

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проєктування та прикладної математики

(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Проектування вебсервісу ресурсу з використанням мови Python»

ІЛЬЧЕНКО ВЛАДИСЛАВ МИКОЛАЙОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ 2026 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій проєктування та прикладної математики

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТППМ

д.т.н., професор Бородавка Є.В.

“ ” 2026 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Проектування вебсервісу ресурсу з використанням мови Python»

Виконав: студент 4-го курсу, групи ІСТ-22

Спеціальності: 126 «Інформаційні системи та технології»

(шифр і назва спеціальності)

Ільченко В.М.

(прізвище та ініціали)

Керівник доктор філософії Волох Б.Ю.

(прізвище та ініціали)

Рецензент доктор філософії Рябчун Ю.В.

(прізвище та ініціали)

Київ 2026 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій проєктування та ПМ

Освітній рівень: «бакалавр» за ОПП

Спеціальність: 126 «Інформаційні системи та технології»

Освітня програма: «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТППМ
д.т.н., професор Бородавка Є.В.

(підпис)

“ ____ ” _____ 2026р.

**З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Ільченко Владислав Миколайович

1. Тема роботи: “Проектування вебсервісу ресурсу з використанням мови Python”
затверджена наказом ректора КНУБА № 444/52-15/13.6/26 від “08” квітня 2026 року
2. Керівник роботи: Волох Богдан Юрійович, доктор філософії, старший викладач кафедри ІТППМ
3. Строк подання студентом роботи до захисту: червень 2026 року
4. Зміст пояснювальної записки за розділами:
 - Р.1. Аналітичний огляд та постановка задачі
 - Р.2. Інформаційне та математичне забезпечення
 - Р.3. Розробка програмного забезпечення
5. Інформаційні слайди:
 - С.1. Титульний слайд
 - С.2. Побудова концепції сайту
 - С.3. Логічне моделювання
 - С.4. Архітектура БД сайту
 - С.5. Backend частина розробки сайту
 - С.6. Frontend частина розробки сайту
 - С.7. Висновки

6. Календарний план виконання кваліфікаційної роботи

Види робіт та їх зміст	Дата виконання
Р.1. Аналітичний огляд та постановка задачі	Лютий 2026 р.
Р.2. Інформаційне та математичне забезпечення	Березень 2026 р.
Р.3. Розробка програмного забезпечення	Травень 2026 р.
Остаточне оформлення роботи	Травень 2026 р.
Направлення роботи на рецензування	травень 2026 р.
Попередній захист роботи на кафедрі	червень 2026 р.

7. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	Дата	Підпис
Прийом програмного продукту	Доктор філософії, доцент Рябчун Ю.В.		

8. Дата видачі завдання: 12 січня 2026 року

Керівник

Волох Б.Ю.

 (підпис) (прізвище та ініціали)

Бакалавр

Ільченко В.М.

 (підпис) (прізвище та ініціали)

АНОТАЦІЯ

Проектування вебсервісу ресурсу з використанням мови Python

Кваліфікаційна робота бакалавра за спеціальністю: 126 «Інформаційні системи та технології», освітня програма: «Інформаційні системи та технології». – Київський національний університет будівництва та архітектури. – Київ, 2026.

Вибір теми проекту обґрунтований потребами користувачів в наявності можливості ознайомлення з будь-яким продуктом в електронному вигляді.

У результаті роботи розроблено принцип побудови web-ресурсу та структуру бази даних, з використанням мови програмування python.

Показаний приклад реалізації даного продукту на основі Python-фреймворку для розробки веб-систем – Django .

Ключові слова: сайт, база даних, Django.

SUMMARY

"Project of WEB-resource service based on python".

Bachelor's qualification work in the specialty: 126 "Information systems and technologies", educational program: "Information systems and technologies". - Kyiv National University of Construction and Architecture. - Kyiv, 2026.

The choice of the project theme is justified by the needs of users in the availability of the opportunity to get acquainted with any product in electronic form.

As a result of work on the diploma project the principle of construction of a web-resource and structure of a database, with use of python programming language is developed.

The example of realization of this product on the basis of Python-framework for development of web systems - Django is shown.

Key words: site, database, Django.

ЗМІСТ

Вступ	7
1.АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАДАЧІ	8
1.1 Аналітичний огляд	8
1.2 Дерево цілей та дерево задач	20
1.3 Постановка задачі	21
2.ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	23
2.1 Аналіз предметної області	25
2.2 Побудова концепції сайту	27
2.3 Логічне моделювання	31
2.3.1 Діаграма варіантів використання сайту	33
2.3.2 Діаграма класів сайту	35
2.4 Архітектура БД сайту	37
2.5 Інструменти розробки	39
3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	41
3.1 Загальний опис та файлова структура сайту	42
3.2 Backend частина розробки сайту	44
3.2.1 Додаток main	49
3.2.2 Додаток accounts	49
3.3 Frontend частина розробки сайту	52
3.3.1 Додаток main	54
3.3.2 Додаток accounts	54
3.4 Приклад роботи сайту	57
ВИСНОВКИ	59
СПИСОК ДЖЕРЕЛ	60

ВСТУП

Оскільки в наш час з'явилося безліч різних видів послуг, сервісів, брендів та продуктів споживання, що потребують оцінки, то найкращий спосіб отримати чесну оцінку - це зібрати статистику з споживачів цих самих сервісів, або продуктів за рахунок їх відгуків, бо саме вони як споживачі цих самих послуг та продуктів зможуть дати свою персональну, прозору оцінку.

Метою та необхідністю розробки створення даного веб-ресурсу для відгуків є збирання статистики, щоб у разі якщо сервіс, послуга, продукт не є тим чим вони себе позиціонують, не надають гідного результату, або взагалі не справляють позитивного враження, то будь який зареєстрований користувач даного веб-ресурсу, що використав дані сервіси, послуги, продукти могли попередити про їхню некомпетентність інших людей, які ще не були впевнені у використанні даних об'єктів споживання. Водночас, якщо сервіс, послуга або продукт виправдовують себе, то користувачі даного сайту зможуть надати позитивний відгук або рецензію, що надалі може надати цим самим раніше перерахованим речам нових клієнтів, що призводить до позитивного кругообігу.

Для досягнення цієї мети необхідно розробити платформу, де кожний зареєстрований користувач може створити свій особистий прозорий, чесний, справедливий та не рекламний відгук про будь який сервіс, а також про музикальні композиції, книжки, фільми, комп'ютерні ігри та інші схожого роду речі.

Також за допомогою цієї статистики можна буде зробити аналіз помилок у разі великої кількості негативних відгуків та рецензій, або аналіз позитивних оцінок, щоб знайти речі на яких можна зосередитися для подальшого покращення результату.

1. АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналітичний огляд

Основні відомості про відгуки

Відгук – це думка про когось або щось, оцінка кого-чого-сь.

Основні відомості про рецензії

Рецензія (нім. Rezension, лат. Recensio, англ. Review – огляд, оцінка) – аналіз, розбір, деяка оцінка публікації, твору або продукту, жанр газетно-журнальної публіцистики та літературної критики. Рецензія може відноситися до матеріальних речей (прилади, аксесуари, побутова техніка), комп'ютерних технологій, художньої літератури, музики, фільмів, комп'ютерних ігор. Рецензувати можуть також поточні події, громадські заяви і події. На додаток до критичного твердження, автор рецензії може виставити предмету рецензування певну оцінку для визначення відносної цінності рецензованого предмета.

Види рецензій

Споживча рецензія (англ. Consumerreview) – це рецензія, написана власником продукту або послуги, або ж написана тими користувачами продукту або послуги, які мають достатній досвід для того, щоб прокоментувати надійність товару / послуги та його відповідність заявленим виробником специфікаціям.

Експертна рецензія (англ. Expertreview) – це рецензія, написана будь-ким, хто протестував кілька продуктів або послуг для визначення, який з них пропонує найкраще співвідношення ціни і якості або найкращий набір особливостей.

Куплена рецензія (англ. Boughtreview) – це система, де творець (зазвичай компанія) нового продукту платить рецензенту, щоб він прорецензував даний продукт. Перш за все використовується в автомобільній індустрії, фільмах і комп'ютерних іграх, ця система створює свого роду таємну рекламу. Як правило, куплені огляди є упередженими, хоча бувають і винятки.

Авторерецензія – оцінка автором власного твору; жанр автокритики, що публікується під псевдонімом чи криптонімом.

Приблизна структура рецензії:

- 1) Тема або найменування рецензованої роботи;
- 2) Її автор;
- 3) Актуальність і правильність обраної теми;
- 4) Вдалість огляду предмету;
- 5) Використання системи доказів;
- 6) Повнота розкриття проблеми;
- 7) Результати аналізу економічної доцільності, якщо необхідні;
- 8) Наявність чітких висновків;
- 9) Використання наукового апарату;
- 10) Якість оформлення роботи;
- 11) Недоліки, що є в роботі;
- 12) Висновок про можливість допуску роботи до захисту (або надання іншого права).

Основні відомості про веб-сайти

Веб-сайт – це сукупність логічно зв'язаної гіпертекстової інформації, оформленої у вигляді окремих сторінок і доступної в мережі Інтернет.

Подібне визначення веб-сайту було правильним на початку існування Інтернету, коли Мережа і веб-сайти використовувалися в основному як розважальна система. До кінця 90-х років веб-сайти дійсно були в основному статичними сторінками. Для створення веб-сайту було потрібне лише знання мови гіпертекстової розмітки – HTML. Якщо ж сторінка надавала якісь програмні засоби – це були виключно засоби, що міг надати сервер, на якому розташований веб-сайт. Про зручність і красу тогочасних веб-сайтів взагалі особливо не доводилося говорити. Час спливає, розвиваються мови програмування, розширюються канали передачі інформації. Зараз Інтернет вже є самодостатньою галуззю економіки, а веб-сайти стали повноправними представництвами фірм в Інтернеті.

Веб-сайт — сучасне визначення

Веб-сайт — це сукупність вебсторінок, доступних у мережі Інтернет, які об'єднані як за змістом, так і за навігацією під єдиним доменним ім'ям. Фізично сайт може розміщуватися як на одному, так і на кількох серверах.

Завдання сучасного веб-сайту

Веб-сайт виконує такі основні завдання:

- а. реклама продукції, послуг, ідей. Правильно зроблений веб-сайт із легкістю приведе клієнта до висновку про необхідність покупки товару, або послуг, або ідей, що пропагуються на ньому;
- б. продаж товарів, послуг, інформації, ідей. У сучасної людини немає багато часу для ходіння по магазинах. Тому можливість замовлення товарів і послуг, не відходячи від комп'ютера, значно розширює можливості і клієнта, і продавця;
- в. безкоштовне надання інформації або послуг. Насправді надання інформації або послуг — це засіб залучення відвідувачів до даного ресурсу для здобуття, наприклад, статистичної інформації або ж для показу реклами, якщо це рекламний майданчик;
- г. підтримка клієнтів.

