli_fetch_assoc($usersDataN);
    $usersCheckE = mysqli_fetch_assoc($usersDataE);

    if (preg_match("/^[\\w_-]{3,20}$/", $editnickname) &&
preg_match("/^[\\w\\-\\.]+@[a-z]+[.]+[a-z]/", $editemail)) {

        if ($usersCheckN['nickname'] == $editnickname &&
$usersCheckE['email'] == $editemail) {
            mysqli_query($connect, "UPDATE `users` SET `bdate`=
'$editbdate', `country`= '$editcountry' WHERE `id`=
{$_SESSION['user']['id']}");

            $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Ім'я користувача та адреса електронної пошти вже
zareєстровані та не були змінені!</span>";
            header('Location: ../profile.php');

        } else if ($usersCheckN['nickname'] == $editnickname) {

            mysqli_query($connect, "UPDATE `users` SET `email`=
'$editemail', `bdate`= '$editbdate', `country`= '$editcountry'
WHERE `id`= {$_SESSION['user']['id']}");

            $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Ім'я користувача вже зареєстровано та не було
змінено!</span>";
            header('Location: ../profile.php');

        } else if ($usersCheckE['email'] == $editemail) {

            mysqli_query($connect, "UPDATE `users` SET `nickname`=
'$editnickname', `bdate`= '$editbdate', `country`= '$editcountry'
WHERE `id`= {$_SESSION['user']['id']}");
```

```
        $_SESSION['message'] = "<span style='color:red;font-  
weight:bold;'>Адреса електронної пошти вже зареєстрована та не  
була змінена!</span>";  
        header('Location: ../profile.php');  
  
    } else {  
  
        mysqli_query($connect, "UPDATE `users` SET `nickname`=  
'$editnickname', `email`= '$editemail', `bdate`= '$editbdate',  
`country`= '$editcountry' WHERE `id`= {$_SESSION['user']['id']}");  
  
        $_SESSION['message'] = "<span style='color:green;font-  
weight:bold;'>Дані успішно оновлено!</span>";  
        header('Location: ../profile.php');  
  
    }  
    } else {  
        $_SESSION['message'] = "<span style='color:red;font-  
weight:bold;'>Введені в поля дані не відповідають вказаним  
форматам!</span>";  
        header('Location: ../profile.php');  
    }  
    ?>
```


Програмний код файлу accdelete.php

```
<?php

    session_start();
    require_once 'connect.php';
    if (!$_SESSION['user']) {
        header('Location: /');
    }

    $sid = $_SESSION['user']['id'];

    $sucomreports = mysqli_query($connect, "SELECT * FROM
`video_comments` WHERE `id_user` = '$sid'");

    while ($sucreports = mysqli_fetch_assoc($sucomreports))
    {
        $sucrId = $sucreports['id'];

        mysqli_query($connect, "DELETE from reports_comments
WHERE id_comment ='$sucrId'");
    }

    $suvideos = mysqli_query($connect, "SELECT * FROM
`users_videos` WHERE `id_user` = '$sid'");

    while ($videos = mysqli_fetch_assoc($suvideos)) {
        $videoId = $videos['id'];

        mysqli_query($connect, "DELETE from video_comments
WHERE id_video='$videoId'");

        mysqli_query($connect, "DELETE from reports_videos
WHERE id_video='$videoId'");
    }

    $sunews = mysqli_query($connect, "SELECT * FROM
`users_news` WHERE `id_user` = '$sid'");

    while ($news = mysqli_fetch_assoc($sunews)) {
        $newsId = $news['id'];

        mysqli_query($connect, "DELETE from reports_news
WHERE id_news='$newsId'");
    }

    mysqli_query($connect, "DELETE from users WHERE
id='$sid'");

    mysqli_query($connect, "DELETE from users_videos WHERE
id_user='$sid'");
    mysqli_query($connect, "DELETE from rate WHERE
id_user='$sid'");
```

```
        mysqli_query($connect, "DELETE from logs WHERE
id_user='$id'");

        mysqli_query($connect, "DELETE from
users_videos_marks_list WHERE id_user='$id'");

        mysqli_query($connect, "DELETE from
users_videos_marks_list WHERE id_author='$id'");

        mysqli_query($connect, "DELETE from video_comments
WHERE id_user='$id'");

        mysqli_query($connect, "DELETE from reports_accepted
WHERE id_user='$id'");

        mysqli_query($connect, "DELETE from users_subscribes
WHERE id_user='$id'");

        mysqli_query($connect, "DELETE from users_subscribes
WHERE id_author='$id'");

        mysqli_query($connect, "DELETE from users_news WHERE
id_user='$id'");

        unset($_SESSION['user']);
        header('Location: ../index.php');
?>
```

Програмний код файлу uploaduservideo.php

```
<?php

    session_start();
    require_once 'connect.php';

    $videoname = $_POST['videoname'];
    $videodesc = $_POST['videodesc'];
    $iduser = $_SESSION['user']['id'];
    $videodate = date('Y-m-d H:i');
    $videofilmtag = $_POST['videofilmtag'];
    $targettag = $_POST['targettag'];
    $targettag2 = $_POST['targettag2'];
    $targettag3 = $_POST['targettag3'];

    if ($videoname == NULL OR $videodesc == NULL) {
        $_SESSION['message'] = "<span style='color:red;font-weight:bold;'>Всі поля повинні бути заповненими!</span>";
        header('Location: ../profile.php');
    }else{
        $array = explode(" ", $videodesc);
        $success = true;

        foreach ($array as $word) {
            if(strlen($word) > 45) {
                $success = false;
                $_SESSION['message'] = "<span style='color:red;font-weight:bold;'>В звичайній розмовній лексиці слів довше 45-ти символів не буває! :)</span>";
                header('Location: ../profile.php');
            }
        }
        if($success) {
            if ($_FILES['uploadvideo']['size'] > 0 &&
$_FILES['uploadpreview']['size'] > 0) {
                $path = 'uploads/upvideos/' . time() .
$_FILES['uploadvideo']['name'];
                if
(!move_uploaded_file($_FILES['uploadvideo']['tmp_name'], '../' .
$path)) {
                    $_SESSION['message'] = 'Помилка!';
                    header('Location: ../profile.php');
                }

                $preview = 'uploads/upvideos/previews/' . time() .
$_FILES['uploadpreview']['name'];
                if
(!move_uploaded_file($_FILES['uploadpreview']['tmp_name'], '../' .
$preview)) {
                    $_SESSION['message'] = 'Помилка!';
                    header('Location: ../profile.php');
                }
            }
        }
    }
}
```

```

        $vidscountselect = mysqli_query($connect, "SELECT
`videos_count` FROM `users` WHERE `id` = '$iduser'");
        $videoscountu = mysqli_fetch_assoc($vidscountselect);
        $videoscount = $videoscountu['videos_count'];
        $newvideoscount = $videoscount + 1;

        mysqli_query($connect, "INSERT INTO `users_videos` (`id`,
`id_user`, `video_name`, `video_desc`, `video_film_tag`,
`video_target_tag`, `video_target_tag2`, `video_target_tag3`,
`user_video`, `video_preview`, `video_date`) VALUES (NULL,
'$iduser', '$videoname', '$videodesc', '$videofilmtag',
'$targettag', '$targettag2', '$targettag3', '$path', '$preview',
'$videodate')");

        mysqli_query($connect, "UPDATE `users` SET `videos_count`
= '$newvideoscount' WHERE `id` = '$iduser'");

        $_SESSION['message'] = "<span style='color:green;font-
weight:bold;'>Відео опубліковано!</span>";
        header('Location: ../profile.php');
    }
    else {
        $_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>Відео або прев'ю не вибрано!</span>";
        header('Location: ../profile.php');
    }
}
?>

```

Програмний код файлу reportsend.php

```
<?php
    session_start();
    require_once 'connect.php';

    $idUser = $_SESSION['user']['id'];

    if (isset($_POST['btnreportvideo']) &&
is_numeric($_POST['btnreportvideo'])) {
        $videoId = $_POST['btnreportvideo'];
        $reporttext = $_POST['txtareareportvideo'];
        $reportcause = $_POST['reportcause'];
        $reportdate = date('Y-m-d H:i');

        $array = explode(" ", $reporttext);
        $success = true;

        foreach ($array as $word) {
            if(strlen($word) > 45) {
                $success = false;
                $_SESSION['message'] = "<span
style='color:red;font-weight:bold;'>В звичайній розмовній лексичі
слів довше 45-ти символів не буває! :)</span>";
                header("Location: ".$_SERVER['HTTP_REFERER']);
            }
        }
        if($success) {
            mysqli_query($connect, "INSERT INTO `reports_videos`
(`id`, `id_video`, `id_user`, `report_cause`, `report_text`,
`report_date`) VALUES (NULL, '$videoId', '$idUser',
'$reportcause', '$reporttext', '$reportdate')");