Класифікація веб-сайтів

Сайт візитка

Сайт візитка – зазвичай це сайт з кількох сторінок, на якому міститься інформація про діяльність фірми чи приватної особи, послуги і контактні дані для потенційних клієнтів. Також за бажанням можна розмістити історію фірми, види діяльності, заслуги, фотографії тощо. Такий сайт являється повноцінним віртуальним ресурсом з унікальним дизайном та оформленням.

Корпоративний сайт

Корпоративний сайт є повноцінним представленням Вашої компанії в інтернеті. Це обов'язковий інструмент що допомагає забезпечити успішне ведення бізнесу будь-якої компанії. Мета створення корпоративного сайту – довести інформацію про компанію, її діяльність, товари або послуги, до відома якомога більшої кількості людей, в яких зацікавлена компанія як в майбутніх споживачах або партнерах.

Сайт каталог (вітрина)

Призначений для продажу товарів чи групи товарів. На таких сторінках потрібно розмістити всю інформацію, необхідну для того, щоб клієнт побачив, зацікавився і купив даний товар. Це структурований каталог продукції Вашої компанії. На таких сайтах розміщується інформація про товари/послуги, яку неможливо помістити в прайс-лист, детальний опис, фото, ціна.

Сайти електронної комерції (інтернет магазини)

На відміну від сайту-вітрини, де клієнт може переглянути наявність всіх товарів, за допомогою інтернет-магазину клієнт може ще й зробити замовлення, вибрати варіант розрахунку, спосіб отримання замовлення та одержати рахунок на оплату. І все це не відходячи від власного комп'ютера.

Інформаційні портали та сервіси

Інформаційний портал – це веб-сайт, який об'єднує в собі безліч різних сервісів, і надає користувачеві мережі Інтернет доступ до них відповідно обраного

тематичного розділу порталу, об'ємний, він має велику кількість інформації, і може запропонувати велику кількість послуг. Інформаційний портал є засіб збору, обробки і розповсюдження інформації з метою рішення конкретних завдань. Тут користувач може зареєструватися і отримати доступ до всіх послуг цього вузла — електронної пошти, блогів, пошуку роботи, служби оголошень, погоди, новин, гороскопів, а також, відділу торгівлі, подорожей, відсилення SMS та багато іншого.

Промо-сайт

Такий сайт це насамперед рекламний інструмент, для реклами конкретного товару чи послуги чи навіть якогось заходу. На промо-сайтах розміщується вичерпна інформація про бренд, різні рекламні акції (конкурси, вікторини, ігри тощо). Не великий за об'ємом текстового наповнення, повинен володіти яскравим, незабутнім, стильним дизайном, що поєднує велику кількість графіки і здатним ефектно проінформувати користувача про товар чи послугу.

Сайт-брошура

Сайти-брошури є найпоширенішими, вони містять практично ті ж матеріали, що й традиційні рекламні брошури. Їхньою метою є рекламування послуг та товарів певної компанії.

Особистий сайт

Назва говорить сама за себе – особистий сайт. Це може бути що завгодно. Розкажи про себе цілому світові, пиши свої статті, історії, розміщуй фото, відео тощо. Часто створюють особистий сайт з метою ведення блогу, інформування про свою подорож онлайн, презентацію власних робіт тощо. Інколи особисті сайти створюються без попередньо визначеної мети, просто так, для себе, а з часом, при наповненні контентом, можуть перетворитися в потужний інтернет ресурс.

*Блог – це веб-сайт, головний зміст якого – записи, зображення чи мультимедіа, що регулярно додаються. Для блогів характерні короткі записи тимчасової значущості.

Інші веб-сайти

Це найбільш поширені категорії інтернет сторінок (веб-сторінок) які найчастіше створюються на основі заготовок, шаблонів. Звичайно існує багато інших видів сторінок, таких як розважальних, інформаційних, всілякі форуми і інші.

Технологія відображення ресурсу:

Статичні сайти – складаються з Html файлів і передаються користувачеві у вигляді, в якому зберігаються на сервері

Динамічні сайти – як правило, працюють на основі системи управління сайтом (CMS). Вміст генерується, за допомогою різних скриптів використовуючи базу даних

Сайт візитка – представництво компанії в мережі Інтернет. Надає загальну інформацію про власника (компанія, підприємець і тд). Як правило, складається з 5-7 сторінок, при цьому максимально описує вид діяльності. Головні розділи - про компанію, прайс-листи, контактні дані, схема проїзду.

Бізнес сайт – це каталог продукції компанії (товари або послуги). У каталозі присутній докладний опис товару, публікуються сертифікати й нагороди, відгуки клієнтів і звичайно технічні дані. Сайт даної специфікації незамінний для будь-якої компанії.

Промо сайт–веб сайт конкретної торгової марки (бренду). Створюється з метою реклами нового продукту або бренду в цілому. Розділи: акції, вікторини, голосування, ігри і тд.

Інтернет-магазин– це каталог товарів з можливістю онлайн замовлення. Клієнт має можливість оплати товару, за допомогою різних сервісів використовуючи при цьому пластикову карту, рахунок або Інтернет гроші. Розділи: про магазин, допомога, каталог, кошик, сторінка замовлення, контакти.

Корпоративний сайт – містить повну інформацію про компанію (послуги, товари і тд.). Як правило, має складну структуру і має розширену

функціональність. Може бути відкритий для відвідувача або призначений для внутрішнього використання (зв'язок, документообіг і тд.).

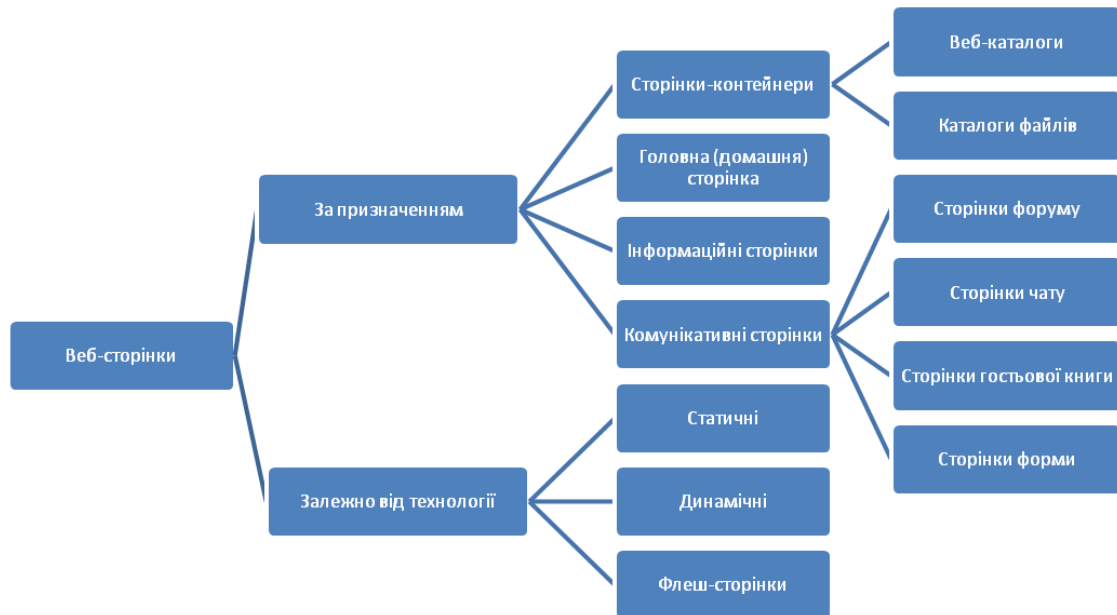


Рисунок 1.1 Класифікація веб-сайтів

Створення сайтів: методи

Творці сайтів, беручись за розробку веб-проекту, як правило, переслідують декілька цілей, які досягаються вже перевіреними та новими методами.

Представляючи свій веб-ресурс, треба прагнути підкреслити свою інформативність та прозорість інформації. На кваліфіковано спроектованому веб-сайті розробник може вигідно піднести інформацію сам, або надати потрібні для піднесення корисної інформації інструменти користувачам.

Також є можливість надати це все в зручному для користувачів форматі (наприклад у вигляді каталогу). Розробник має можливість створити зручний формат подачі інформації та представити її таким чином, щоб у користувача після відвідування інтернет-ресурсу з'явилося певне враження про інформацію яку він отримав, а також можлива думка про подальше відвідування даного інтернет-ресурсу.

Більшість веб-ресурсів створюються з метою поширення певної інформації. Мета таких ресурсів полягає в інформуванні якомога більшої кількості відвідувачів та користувачів даних сайтів. Саме за допомогою Інтернету інформація розлітається з величезною швидкістю, дозволяючи тим самим з часом

збільшити трафік відвідувань. Користувач зазвичай має можливість самостійно отримати потрібну інформацію про продукт, сервіс, або послугу, почитати опис, зрівняти ціни та інші фактори порівнянь.

Фактори успішних веб-сайтів

Будь-який сайт повинен мати певні грамотні складові, для того щоб бути успішним і мати можливість конкурувати у своїй галузі. Ось кілька з правил для створення дійсно якісного веб-проекту незалежно від його спрямованості:

1. Дизайн
2. Контент
3. Унікальний, справжній трафік

Дизайн – це перший пункт, який потрібно виконати для успішного конкурування сайту. Зовнішнє оформлення повинно бути привабливим, щоб залучити і користувача в споживанні інформації яка надається сайтом. Дизайнер-розробник в ідеалі повинен не тільки дати сайту життя, але й провести необхідні тестування і доопрацювання з поправкою на думку споживачів. Не можна кидати роботу над оформленням, роботу потрібно підтримувати до досконалості як тільки цього потребує споживач.

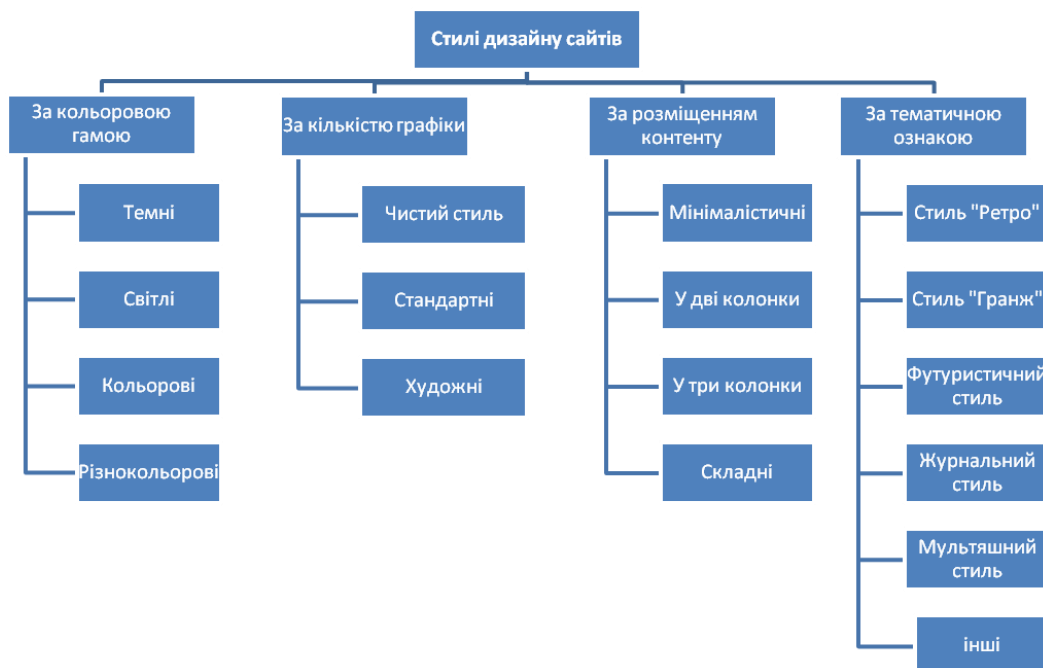


Рисунок 1.2 Стили дизайну сайтів

Контент – інформаційна наповнюваність ресурсу також є важливим фактором його популярності. Хорошим варіантом буде початкове створення інформаційного простору та надання інструментів та інтерфейсу для його покращення, а вже потім художнє оформлення сайту. Таким чином, вийде повна взаємодія вербальної і візуальної складових.

Унікальний, справжній трафік – попадання користувача на певний інтернет-ресурс здійснюється за допомогою пошуковиків. Такий спосіб найбільш цільовий, оскільки користувач у пошуках потрібної інформації потрапляє на потрібне джерело. Таким чином користувач свідомо має інтерес в потрібній йому інформації. Подальшу завдання веб-ресурсу утримати інтерес відвідувача і мотивувати його до подальшого використання сервісу чи ресурсу. Ефективними методиками залучення трафіку відвідувань на певні сайти є просування сайту у мережі Інтернет. Не варто забувати про них, оскільки ці методики забезпечують повноцінну роботу будь-якого веб-сайту.

Вибір інструментів для розробки

Незалежно від обраної архітектури в кожному проекті з розробки додатків потрібно вибирати якісь технології, мови програмування і інструменти, за допомогою яких будуть створюватися ці додатки. При цьому потрібно враховувати, що вибір може обмежуватися вимогами замовників.

Самостійна розробка

Самостійна розробка сайту потребує спеціалізованих знань таких як гіпертекстова розмітка HTML, каскадні стилі CSS, мова керування сценаріями JavaScript, а також знання програмно-апаратної частини (Php, Python, Ruby). Ще один з недоліків є час розробки – бо треба писатисамостійно не тількинаписатифункціонал, а й розробити дизайн, зверстати.

З іншого боку вимаєте контроль на всіхетапахвиробництва: створеннявласного дизайну якийзадовольнитьвсіваші потреби, додаваннясвоїфункції і щебагаточого.

Front-end розробка

HTML. HTML (HyperTextMarkupLanguage) - стандартна мова розмітки гіпертекстових сторінок в Інтернеті. Є й інші мови розмітки гіпертексту, але велика частина сторінок сайтів Інтернету розмічена саме на мові HTML. Такі сторінки успішно інтерпретуються браузером, які відображають їх на екранах різних електронних пристроїв в зручному для людини вигляді.

При роботі з HTML ми використовуємо прості структури коду (теги і атрибути), щоб розмітити сторінку веб-сайту.

HTML документ починається з `<!DOCTYPE html>` - кожна веб сторінка повинна починатися саме з нього. Раніше була величезна кількість варіантів тега DOCTYPE, але відповідно до новою версією HTML, потрібен тільки цей `!DOCTYPE`, який в прикладі. Що таке DOCTYPE? Це тег, який повідомляє браузеру версію HTML. Цей тег повинен бути написаний на першій сходинці HTML документа.

Після йде парний тег `<html></html>`. Цей тег є контейнером веб сторінки, тобто до нього і після нього нічого не повинно бути.

У парному тезі `<head></head>` пишеться обов'язковий тег `<title></title>`. У тезі тайтл пишеться заголовок сторінки, який відображається у видачі пошукової системи і у вкладці браузера. У контейнері `<head></head>` зазвичай знаходяться різні мета-теги і підключаються стилі і скрипти, більше він не для чого не потрібен.

Весь текст, графіка та інший код повинен знаходитися усередині тегів `<body></body>`. В даному прикладі HTML документа створений заголовок першого рівня і абзац.

HTML5 залишився мовою розмітки, але відрізняється від попередньої версії досить серйозно. Якщо минула четверта версія була тільки мовою розмітки, то нова версія містить мову розмітки, засоби програмування CANVAS на базі JavaScript, розширені можливості роботи з медіа, розширені елементи форм. Дана мова взагалі називають мовою семантичної розмітки через чіткої структури

представлення елементів сайту, так як в даній версії замість єдиного тега визначає блок, з'явилася можливість розбиття статті на логічні області за допомогою нових тегів <HEADER><BODY><FOOTER> і семантичного розбиття <SECTION><ARTICLE><DIV><ASIDE><NAV>

Починаючи з 2010 року в якості стандартів була запропонована наступна зв'язка HTML5 + CSS3 + JavaScript.

CSS. CSS - це формальна мова, службовець для опису оформлення зовнішнього вигляду документа, створеного з використанням мови розмітки (HTML, XHTML, XML). Назва походить від англійського CascadingStyleSheets, що означає «каскадні таблиці стилів».

Призначення CSS - відокремлювати те, що задає зовнішній вигляд сторінки, від її змісту. Якщо документ створено тільки з використанням HTML, то в ньому визначається не тільки кожен елемент, але і спосіб його відображення (колір, шрифт, положення блоку і т. Д.). Якщо ж підключені каскадні таблиці стилів, то HTML описує тільки черговість об'єктів. А за все їх властивості відповідає CSS. В HTML досить прописувати клас, не перераховуючи всі стилі кожен раз.

CSS можна охарактеризувати простими словами як набір правил, що описують, як повинен виглядати елемент.

CSS3 - це новий стандарт оформлення HTML документів значно розширює можливості попереднього стандарту CSS2.

Back-end розробка

У наш час існує дуже багато різних мов створення Back-end'a. Найпопулярніші це:

1. Java один з найпопулярніших мов для розробки back-end'a, за допомогою Java можна створювати великі веб-додатки для широкого використання. Але не дивлячись на високий функціонал мови, він працює повільніше подібних йому мов.

2. JavaScript зараз став дуже популярним мовою для створення веб-сайтів і веб-додатків. В основному JavaScript використовується для створення Front-

Endчастини, але за допомогою таких фреймворків як Nodejs, Meteorjs, на ньому можна створювати і back-end частини програми.

3. Asp.net це Framework мови c # для створення back і frontend частин веб-додатки. c # дуже схожий з Java. Це теж багатофункціональна технологія і працює трохи швидше ніж Java. Але для роботи з цією технологією додатково потрібно знати EntityFramework для того щоб не писати величезні рядки коду при роботі з базами даних

4. Php це основна мова для роботи з системами таких як WordPress. Це така система для управління контентом, які в майбутньому можна розширювати. Щоб використовувати цю технологію потрібно на додаток знати html, css, javascript.

5. Дуже популярна мова Python, переваги мови в тому що у нього прекрасна читаність коду і Python вимагає значно менше коду для виконання ніж будь-якої іншої мови програмування.

6. Ruby він дуже схожий на Python, але у нього не така хороша читаність коду. Основні переваги мови це швидкість. Цією мовою написані дуже багато популярних веб-додатків.

Мови JavaScript, Php, Python і Ruby є скриптовими мовами, а решта об'єктно орієнтовані. Скриптові мови (сценарії) опис дію їх системою та вони мають справу з готовими програмними компонентами. А в основі об'єктно орієнтованих мов лежить сукупність об'єктів, які є екземплярами класів, а класи утворюють ієрархію спадкування.

Сучасні динамічні сайти обов'язково використовують базу даних як для зберігання різного контенту, так і для реалізації своїх функцій. Найбільш поширеними системами управління базами даних є MySQL, PostgreSQL, ORACLE. Ці бази даних є полегшеними, тобто мають урізаним функціоналом, достатнім для задач веб розробки. Найбільш поширеною є MySQL. Дана база досить проста в використанні і має високу швидкість обробки запитів, що важливо при великій кількості звернень від користувачів до сервера, на якому розташовується сайт.

1.2Дерево цілей та дерево задач

Головна ціль – розробка веб-ресурсудля відгуківта рецензій.

Перша функція –надати інструменти для створення персоналізованих відгуківта рецензій користувачами даного веб-сайту.

Користувач зможе написати відгук або рецензію в усіх подробицях, а такожнадатиоцінку за шкалою выд 0 до 10, що допоможе у чіткому описі та оцінці сервісу, послуги, бренду або продукту. Таке середовище максимізує ефективність написання відгуківта рецензій,та покращить ефективність збору статистики. Всі відгукита рецензій будуть зберігатися у базі даних.

Друга функція – формування функціїможливості зчитування відгуківта рецензій. Будь-який користувач (навіть не зареєстрований) в будь який час зможе отримати дані про відгук або рецензію. Така система допоможе відвідувачам даного веб-ресурсу знайти потрібну для себе корисну інформацію.

Третя функція – створення системи пошуку. Моніторинг результату допоможе відвідувачу або користувачузнайти потрібні типи відгуків або рецензій з бази даних.

Четверта функція – створення графічної частини сайту. Підібрати найоптимальніший дизайн для сайту, щоб залучити і користувача в споживанні інформації яка надається сайтом.

П'ята функція – створення зручної системи модерації сайту, щоб ручна модераціямогла додавати нові розділи або категорії, а також могли перевірити відгук або рецензію на нормативність, та у разі знаходженні помилок оформлення, або підтвердженні підозрілості відгуку або рецензіїмогли видалити або відредагувати її за необхідності.

На рис. 1.3 зображено ієрархічне дерево цілей веб-ресурсу для відгуків та рецензій.

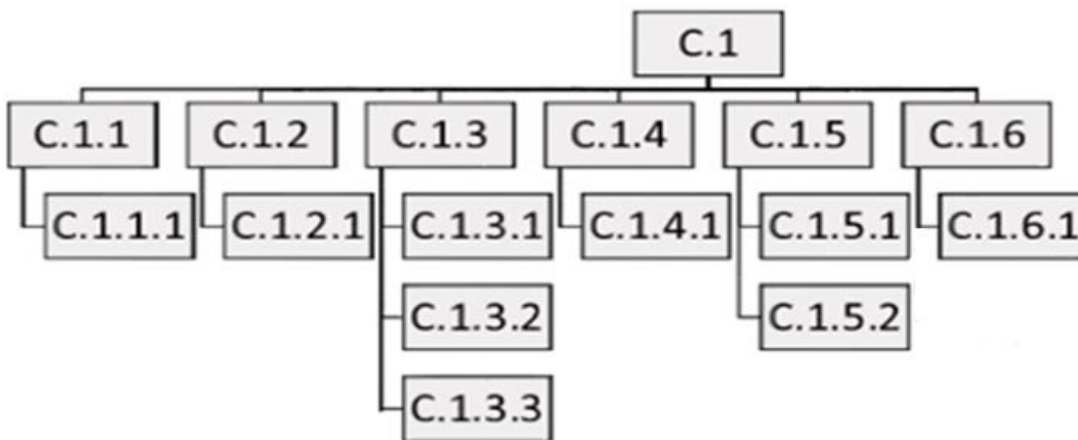


Рисунок 1.3 Ієрархічне дерево цілей веб-ресурсу для відгуків

С.1 Розробка сайту для створення відгуків та рецензій.