            $_SESSION['message'] = "<span style='color:green;font-
weight:bold;'>Скаргу відправлено модератору!</span>";
            header("Location: ".$_SERVER['HTTP_REFERER']);
        }
    }else if(isset($_POST['btnreportcomment']) &&
is_numeric($_POST['btnreportcomment'])) {
        $commentId = $_POST['btnreportcomment'];
        $reporttext = $_POST['txtareareportcomment'];
        $reportcause = $_POST['reportcause'];
        $reportdate = date('Y-m-d H:i');

        $array = explode(" ", $reporttext);
        $success = true;

        foreach ($array as $word) {
            if(strlen($word) > 45) {
                $success = false;
```

```

$_SESSION['message'] = "<span style='color:red;font-
weight:bold;'>В звичайній розмовній лексиці слів довше 45-ти
символів не буває! :)</span>";
        header("Location: ".$_SERVER['HTTP_REFERER']);
    }
}
if($success) {
    mysqli_query($connect, "INSERT INTO `reports_comments`
(`id`, `id_comment`, `id_user`, `report_cause`, `report_text`,
`report_date`) VALUES (NULL, '$commentId', '$idUser',
'$reportcause', '$reporttext', '$reportdate')");

    $_SESSION['message'] = "<span style='color:green;font-
weight:bold;'>Скаргу відправлено модератору!</span>";
    header("Location: ".$_SERVER['HTTP_REFERER']);
}
}else if (isset($_POST['btnreportnews']) &&
is_numeric($_POST['btnreportnews'])) {
    $newsId = $_POST['btnreportnews'];
    $reporttext = $_POST['txtareareportnews'];
    $reportcause = $_POST['reportcause'];
    $reportdate = date('Y-m-d H:i');

    $array = explode(" ", $reporttext);
    $success = true;

    foreach ($array as $word) {
        if(strlen($word) > 45) {
            $success = false;
            $_SESSION['message'] = "<span
style='color:red;font-weight:bold;'>В звичайній розмовній лексиці
слів довше 45-ти символів не буває! :)</span>";
            header("Location: ".$_SERVER['HTTP_REFERER']);
        }
    }
    if($success) {
        mysqli_query($connect, "INSERT INTO `reports_news`
(`id`, `id_news`, `id_user`, `report_cause`, `report_text`,
`report_date`) VALUES (NULL, '$newsId', '$idUser', '$reportcause',
'$reporttext', '$reportdate')");

        $_SESSION['message'] = "<span style='color:green;font-
weight:bold;'>Скаргу відправлено модератору!</span>";
        header("Location: ".$_SERVER['HTTP_REFERER']);
    }
}
?>

```

Програмний код файлу `uservideodisplay.php`

```
<?php
    session_start();
    require_once 'vendor/connect.php';

    while($userid = mysqli_fetch_assoc($result)){

        $videoId = $userid['id'];
        $videoIduser = $userid['id_user'];

        $likescount = $userid['video_likes'];
        $dislikescount = $userid['video_dislikes'];
        $totalcount = $userid['video_total_marks'];
        $commentscount = $userid['video_comments'];
        $videofilmtag = $userid['video_film_tag'];
        $videotargettag = $userid['video_target_tag'];
        $videotargettag2 = $userid['video_target_tag2'];
        $videotargettag3 = $userid['video_target_tag3'];
        $videoreportscount = $userid['video_reports'];

        $vidslist = mysqli_query($connect, "SELECT * FROM
`users_videos_marks_list` WHERE `id_user` = '$iduser' AND
`id_video` = '$videoId'");
        $videostatus = mysqli_fetch_assoc($vidslist);

        if ($videostatus == NULL) {
            //стандартні кнопки оцінок
            $btnlikename = 'btnlike';
            $btndislikename = 'btndislike';
            $btnlikeimg = 'https://img.icons8.com/external-justicon-
lineal-justicon/64/000000/external-like-notifications-justicon-
lineal-justicon.png';
            $btndislikeimg =
'https://img.icons8.com/ios/50/000000/poor-quality.png';

        }else if ($videostatus['mark'] == '1' ) {
            //якщо на відео є лайк
            $btnlikename = 'btnlikedeleter';
            $btndislikename = 'btndislike';
            $btnlikeimg = 'https://img.icons8.com/external-yogi-
aprelliyanto-glyph-yogi-aprelliyanto/32/000000/external-like-
shopping-and-ecommerce-yogi-aprelliyanto-glyph-yogi-
aprelliyanto.png';
            $btndislikeimg =
'https://img.icons8.com/ios/50/000000/poor-quality.png';
        }else if ($videostatus['mark'] == '0' ) {
            //якщо на відео дизлайк
            $btnlikename = 'btnlike';
            $btndislikename = 'btndislikedeleter';
```