С.1.1 Розробити і проаналізувати ідею.

С.1.1.1 Сформулювати загальну концепцію сайту.

С.1.2 Сформувати інтерфейс сайту.

С.1.2.1 Підібрати якісний графічний стиль для сайту.

С.1.3 Створити програму частину сайту.

С.1.3.1 Розробити алгоритм роботи.

С.1.3.2 Створити програму.

С.1.3.3 Протестувати програму.

С.1.4 Налаштувати роботу щодо підтримки сайту.

С.1.4.1 Оновлювати бази даних на сервері.

С.1.5 Сформувати статистику відгуків та рецензій.

С.1.5.1 Фіксація кожного відгуку та рецензії.

С.1.5.2 Створення статистики відгуків та рецензій.

С.1.6 Проаналізувати діяльність сайту.

С.1.6.1 Дослідити загальну оцінку сервісів, послуг, продуктів та т.д.

1.3 Постановка задачі

Оскільки метою даного проекту є розробка платформи, де кожний зареєстрований користувач може створити свій особистий відгук або рецензію про будь який сервіс, а також про музикальні альбоми, книжки, фільми, комп'ютерні ігри та інші схожого роду речі, то проводиться подальша постановка задачі.

Опис постановки задачі (комплексу задач)

Проект створюється для:

- збирання статистики за рахунок відгуків та рецензій;
- ведення історії відгуків та рецензій;
- створення зручної інтернет платформи для відгуків та рецензій;

Вимоги до додатку:

- веб-ресурс повинен бути зручним та простим у використанні;
- веб-ресурс повинен правильно функціонувати, тобто виконувати ті функції, що передбачено при розробці;
- веб-ресурс повинен спрощувати пошук відгуків та рецензій в інтернет-просторі;

2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз предметної області

В наш час з'явилося безліч різних родів послуг, сервісів та продуктів споживання, що потребують оцінки, то найкращий спосіб отримати чесну оцінку - це зібрати статистику з споживачів цих самих сервісів, або продуктів за рахунок їх відгуків та рецензій, бо саме вони як споживачі цих самих послуг та продуктів зможуть дати свою персональну, прозору оцінку.

В нашому випадку наш сайт є інформаційним сайтом, оскільки на сайті можна знайти потрібну користувачу інформацію.

Основною діяльністю даного сайту є розповсюдження інформації.

Залежно від тематики розрізняють:

1) Портали вузької спрямованості, де детально висвітлюється певна тема. Приклад: сайт по ремонту комп'ютерів або електронні ЗМІ. Успішні та престижні інформаційні сайти, дотримуються трьох правил при створенні продукту: якість, достовірність і повнота інформації.

2) Портали широкої спрямованості, що охоплюють популярні теми або події. Приклад: сайт про музику, де існує навігаційна система, пов'язана глобальним тегом. Тут можуть бути присутніми і розважальні канали, тому в цьому випадку дотримуватися трьох вище описаних правил необов'язково.

Залежно від специфіки інформаційні сайту можуть відрізнятися за типами подачі, які можна розглянути на прикладі популярних соціальних мереж. Існує сайти, які об'єднують нижче викладені способи подачі або використовують інші майданчики (портали, сайти, форуми і т.д.) для їх реалізації:

1) Текстові: Facebook, Twitter, LiveJournal.

2) Відео або аудіо: YouTube, Вконтакте.

3) Графічні Instagram.

Основна класифікація інформаційних сайтів

1) Довідники, словники та електронні енциклопедії

Даний тип сайтів - це канонічна форма інформаційних ресурсів. В основному розміщуються дані, що носять науковий або публіцистичний характер. Подібні ресурси бувають безкоштовні та платні, де за доступ до певної інформації доведеться платити.

2) Електронні ЗМІ

Відносно новий вид ЗМІ, який поширюється через глобальну павутину. Більшість великих телеканалів і друкованих видань мають електронні версії новин власного виробництва. Разом з тим існують і справжні інтернет ЗМІ, де мережа є основним джерелом подачі інформації. Популярність електронних ЗМІ обумовлюється зменшенням деяких статей фінансових витрат (оренда офісу, робочий персонал, обладнання тощо). Під статтю інтернет засобів масової інформації можуть потрапляти і персональні блоги.

3) Форуми, де люди відповідають на життєві питання

Епоха інтернет технологій з одного боку зробили людей самотніми, а з іншого надала можливість людям з різних куточків планети ділитися враженнями, давати поради або отримувати відповіді на питання. Одним з таких типів сайтів, який зможе відповісти на багато життєві питання, це веб-портали, що відкриває можливість отримувати відповіді на питання різної спрямованості. Головною перевагою таких ресурсів є широка цільова аудиторія, яка дає поради на основі життєвого чи професійного досвіду користувачів. Найчастіше на питання відповідає не один, а кілька користувачів внаслідок чого створюється ціла дискусія, де правильну відповідь - справа часу.

4) Дошки оголошень

Ринок вторинних товарів живе завдяки електронним дошкам оголошень. Реалії пострадянського простору не дозволяє жителям цих країн постійно купувати імпорتنі товари в «фірмовою» упаковці. Такі портали дозволяють заощадити гроші на послугах посередників, а при налаштуванні спеціальних критеріїв знайти необхідний товар в пішої доступності.

5) Соціальні мережі

Спочатку створювалися для спілкування між людьми. У зростання популярності перетворилися в рекламну середу для просування окремих особистостей, товарів і послуг. Сьогодні соціальні - безкоштовний ресурс, де розміщується інформація різної спрямованості: комерційні пропозиції, твори мистецтва, оголошення, наукові статті, політична агітація та інша інформація.

Описаними видами ресурсів користувався практично кожен користувач. Завдяки таким сайтам, відкривається можливість знайти будь-яку необхідну інформацію пізнавального, розважального або комерційного характеру.

Аналіз сайту з допомогою SWOT

SWOT аналіз є одним з найбільш актуальних ефективних і інструментів маркетингу і менеджменту. Даний інструмент дає чітке уявлення про слабких і сильних сторонах об'єкта дослідження (товару, фірми або окремого бізнес-процесу), дозволяє виявити можливості мінімізації ризиків і загроз за допомогою його сильних сторін, а також вказує вірні напрямки подальшого розвитку об'єкта.

Цей вид досліджень застосовується і в сфері веб-ресурсів. Так, SWOT аналіз сайту є основою для подальшої розробки маркетингової стратегії розвитку і просування веб-ресурсу, а також базою при побудові гіпотез перевірок.

Аналіз сайту полягає в обробці та аналізі всієї доступної маркетингової інформації, за допомогою якої виявляються причини ефективного або неефективного функціонування ресурсу, визначаються пріоритетні напрямки розвитку.

Класичний SWOT - аналіз сайту ставить перед собою 4 ключових мети: виявити сильні сторони ресурсу і конкурентів; слабкі сторони; потенційні зовнішні загрози і сприятливі можливості для розвитку проекту (з урахуванням дослідження конкурентів).

S – Переваги

- 1) Відповідний, з точки зору технічних вимог, для просування сайт
- 2) За можливості може бути велика кількість представленої на сайті інформації
- 3) Широкий спектр напрямків для збору інформації

W – Недоліки

- 1) Низька «пошуковість» сайту через відсутність досвіду в галузі пошукової оптимізації

O – Можливості

- 1) Оптимізація надасть можливість зайняття високих позицій по низькочастотних и високочастотних запитах
- 2) Є можливість використання контекстної реклами для залучення відвідувачів

T – Загрози

- 1) Конкуренти вже мають високі позиції в пошуковій видачі
- 2) Конкурент може збільшити активність в пошуковому просуванні

2.2 Побудова концепції сайту

Виділимо сторінки які нам необхідні використовуючи прикладом Metacritic. Головна сторінка, це перше, що бачить користувач. Заходячи на головну сторінку, ми бачимо:

- навігація по каталогу за допомогою їх категорій
- нові надходження
- пошук
- невеликі відомості про популярні або нещодавні надходження
- популярні або нові категорії
- досить докладна інформація про сайт

Огляд постів (відгуків) можливий не тільки для зареєстрованих користувачів, але і для звичайних відвідувачів.

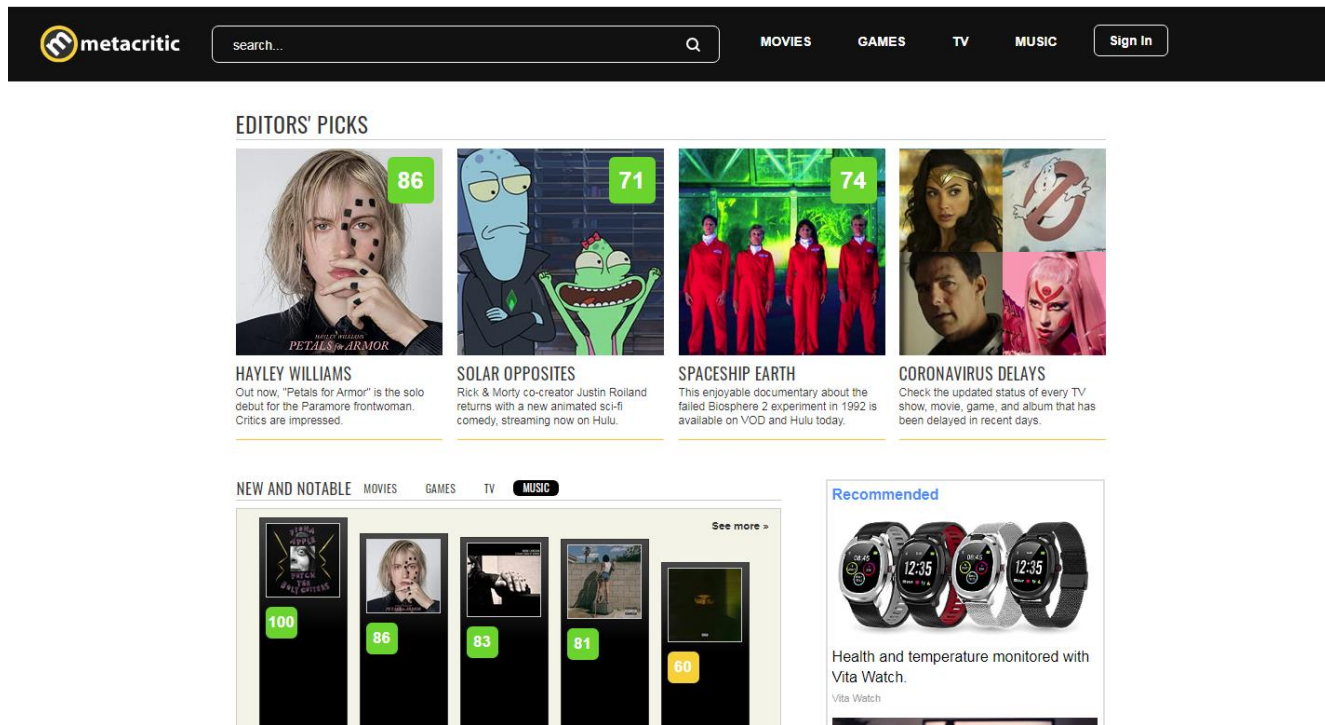


Рисунок 2.1 Головна сторінка

Далі іде сторінка продукту споживання, де відвідувач зможе побачити фото або опис, характеристики, рейтинг, відгук.

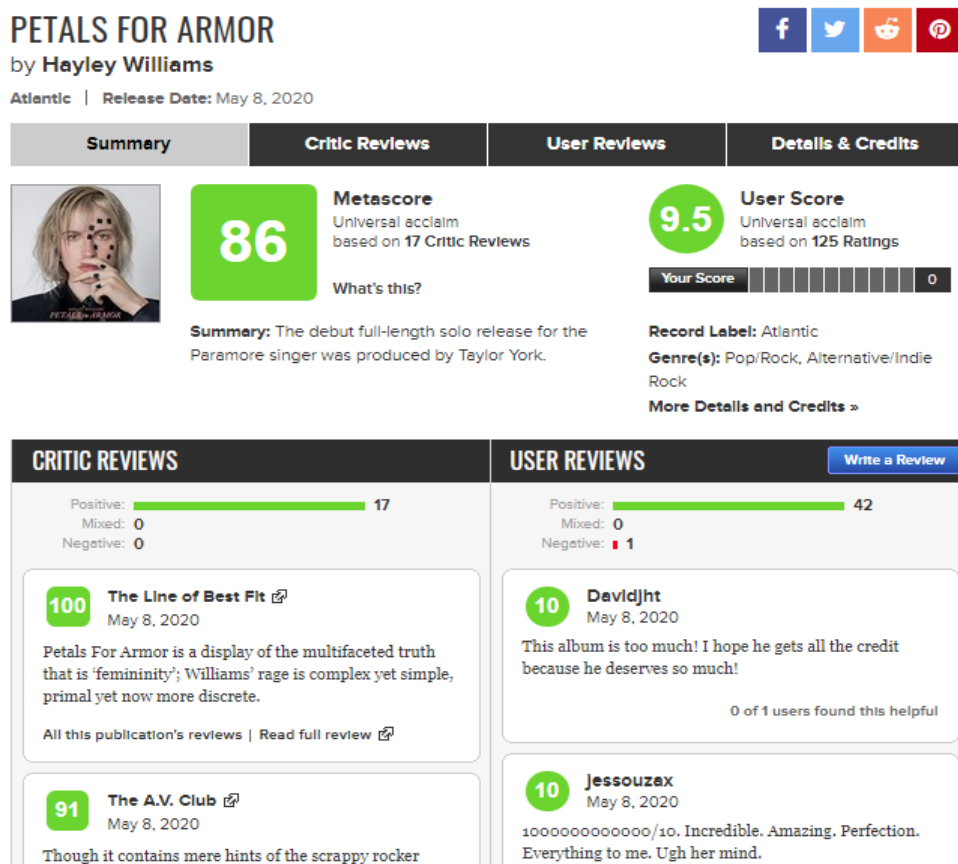


Рисунок 2.2 Сторінка продукту споживання

Можна з'явити увагу і на такі сторінки як:

- Особистий кабінет
- Пошук
- Інформаційна сторінка

На цій основі можна побудувати приблизну структуру інтерфейсу сайту:

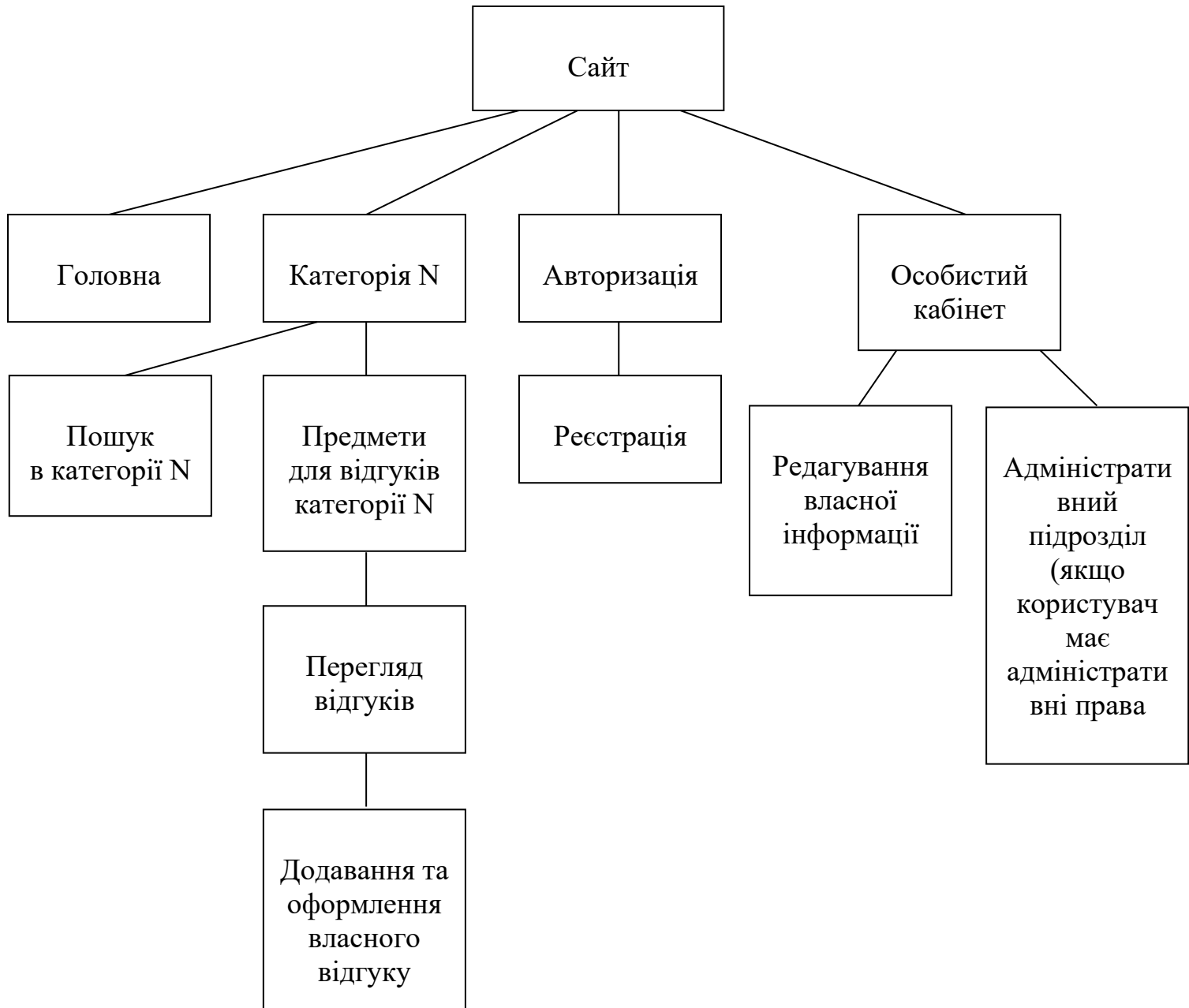
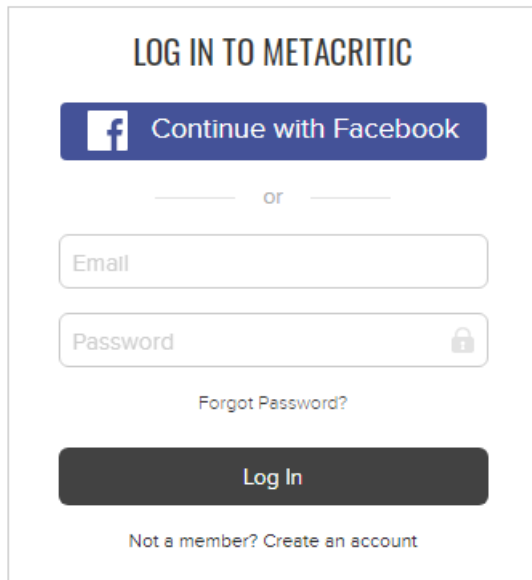


Рисунок 2.3 Структура інтерфейсусайту

На прикладі Metacritic ми можемо розглянути функції, які знадобляться нам для розробки власного сайту.

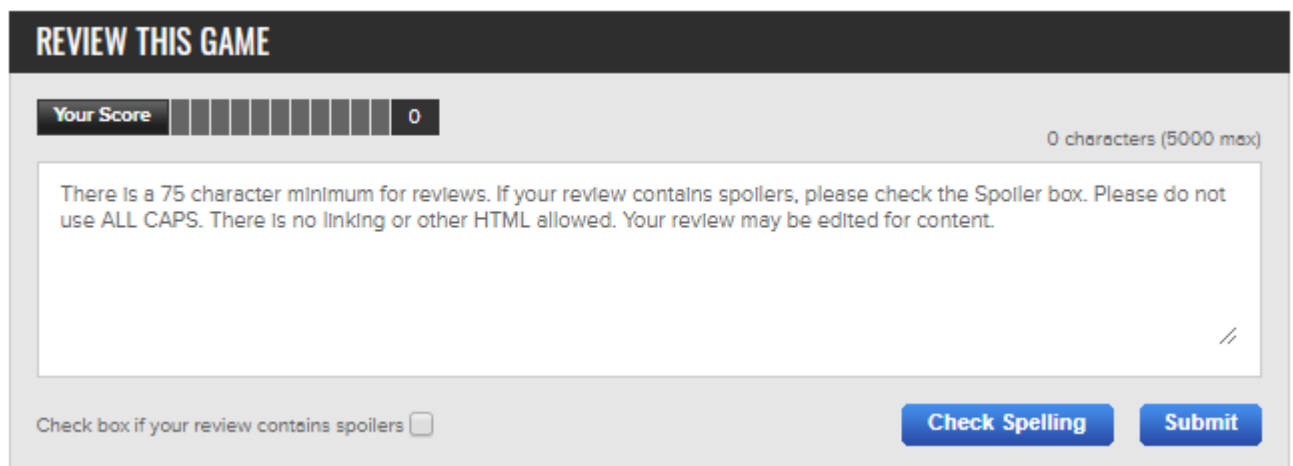
Реєстрація дає можливість користувачу залишати ці самі відгуки, в той час незареєстрованні користувачі можуть тільки переглядати чужі відгуки.