```

        $btnlikeimg = 'https://img.icons8.com/external-justicon-
lineal-justicon/64/000000/external-like-notifications-justicon-
lineal-justicon.png';
        $btndislikeimg = 'https://img.icons8.com/external-yogi-
aprelliyanto-glyph-yogi-aprelliyanto/32/000000/external-dislike-
shopping-and-ecommerce-yogi-aprelliyanto-glyph-yogi-
aprelliyanto.png';
    }

    if ($totalcount == 0) {
        $percent = 0;
    }else{
        $percent = $likescount / $totalcount * 100;
    }

    $authorS = mysqli_query($connect,"SELECT * FROM `users` WHERE
`id` = '$videoIduser'");
    $author = mysqli_fetch_assoc($authorS);

?>
    <div class="modal fade" id="videoReportModal<?php echo
$videoId ?>" tabindex="-1" role="dialog" aria-
labelledby="videoReportModal<?php echo $videoId ?>" aria-
hidden="true">
        <div class="modal-dialog modal-lg" role="document">
            <div class="modal-content bg-plus-col">
                <div class="modal-header">
                    <h5 class="modal-title"
id="videoReportModal<?php echo $videoId ?>">Скарга</h5>
                    <button class="close col-white"
type="button" data-dismiss="modal" aria-label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
                <div class="modal-body">
                    <p>Виберіть причину скарги:</p>
                    <form method="post"
action="vendor/reportsend.php" enctype="multipart/form-data">
                        <select name="reportcause" class="form-
control">
                            <option value="Демонстрація шкідливих
чи небезпечних дій">Демонстрація шкідливих чи небезпечних
дій</option>
                            <option value="Жорстоке поводження з
дітьми">Жорстоке поводження з дітьми</option>
                            <option value="Інформаційна
пропаганда">Інформаційна пропаганда</option>
                            <option value="Контент для дорослих
без мітки">Контент для дорослих без мітки</option>
                            <option value="Невідповідність
вказаному фільму">Невідповідність вказаному фільму</option>
                            <option value="Невідповідність
ключових тегів">Невідповідність ключових тегів</option>

```



```

                                <option value="Некорректні
висловлювання в сторону людей/меншинств">Некорректні висловлювання
в сторону людей/меншинств</option>
                                <option value="Призов до шкідливих чи
небезпечних дій">Призов до шкідливих чи небезпечних дій</option>
                                <option
value="Погрози/шантаж">Погрози/шантаж</option>
                                <option value="Порушення авторських
прав">Порушення авторських прав</option>
                                <option value="Порушення прав
людини">Порушення прав людини</option>
                                <option value="Пропаганда
ненависті/тероризму/війни">Пропаганда
ненависті/тероризму/війни</option>
                                <option value="Спам">Спам</option>
                                <option value="Сцени наруги чи
насилля">Сцени наруги чи насилля</option>
                                </select>
                                <br>
                                <textarea name="txtareareportvideo"
id="txtareareportvideo" cols="70" rows="10" type="text"
class="form-control" placeholder="Опишіть причину скарги більш
детально" required style="background-
color:#AEADAD;color:white;border-radius: 10px; margin-bottom:
10px;"></textarea>
                                <br>
                                <button type="submit" class="btn btn-
secondary" name="btnreportvideo" value="<?php echo $videoId;
?>">Відправити</button>
                                </form>
                                </div>
                                </div>
                                </div>
                                </div>
<div style="width:80%;height:770px;margin-left:10%;border:10px
solid gray;border-bottom-left-radius:10%;border-bottom-right-
radius:10%;background-color:#343A40;color:white;margin-top:50px;"
id="focusClose<?php echo $videoId; ?>">
    <video controls="controls" src="<?php echo
$userid['user_video']; ?>" poster="<?php echo
$userid['video_preview']; ?>" type="video/mp4"
style="width:100%;height:90%;object-fit: cover;"></video>
    <source src="<?php echo $userid['user_video']; ?>"
type='video/mp4; codecs="theora, vorbis"'>
    <div style="text-align:center;" id="focus<?php echo $videoId;
?>">
        <span style="font-size:20px;font-family:Arial,sans-
serif;margin-left:8%;margin-right:8%;"><?php echo
$userid['video_name']; ?></span><br>
        </div>
        <div class="row" style="margin-left: 7%;">
            <div class="col">
                <div class="px-4 py-1">

```