The image shows a login form titled "LOG IN TO METACRITIC". It features a "Continue with Facebook" button with the Facebook logo. Below this is a separator line with the word "or". There are two input fields: "Email" and "Password" (with a lock icon). A link "Forgot Password?" is located below the password field. A large "Log In" button is at the bottom, with a link "Not a member? Create an account" underneath it.

Рисунок 2.4 Форма реєстрації

Одна із основних функцій це створення власного посту(відгуку)і подальшого його редагування.



The image shows a review form titled "REVIEW THIS GAME". It includes a "Your Score" section with a row of 10 boxes and a "0" in the last one. To the right, it says "0 characters (5000 max)". Below this is a large text area with a placeholder message: "There is a 75 character minimum for reviews. If your review contains spoilers, please check the Spoiler box. Please do not use ALL CAPS. There is no linking or other HTML allowed. Your review may be edited for content." At the bottom left, there is a checkbox labeled "Check box if your review contains spoilers". At the bottom right, there are two buttons: "Check Spelling" and "Submit".

Рисунок 2.5 Створення власного відгуку

Для того, щоб представити функціонал сайту, виділено такі процеси як:

- Реєстрація та авторизація користувача
- Створення та оформлення власних відгуків
- Редагування власної інформації
- Перегляд історії відгуківпредмету огляду
- Пошук за назвою в категоріях

2.3 Логічне моделювання

На даному етапі використовується методологія об'єктно-орієнтованого проектування та аналізу, заснована на мові візуального моделювання UML.

Згідно з методологією ОМТ, для опису об'єктної моделі інформаційної системи на логічному рівні досить побудувати діаграми UML, що відображають функціональний, статичний і динамічний аспекти проектованої інформаційної системи.

2.3.1 Діаграма варіантів використання сайту

Щоб повністю показати функціональний аспект даної системи потрібно побудувати діаграму варіантів використання.

Дана діаграма описує всі функції даної ІС, показуючи додаткову інформацію про відносинах між різними варіантами використання.

Застосовуються діаграми варіантів використання при моделюванні видів робіт, виконуваних на підприємстві і функціональних вимог до проектованої системи при її проектуванні і розробці.

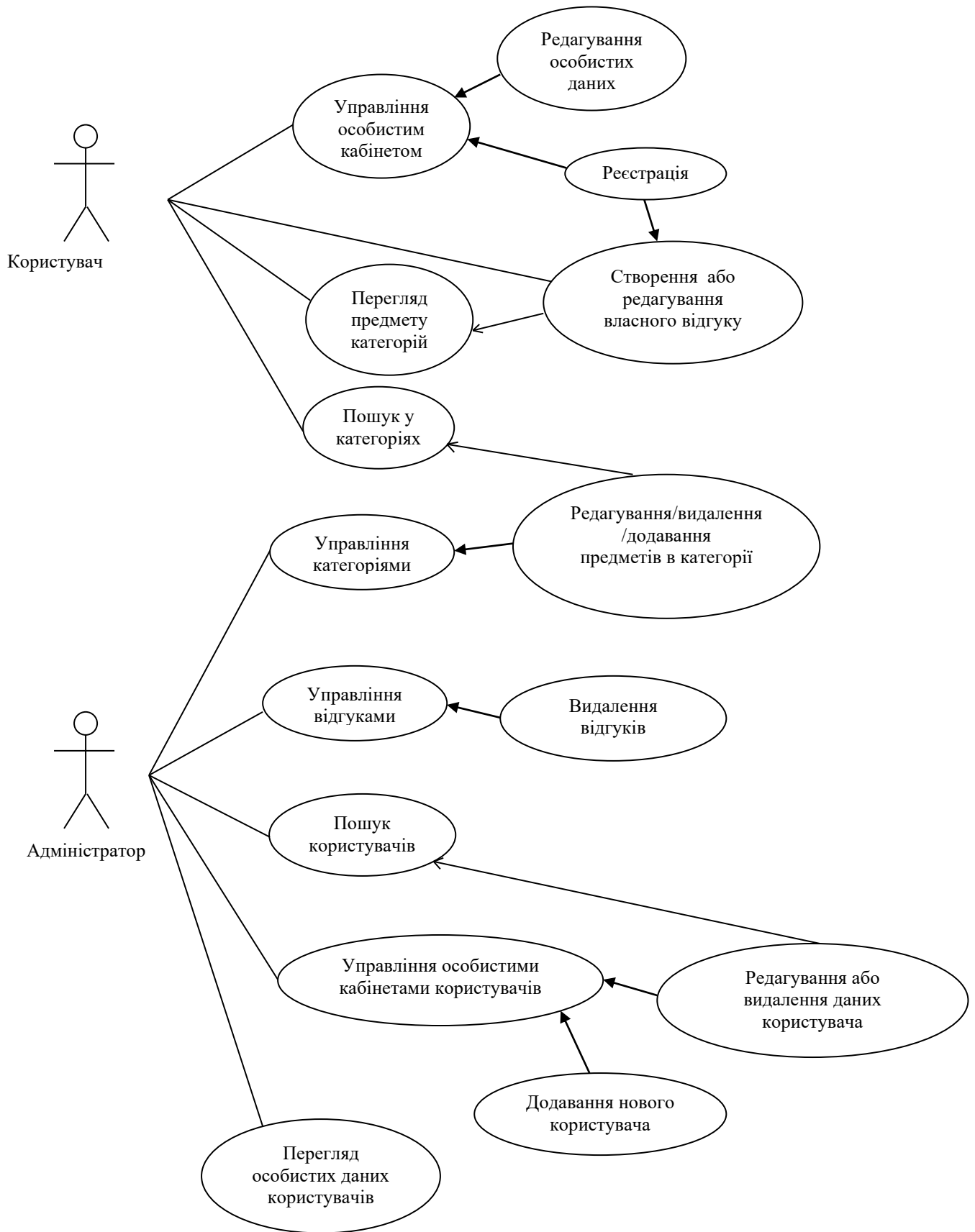


Рисунок 2.6 Діаграма варіантів використання сайту

Склад діаграми

Актори:

- 1) Користувач - відвідувач сайту який має право переглядати предмети що знаходяться в категоріях, а також залишати на них свої відгуки. Має можливість переглянути свій особистий кабінет та відредагувати власну інформацію.
- 2) Адміністратор сайту - зареєстрований користувач ресурсу, в обов'язки якого входить оновлення та додавання предметів для відгуків у категоріях. Окрім того має права видаляти відгуки які не відповідають нормам. Володіє правами адміністратора ресурсу і керуючий контентом сайту.

Варіанти використання:

Керування обліковим записом

-

незареєстрований користувач може зареєструватися.

При

реєстрації користувач отримує логін і пароль, які він буде в подальшому використовувати для доступу до ресурсу.

Перегляд предметів - Клієнт може ознайомитися з предметом, а також залишати свій відгук (якщо він зареєстрований) та бачити відгуки інших людей.

Пошук - Клієнту надається можливість швидкого пошуку в категоріях за назвою предмету.

Створення власного відгуку - Користувач може створити власний відгук, якщо він є зареєстрованим користувачем даного сайту.

Управління відгуками - адміністрація сайту має контроль над відгуками, і у разі виявлення некоректного відгуку має право їх видаляти.

Служба захисту користувачів - адміністратор сайту має можливість знаходити, додавати, редагувати і видаляти зареєстрованих користувачів.

2.3.2 Діаграма класів сайту

Діаграма класів відображає взаємозв'язки між предметною областю, такими як об'єкти і підсистеми, а також описують типи відносин і внутрішню структуру.

Позначення класу в мові UML служить для позначення безлічі об'єктів, які мають однакову структуру, і відносини з об'єктами інших класів.

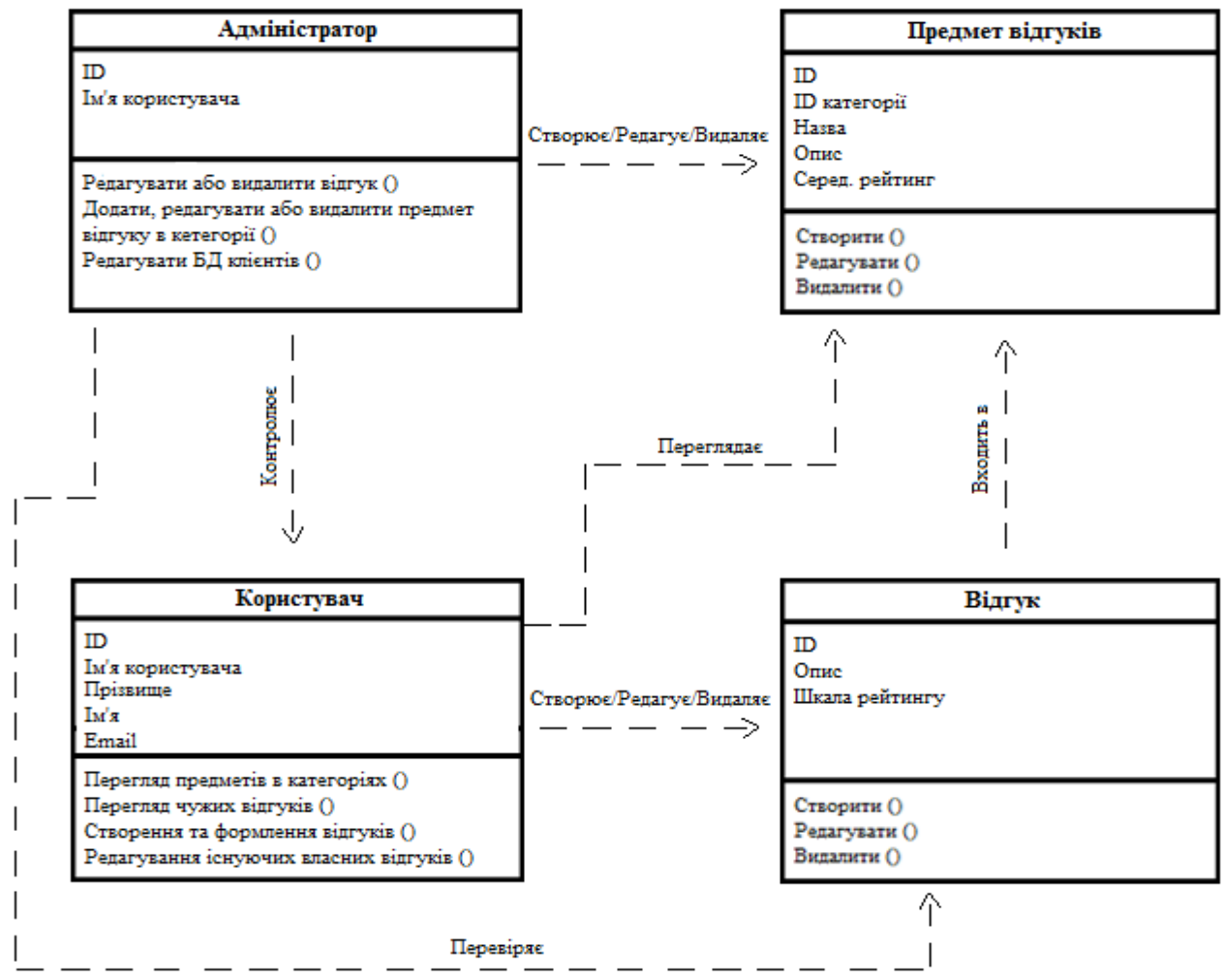


Рисунок 2.7 Діаграма класів сайту

Специфікація класів сайту:

- 1) Адміністратор - клас об'єктів-користувачів сайту, що виконують дії з предметами відгуків, додаючи, редагуючи і видаляючи їх. Має контроль над користувачами та їх відгуками.
- 2) Відгук - клас об'єктів-відгуків від користувачів;
- 3) Користувач - клас об'єктів-користувачів сайту, які мають можливість переглядати предмети відгуків, а також створювати до них власні відгуки з можливістю подальшого редагування або видалення їх.
- 4) Предмет відгуків - об'єктів-предметів що створюються адміністраторами. Мають короткий опис та середній рейтинг який вираховується на основі рейтингу всіх відгуків.

2.4 Архітектура БД сайту

Створимо архітектуру баз даних нашого сайту. Для початку сайт буде мати 5 категорії для відгуків, а саме ігри, кіно, музика, книжки та послуги-сервіси.

В БД створимо таблицю під назвою Категорії в якій буде міститись інформація з назвами категорій веб-сайту. Також створимо таблицю Відгук з полем категорія, яке буде зовнішнім ключем таблиці Категорії. Для реєстрації нових користувачів нам знадобиться таблиця під назвою Користувач.

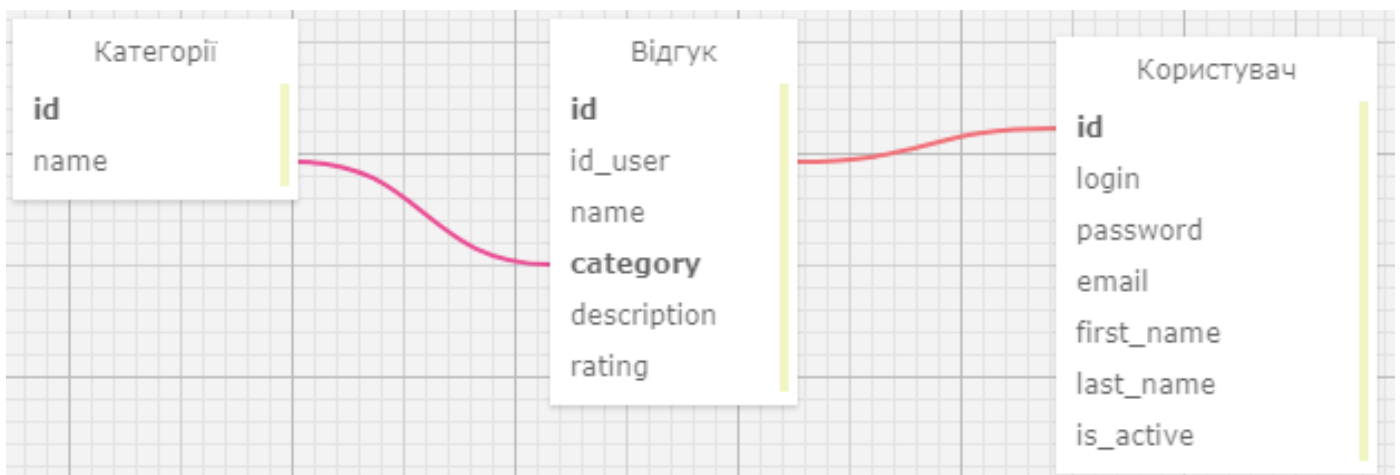


Рисунок 2.8 Концептуальна модель бази дани

Для наочного прикладу представимо нашу базу даних у таблиці, де назвастаблиці – об'єкт, а ім'я поля – ідентифікатор.

Об'єкт	Властивість	Тип	Ідентифікатор
Категорії	Назва	Char	name
Відгук	Код користувача	Integer	id_user
	Назва	Char	name
	Категорія	ForeignKey	category
	Опис	Text	description
	Рейтинг	Char	rating
Користувач	Логін	Char	Login
	Пароль	Char	Password
	Електронна пошта	Email	email
	Ім'я	Char	first_name
	Прізвище	Char	last_name
	Активний статус	Char	is_active

Таблиця 2.1 Концептуальна модель бази даних

2.5 Інструменти розробки

Python – мова програмування, що дуже швидко розвивається завдяки своїй простоті і універсальності. Він досить простий у вивченні і розумінні синтаксису. При цьому досить потужний за своїми можливостями. На ньому можна писати як простенькі скрипти, так і складні додатки і навіть деякі види ігор. Ще одним плюсом мови Python є його кросплатформеність, тобто написаний код буде працювати на будь-якій операційній системі і на пристроях різного типу - від комп'ютера до смартфона. Головне - встановити інтерпретатор мови Python на пристрій в потрібній версії і можна виконувати код.

Серед основних її переваг можна назвати такі:

1. чистий синтаксис (для виділення блоків слід використовувати відступи);
2. переносність програм (що властиве більшості інтерпретованих мов);
3. стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
4. можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
5. стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написане мовою Python;
6. зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
7. відкритий код (можливість редагувати його іншими користувачами).

Для написання сайту також необхідний фреймворк. Фреймворк - набір стандартного інструментарію для створення сайтів. Фреймворк може включати реалізацію стандартних функцій, які часто використовуються при створенні сайту і служить для прискорення процесу розробки. При цьому фреймворк - це не конструктор, а всього лише база, що дозволяє швидко запустити сайт і забезпечити його роботу.

Для нашого проекту ми обрали Django, бо це безкоштовний і вільний фреймворк для веб-додатків, написаний на Python.

Кожен раз при розробці веб-сайтів потрібні схожі компоненти: спосіб аутентифікувати користувачів (вхід, вихід, реєстрація), панель управління сайтом, форми, інструменти для завантаження файлів і так далі. Інші люди звернули увагу на виникнення однотипних проблем при веб-розробці, так що вони об'єдналися і створили фреймворки (Django і інші), які пропонують нам готові шаблони для використання.

Деякі можливості Django:

1. ORM, API доступу до БД з підтримкою транзакцій
2. вбудований інтерфейс адміністратора, з уже наявними перекладами на більшості мов
3. диспетчер URL на основі регулярних виразів
4. розширювана система шаблонів з тегами та наслідуванням
5. система кешування
6. інтернаціоналізація
7. архітектура застосунків, що підключаються, які можна встановлювати на будь-які Django-сайти
8. «generic views» - шаблони функцій контролерів
9. авторизація та аутентифікація, підключення зовнішніх модулів аутентифікації: LDAP, OpenID та ін.
10. система фільтрів («middleware») для побудови додаткових обробників запитів, наприклад включені в дистрибутив фільтри для кешування, стиснення, нормалізації URL і підтримки анонімних сесій
11. бібліотека для роботи з формами (наслідування, побудова форм за існуючою моделлю БД)
12. вбудована автоматична документація по тегах шаблонів та моделям даних, доступна через адміністративний застосунок

3.ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Загальний опис та файлова структура сайту

Програмне забезпечення цього сайту розробляється на базі Python з використанням фреймворку Django. Для БД використовується стандартна БД Django – SQLite. Усі тести цього продукту проводяться у віртуальному середовищі що Python надає. В якості тестового хостингу ми використовуємо локальний хостинг з адресою 127.0.0.1 та портом :8000. В майбутньому на фазі впровадження сайту в простір Інтернету можна буде використати безкоштовний хостинг Heroku або придбати хостинг деінде.

Загальна файлова структура сайту містить:

- 2 основні папки з додатками (main та accounts) в яких розписана майже уся Backend та Frontend частини розробки сайту відгуків.
- Папка віртуального середовища env, де знаходяться усі бібліотеки та модулі сайту, включаючи сам Django.
- Папка з налаштуваннями та конфігурацією URL-адрес – reviewwebsite
- Папка staticfiles що містить статистичні файли що використовуються в Frontend розробці.
- База даних db.sqlite3 де містяться усі наші дані що потім використовуються у виді моделей за допомогою Django в Backend частині розробки сайту.
- Файл manage.py що утилітою командного рядка Django для адміністративних завдань.

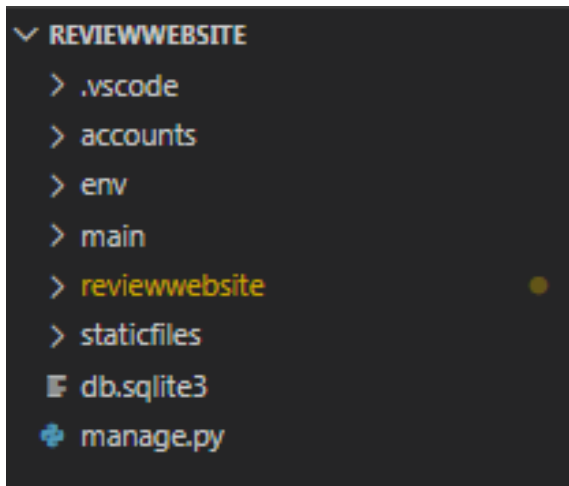


Рисунок 3.1 Загальна файлова структура сайту

3.2 Backend частина розробки сайту

Вся основна Backend частина розробки сайту знаходиться в 2-хосновних папках з додатками (main та accounts).

За допомогою команд Django можна активувати заздалегідь заданий самим Django шаблон для створення додатків.

3.2.1 Додаток main

Цей додаток відповідає за базу категорій та відгуків сайту.

Файлова структура Backend частини main містить:

- Папку migrations, де зберігаються будь які зміни внесені до бази даних за допомогою моделей в models.py.
- Файл admin.py, що відповідає за відображення БД в адміністративному розділі сайту.
- Файл apps.py, де виділений клас що відповідає за налаштування даного додатку.
- Файл forms.py, де створюються та зберігаються форми що використовує сайт. Всі дані для форм імпортуються з моделей БД із models.py.

- Файл `models.py`, де створюються усі класи моделей даних даного додатку, що потім зберігаються у БД SQLite.
- Файл `urls.py`, де встановлюється та зберігається маршрутизація URL-адрес даного додатку.
- Файл `views.py`, де прописані усі функції даного додатку.