```

        <form method="post" action="vendor/videostats.php">
            <button type="submit" name="<?php echo $btnlikename; ?>"
value="<?php echo $videoId; ?>" style="background-
color:#343A40;color:white;border:0;margin-right:3%;"></button>
            <span style="background-
color:#343A40;color:white;border:0;margin-right:2%;"><?php echo
round($percent); ?>%</span>
            <button type="submit" name="<?php echo $btndislikename;
?>" value="<?php echo $videoId; ?>" style="background-
color:#343A40;color:white;border:0;margin-right:3%;"></button>
        </form>
    </div>
    <div class="col">
        <div class="px-4 py-1">
            <a download="<?php echo $userid['user_video']; ?>"
href="<?php echo $userid['user_video']; ?>" style="font-
size:20px;font-family:Arial,sans-serif;color:#AEADAD;padding-
left:15%;"></a>
            <button class="infobtn" onclick="Click<?php echo
$videoId; ?>()" type="submit" name="infobtn" style="background-
color:#343A40;color:white;border: 0;"></button>
            <button type="submit" name="reportvideo" data-
toggle="modal" data-target="#videoReportModal<?php echo $videoId
?>" style="background-color:#343A40;color:white;border: 0;"></button>
        </div>
    </div>
    <div class="col">
        <div class="px-4 py-1" style="text-align: center;">
            <form method="post"
action="vendor/videodelete.php">
                <span style="margin-right:37px;"><?php echo
$userid['video_date']; ?>
                <?php
                    if ($userid == $videoIduser OR $rank
== $admin) {
                        ?>
                        <button type="submit" name="delete"
value="<?php echo $videoId; ?>" style=";background-
color:#343A40;color:white;border: 0"></button>
    <?php
        }
    <?>
    </span>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
<div id="hb<?php echo $videoId; ?>" style="display: none;">
    <div id="info<?php echo $videoId; ?>" style="padding-right:
10px;">
        <div style="height: 500px; overflow: auto; overflow-x: hidden;
border-bottom: 1px solid white;margin-bottom: 15px;">
            <div class="row">
                <div class="col">
                    <div class="px-6 py-1">
                        <form method="post" action="author.php">
                            <b>Автор:</b>
                            <p><button class="btn btn-secondary"
type="submit" name="page" value="<?php echo $videoIduser; ?>"
style="background-color: #343A40; border:0;padding-left:0;"><?php
echo $author['nickname']; ?></button></p>
                            </form>
                        </div>
                    </div>
                <div class="col">
                    <div class="px-6 py-1">
                        <b>Назва:</b>
                        <p><?php echo $userid['video_name']; ?></p>
                    </div>
                </div>
            </div>
            <div class="row">
                <div class="col">
                    <div class="px-6 py-1">
                        <b>На основі фільму:</b>
                        <p><?php echo $videofilmtag ?></p>
                    </div>
                </div>
                <div class="col">
                    <div class="px-6 py-1">
                        <b>Ключові теги:</b>
                        <p><?php echo $videotargettag ?>, <?php echo
$videotargettag2 ?>, <?php echo $videotargettag3; ?></p>
                    </div>
                </div>
            </div>
            <div class="row">

```

```

<div class="col">
    <div class="px-6 py-1">
        <b>Кількість позитивних оцінок:</b>
        <p><?php echo $likescount; ?></p>
    </div>
</div>
<div class="col">
    <div class="px-6 py-1">
        <b>Кількість негативних оцінок:</b>
        <p><?php echo $dislikescount; ?></p>
    </div>
</div>
</div>
<div class="row">
    <div class="col">
        <div class="px-6 py-1">
            <b>Кількість коментарів:</b>
            <p><?php echo $commentscount; ?></p>
        </div>
    </div>
    <div class="col">
        <div class="px-6 py-1">
            <b>Дата публікації:</b>
            <p><?php echo $userid['video_date']; ?></p>
        </div>
    </div>
</div>
<?php
    if ($iduser == $videoIduser OR $rank == $admin) {
?>
        <b>Кількість підтверджених скарг на відео:</b>
        <p><?php echo $videoreportscount; ?></p>
    <?php
        }
    ?>
    <b>Опис:</b>
    <p><?php echo $userid['video_desc']; ?></p>
    <p style="text-align: center;"><b>Коментарі</b></p>
    <?php
        $commentslist = mysqli_query($connect, "SELECT * FROM
`video_comments` WHERE `id_video` = '$videoId' ORDER BY `id`
DESC");

        while ($commentsvid =
mysqli_fetch_assoc($commentslist)) {
            $commentId = $commentsvid['id'];
            $commentIduser = $commentsvid['id_user'];
            $commentUavatar = mysqli_query($connect, "SELECT
`avatar` FROM `users` WHERE `id` = '$commentIduser'");
            $commentavatar =
mysqli_fetch_assoc($commentUavatar);
            ?>
                <!--Модальне вікно репортів на коментар-->

```

```

        <div class="modal fade"
id="commentReportModal<?php echo $commentId ?>" tabindex="-1"
role="dialog" aria-labelledby="commentReportModal<?php echo
$commentId ?>" aria-hidden="true">
        <div class="modal-dialog modal-lg"
role="document">
                <div class="modal-content bg-plus-col">
                        <div class="modal-header">
                                <h5 class="modal-title"
id="commentReportModal<?php echo $commentId ?>">Скарга на коментар
</h5>
                                <button class="close col-white"
type="button" data-dismiss="modal" aria-label="Close">
                                        <span aria-
hidden="true">&times;</span>
                                </button>
                        </div>
                        <div class="modal-body">
                                <p>Виберіть причину скарги:</p>
                                <form method="post"
action="vendor/reportsend.php" enctype="multipart/form-data">
                                        <select name="reportcause"
class="form-control">
                                                <option value="Інформаційна
пропаганда">Інформаційна пропаганда</option>
                                                <option value="Некорректні
висловлювання в сторону людей/меншинств">Некорректні висловлювання
в сторону людей/меншинств</option>
                                                <option value="Нецензурна
лексика">Нецензурна лексика</option>
                                                <option
value="Погрози/шантаж">погрози/шантаж</option>
                                                <option value="Призов до
шкідливих чи небезпечних дій">Призов до шкідливих чи небезпечних
дій</option>
                                                <option value="Порушення прав
людини">Порушення прав людини</option>
                                                <option value="Пропаганда
ненависті/тероризму/війни">Пропаганда
ненависті/тероризму/війни</option>
                                                <option
value="Спам">Спам</option>
                                        </select>
                                        <br>
                                        <textarea
name="txtareareportcomment" id="txtareareportcomment" cols="70"
rows="10" type="text" class="form-control" placeholder="Опишіть
причину скарги більш детально" required style="background-
color:#AEADAD;color:white;border-radius: 10px; margin-bottom:
10px;"></textarea>
                                        <br>

```

```

        <button type="submit" class="btn
btn-secondary" name="btnreportcomment" value="<?php echo
$commentId; ?>">Відправити</button>
    </form>
</div>
</div>
</div>
</div>
<div style="width:90%;margin-left:5%;border:5px
solid gray;border-bottom-left-radius:40px;border-bottom-right-
radius:40px;background-color:#343A40;color:white;margin-
bottom:30px;">
    <h5 style="text-align:center;width:90%;margin-
left:5%"><?php echo $commentsvid['user_nickname'];
?></h5>
    <p style="margin-left:10px;margin-
right:10px;"><?php echo $commentsvid['comment_text']; ?></p>
    <div class="row">
        <div class="col">
            <div class="px-4 py-1">
                <form method="post"
action="vendor/commentdelete.php">
                    <?php
                        if ($iduser ==
$commentIduser OR $rank == $admin) {
                            ?>
                            <button type="submit"
name="commentdelete" value="<?php echo $commentId; ?>"
style="background-color:#343A40;color:white;border: 0;"></button>
                                <?php
                                    }
                                ?>
                            </form>
                        </div>
                    </div>
                    <div class="col">
                        <div class="px-4 py-1">
                            <span style="padding-left:
10%"><?php echo $commentsvid['comment_date']; ?></span>
                        </div>
                    </div>
                    <div class="col">
                        <div class="px-4 py-1">
                            <button type="submit"
name="reportcomment" data-toggle="modal" data-
target="#commentReportModal<?php echo $commentId ?>"
style="background-color:#343A40;color:white;border: 0;margin-

```



```
    }  
  
    Close<?php echo $videoId; ?> = function() {  
        document.getElementById("hb<?php echo $videoId;  
?>").setAttribute("style","display:none;");  
        $(window).scrollTop($('#focusClose<?php echo $videoId;  
?>').offset().top);  
    }  
  
</script>  
  
<?php  
    }  
>
```


Програмний код файлу film.php

```

<?php

session_start();
require_once 'connect.php';

//Додати новий фільм
if (isset($_POST['btnnewfilm'])) {
    $fname = $_POST['filmname'];
    $fdesc = $_POST['filmdescription'];
    $fdate = $_POST['filmdate'];
    $fcountry = $_POST['filmcountry'];
    $ftag1 = $_POST['filmtag1'];
    $ftag2 = $_POST['filmtag2'];
    $ftag3 = $_POST['filmtag3'];
    $filmtagrating = $_POST['filmtagrating'];
    $filmposter = $_POST['filmposter'];
    $filmvideo = $_POST['filmvideo'];
    $factors = $_POST['filmactors'];
    $ffulldate = $_POST['fulldate'];

    if ($fname == NULL OR $fdesc == NULL) {
        $_SESSION['message'] = "<span style='color:red;font-weight:bold;'>Всі поля повинні бути заповненими!</span>";
        header("Location: ".$_SERVER['HTTP_REFERER']);
    }else{
        $array = explode(" ", $fdesc);
        $success = true;

        foreach ($array as $word) {
            if(strlen($word) > 45) {
                $success = false;
                $_SESSION['message'] = "<span style='color:red;font-weight:bold;'>В звичайній розмовній лексиці слів довше 45-ти символів не буває! :)</span>";
                header("Location: ".$_SERVER['HTTP_REFERER']);
            }
        }

        if($success) {
            if ($_FILES['filmvideo']['size'] > 0 &&
            $_FILES['filmposter']['size'] > 0) {
                $path = 'uploads/films/' .
            $_FILES['filmvideo']['name'];
                if
            (!move_uploaded_file($_FILES['filmvideo']['tmp_name'], '../' .
            $path)) {
                    $_SESSION['message'] = 'Помилка!';
                    header("Location:
            ".$_SERVER['HTTP_REFERER']);
                }
            }
        }
    }
}

```

```

        $poster = 'uploads/films/posters/' .
$_FILES['filmposter']['name'];
        if
(!move_uploaded_file($_FILES['filmposter']['tmp_name'], '../' .
$poster)) {
                $_SESSION['message'] = 'Помилка!';
                header("Location:
".$_SERVER['HTTP_REFERER']);
        }

        mysqli_query($connect, "INSERT INTO `films`
(`film_name`, `film_desc`, `country`, `film_tag_1`, `film_tag_2`,
`film_tag_3`, `film_tag_rating`, `film_poster`, `film_video`,
`film_year`, `film_actors`, `full_date`) VALUES ('$fname',
'$fdesc', '$fcountry', '$ftag1', '$ftag2', '$ftag3',
'$filmtagrating', '$poster', '$path', '$fdate', '$factors',
'$ffulldate')");

        $_SESSION['message'] = "<span
style='color:green;font-weight:bold;'>Фільм опубліковано!</span>";
        header("Location:
".$_SERVER['HTTP_REFERER']);
    }else {
        $_SESSION['message'] = "<span
style='color:red;font-weight:bold;'>Відео або прев'ю не
вибрано!</span>";
        header("Location:
".$_SERVER['HTTP_REFERER']);
    }
}

/*Редагувати фільм (Не зроблено)
}else if (isset($_POST['btnregfilm'])) {
    $id = $_POST['idfilm'];

    mysqli_query($connect, "UPDATE `films` SET ДОПИСАТИ WHERE
id='$id'");*/

//Видалити фільм
}else if (isset($_POST['btndelfilm'])) {
    $id = $_POST['filmselect'];

    mysqli_query($connect, "DELETE from films WHERE
id='$id'");
    //Додати відео-тег
}else if (isset($_POST['btnnewvideotag'])) {
    $newtagname = $_POST['newtagname'];

    mysqli_query($connect, "INSERT INTO `video_tags`
(`tag_name`) VALUES ('$newtagname')");

    //Змінити відео-тег

```

```
}else if (isset($_POST['btnregvideotag'])) {
    $id = $_POST['tagselect'];
    $updatename = $_POST['regtagname'];

    mysqli_query($connect, "UPDATE `video_tags` SET `tag_name`
= '$updatename' WHERE id='$id'");

    //Видалити відео-тег
}else if (isset($_POST['btndelvideotag'])) {
    $id = $_POST['tagselect'];

    mysqli_query($connect, "DELETE from video_tags WHERE
id='$id'");
}

header("Location: ".$_SERVER['HTTP_REFERER']);
?>
```



```

        ?>
            <option value="<?php echo
$filename; ?>"><?php echo $filename; ?></option>
            <?php
                }
            ?>
        </select>
    </div><br>
    <div class="targetTagFilter">
        <label
for="targetTagFilter">Виберіть ключовий тер відео</label>
        <select name="targetTagFilter"
class="form-control" id="targetTagFilter">
            <option value="0">Будь-
який</option>
            <?php
                $vtagsresult =
mysqli_query($connect,"SELECT * FROM `video_tags` ORDER BY
`tag_name`");
                while($vtagslist =
mysqli_fetch_assoc($vtagsresult)){
                    $vtag =
$vtagslist['tag_name'];
                    ?>
                        <option value="<?php echo
$vtag; ?>"><?php echo $vtag; ?></option>
                        <?php
                            }
                        ?>
                    </select>
                </div><br>
                <div class="targetTagFilter3">
                    <label
for="targetTagFilter3">Виберіть вікову категорію відео</label>
                    <select name="targetTagFilter3"
class="form-control" id="targetTagFilter3">
                        <option value="Для всіх">Для
всіх</option>
                        <option value="Для дітей">Для
дітей</option>
                        <option value="Для
дорослих">Для дорослих</option>
                    </select>
                </div><br>
                    <input type="submit" class="btn btn-
primary" name="btnfilter" value="Фільтрувати">
                    <input type="submit" class="btn btn-
warning" name="btnfilterclean" value="Скинути">
                </form>
            </div>
        </div>
    </div>
</div>

```

```

<div id="vidsdiv" style="margin-top:10px;margin-
bottom:30px;">
    <div class="row" style="width:100%;">
        <div style="padding-left:30px; padding-top:10px;">
            <button type="submit" name="btnVideoFilters"
data-toggle="modal" data-target="#videoFiltersModal"
style="background-
color:#343A40;color:white;border:0;width:110px;border-
radius:5px;height:35px;"> Фільтри</button>
        </div>
        <div class="col">
            <div class="px-10 py-1" style="padding-
right:10%;">
                <h2 style="text-align:
center;color:white;opacity:0.7;">Тут Ви можете переглянути
останні опубліковані користувачами відео!</h2>
            </div>
        </div>
    </div>
<?php
    $idUser = $_SESSION['user']['id'];
    $filmnamef = $_POST['videofilmnamefilter'];
    $targettagf = $_POST['targetTagFilter'];
    $targettagf3 = $_POST['targetTagFilter3'];

    if (isset($_POST['btnfilter'])) {
        if ($filmnamef != '0' AND $targettagf == '0') {
            $result = mysqli_query($connect,"SELECT * FROM
`users_videos` WHERE `video_film_tag` = '$filmnamef' AND
`video_target_tag3` = '$targettagf3' ORDER BY `id` DESC");
        }else if ($filmnamef == '0' AND $targettagf !=
'0') {
            $result = mysqli_query($connect,"SELECT * FROM
`users_videos` WHERE `video_target_tag3` = '$targettagf3' AND
(`video_target_tag` = '$targettagf' OR `video_target_tag2` =
'$targettagf') ORDER BY `id` DESC");
        }else if ($filmnamef == '0' AND $targettagf ==
'0') {
            $result = mysqli_query($connect,"SELECT * FROM
`users_videos` WHERE `video_target_tag3` = '$targettagf3'
ORDER BY `id` DESC");
        }else {
            $result = mysqli_query($connect,"SELECT * FROM
`users_videos` WHERE `video_film_tag` = '$filmnamef' AND
(`video_target_tag` = '$targettagf' OR `video_target_tag2` =
'$targettagf') AND `video_target_tag3` = '$targettagf3' ORDER
BY `id` DESC");
        }

        }else if (isset($_POST['btnfilterclean'])) {

```

```
        $result = mysqli_query($connect,"SELECT * FROM
`users_videos` ORDER BY `id` DESC");

        }else{
            $result = mysqli_query($connect,"SELECT * FROM
`users_videos` ORDER BY `id` DESC");
        }
        include('uservideodisplay.php');
    ?>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRk
fj" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/po
pper.min.js" integrity="sha384-
9/reFTGAW83EW2RDu2S0VKAizap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7
lN" crossorigin="anonymous"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/script.js"></script>

</body>
</html>
```