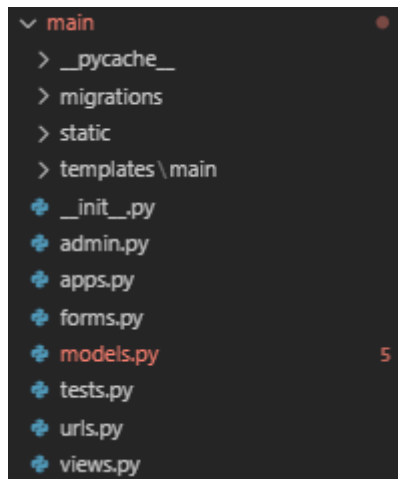


Рисунок 3.2 Файлова структура додатку main

Основними файлами в розробці Backend частини даного додатку у сайті є: `forms.py`, `models.py`, `urls.py` та звичайно `views.py`.

models.py

На зображенні 3.3 зображено приклад класу моделі БД для категорії фільмів, що будуть додаватися адміністратором сайту. Ця модель містить такі поля як:

- Назва фільму
- Режисер фільму
- Акторський склад фільму
- Жанр фільму
- Опис фільму
- Дата виходу фільму у прокат
- Середній рейтинг, визначається на основі відгуків користувачів.
- Зображення


```
class Фільм(models.Model):
    # Поля для фільмів
    назва = models.CharField(max_length=300)
    режисер = models.CharField(max_length=300)
    акторський_склад = models.CharField(max_length=800)
    жанр = models.CharField(max_length=800)
    опис = models.TextField(max_length=5000)
    дата_виходу = models.DateField()
    середній_рейтинг = models.FloatField(default=0)
    зображення = models.URLField(default=None, null=True)

    def __str__(self):
        return self.назва

    def __unicode__(self):
        return self.назва
```

Рисунок 3.3 Приклад класу моделі категорії в models.py

Схожі класи моделей, але з по інколи іншими назвами полів були створені і для категорії ігор, музикальних альбомів, книжок, сервісів та нових публікації від адміністратора.

В models.py також присутній ряд класів моделей для відгуків користувачів що також зберігаються у нашій БД SQLite після проведення усіх міграції.

На зображенні 3.4 зображено клас моделі для відгуку до фільму.

Цей клас містить такі поля як:

- Назва фільму, дані до якого надходять з класу моделей для категорії фільму, за допомогою зовнішнього ключа.
- Ім'я користувача, дані до якого надходять вже з моделей БД додатку accounts, за допомогою зовнішнього ключа.
- Коментар
- Рейтинг

```
class Фільм_відгук(models.Model):
    фільм = models.ForeignKey(Фільм, on_delete=models.CASCADE)
    користувач = models.ForeignKey(User, on_delete=models.CASCADE)
    коментар = models.TextField(max_length=1000)
    рейтинг = models.FloatField(default=0)

    def __str__(self):
        return self.user.username
```

Рисунок 3.4 Приклад класу моделі відгуку до фільмів models.py

Так само як і для моделей категорії сайту, існують схожі моделі і для відгуків на ігри, музикальні альбоми, книжки чи сервіси.

Всі зміни, що вносяться до наведених класів моделей зберігаються за допомогою міграції, що вбудовані в функціонал фреймворка Django.

forms.py

В даному файлі цього додатку створюються та зберігаються Django-форми, що потім використовуються для створення або редагування предметів для відгуків або цих самих відгуків вже за допомогою інтерфейсу самого сайту.

На зображеннях 3.5 та 3.6 зображенні класи цих форм для створення або редагування предмету для категорії фільмів та відгуку до фільму відповідно.

```
class MovieForm(forms.ModelForm):
    class Meta:
        model = Фільм
        fields = ('назва', 'режисер', 'акторський_склад', 'жанр', 'опис', 'дата_виходу', 'зображення')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['назва'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['режисер'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['акторський_склад'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['жанр'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['опис'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['дата_виходу'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['зображення'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
```

Рисунок 3.5 Клас форми (MovieForm) для створення або редагування предмету для відгуків категорії фільмів.

```
class Movie_reviewForm(forms.ModelForm):
    class Meta:
        model = Фільм_відгук
        fields = ("коментар", "рейтинг")

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['коментар'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['рейтинг'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
```

Рисунок 3.6 Клас форми (Movie_reviewForm) для створення або редагування самого відгуків до фільму.

Поля для цих форм імпортуються із models.py із відповідних класів моделей.

views.py

В даному файлі, що є найважливішим у всій Backendчастині сайту, розписані усі функції для даного додатку сайту.

У цілому в даному додатку описані такі функції як:

- Відображення головної сторінки на сайті
- Відображення сторінки категорії
- Відображення сторінки з деталями предмету відгуку
- Додавання предмету для відгуку
- Редагування предмету для відгуку
- Видалення предмету для відгуку
- Додавання відгуку
- Редагування відгуку
- Видалення відгуку

На наступних зображеннях зображенні приклади опису даних функцій.

```
def home(request):
    allWhats_new = Що_нового.objects.all().order_by("-id")

    context = {
        "whats_new": allWhats_new,
    }

    return render(request, 'main/index.html', context)
```

Рисунок 3.7 Функція відображення головної сторінки на сайті

```
def directory_movies(request):
    query = request.GET.get("movie_title")
    allMovies = None
    if query:
        allMovies = Фільм.objects.filter(назва__icontains=query)
    else:
        allMovies = Фільм.objects.all()

    context = {
        "movies": allMovies,
    }

    return render(request, 'main/directory_movies.html', context)
```

Рисунок 3.8 Функція відображення сторінки категорії фільмів

```
def detail_movie(request, id):
    movie = Фільм.objects.get(id=id)
    moviereviews = Фільм_відгук.objects.filter(фільм=id).order_by("коментар")

    average = moviereviews.aggregate(Avg("рейтинг"))["рейтинг__avg"]
    if average == None:
        average = 0
    average = round(average, 2)
    context = {
        "movie": movie,
        "moviereviews": moviereviews,
        "average": average,
    }

    return render(request, 'main/details_movie.html', context)
```

Рисунок 3.9 Функція відображення сторінки з деталями фільму

```
def add_movies(request):
    if request.user.is_authenticated:
        if request.user.is_superuser:
            if request.method == "POST":
                form = MovieForm(request.POST or None)

                if form.is_valid():
                    data = form.save(commit=False)
                    data.save()
                    return redirect("main:directory_movies")
            else:
                form = MovieForm()
                return render(request, 'main/add_movie.html', {"form": form, "controller": "Додати фільм"})
        else:
            return redirect("main:home")
    return redirect("accounts:login")
```

Рисунок 3.10 Функція додавання нового фільму у категорію фільмів.

```
def edit_movies(request, id):
    movie = Фільм.objects.get(id=id)

    if request.user.is_authenticated:
        if request.user.is_superuser:
            if request.method == "POST":
                form = MovieForm(request.POST or None, instance=movie)

                if form.is_valid():
                    data = form.save(commit=False)
                    data.save()
                    return redirect("main:details_movie", id)
            else:
                form = MovieForm(instance=movie)
                return render(request, 'main/add_movie.html', {"form": form, "controller": "Редагувати фільм"})
        else:
            return redirect("main:details_movie", id)
    return redirect("accounts:login")
```

Рисунок 3.11 Функція редагування існуючого фільму в категорії фільмів.

```
def delete_movies(request, id):

    movie = Фільм.objects.get(id=id)

    if request.user.is_authenticated:
        if request.user.is_superuser:

            movie.delete()
            return redirect("main:directory_movies")
        else:
            return redirect("main:details_movie", id)
    return redirect("accounts:login")
```

Рисунок 3.12 Функція видалення існуючого фільму із категорії фільмів.

```
def add_movie_review(request, id):
    if request.user.is_authenticated:
        movie = Фільм.objects.get(id=id)
        if request.method == "POST":
            form = Movie_reviewForm(request.POST or None)
            if form.is_valid():
                data = form.save(commit=False)
                data.коментар = request.POST["коментар"]
                data.рейтинг = request.POST["рейтинг"]
                data.користувач = request.user
                data.фільм = movie
                data.save()
                return redirect("main:details_movie", id)
            else:
                form = Movie_reviewForm()
                return render(request, 'main/details_movie.html', {"form": form})
        else:
            return redirect("accounts:login")
```

Рисунок 3.13 Функція додавання відгуку до фільму.

```
def edit_movie_review(request, movie_id, moviereview_id):
    if request.user.is_authenticated:
        movie = Фільм.objects.get(id=movie_id)
        moviereview = Фільм відгук.objects.get(фільм=movie, id=moviereview_id)

        if request.user == moviereview.користувач or request.user.is_superuser:

            if request.method == "POST":
                form = Movie_reviewForm(request.POST, instance=moviereview)
                if form.is_valid():
                    data = form.save(commit=False)
                    if (data.рейтинг > 10) or (data.рейтинг < 0):
                        error = "За межою оцінювання. Встановіть рейтинг від 0 до 10."
                        return render(request, 'main/edit_movie_review.html', {"error": error, "form": form})
                    else:
                        data.save()
                        return redirect("main:details_movie", movie_id)
                else:
                    form = Movie_reviewForm(instance=moviereview)
                    return render(request, 'main/edit_movie_review.html', {"form": form})
            else:
                return redirect("main:details_movie", movie_id)
        else:
            return redirect("accounts:login")
```

Рисунок 3.14 Функція редагування відгуку до фільму.

```
def delete_movie_review(request, movie_id, moviereview_id):
    if request.user.is_authenticated:
        movie = Фільм.objects.get(id=movie_id)
        moviereview = Фільм відгук.objects.get(Фільм=movie, id=moviereview_id)

        if request.user == moviereview.користувач or request.user.is_superuser:
            moviereview.delete()

        return redirect("main:details_movie", movie_id)

    else:
        return redirect("accounts:login")
```

Рисунок 3.15 Функція видалення відгуку до фільму.

Подібні функції розписані для всіх видів категорій.

Майже усі функції використовують форми із forms.py для формування їх основного функціоналу у виді інтерактивного інтерфейсу.

urls.py

В даному файлі знаходиться маршрутизація сторінок даного додатку сайту.

Кожна URL-адреса відповідає функції в views.py и активує ці самі функції коли користувач потрапляє на відповідну адресу.

На зображенні 3.16 та 3.17 показано всі дані URL-адреси.

```
app_name = "main"

urlpatterns = [
    #головна
    path("", views.home, name="home"),
    #категорії
    path("directory_movies/", views.directory_movies, name="directory_movies"),
    path("directory_games/", views.directory_games, name="directory_games"),
    path("directory_albums/", views.directory_albums, name="directory_albums"),
    path("directory_books/", views.directory_books, name="directory_books"),
    path("directory_services/", views.directory_services, name="directory_services"),
    #подробити
    path("details_movie/<int:id>/", views.detail_movie, name="details_movie"),
    path("details_game/<int:id>/", views.detail_game, name="details_game"),
    path("details_album/<int:id>/", views.detail_album, name="details_album"),
    path("details_book/<int:id>/", views.detail_book, name="details_book"),
    path("details_service/<int:id>/", views.detail_service, name="details_service"),
    #додавання в категорії
    path("add_movie/", views.add_movies, name="add_movie"),
    path("add_game/", views.add_games, name="add_game"),
    path("add_album/", views.add_albums, name="add_album"),
    path("add_book/", views.add_books, name="add_book"),
    path("add_service/", views.add_services, name="add_service"),
    path("add_whats_new/", views.add_whats_new, name="add_whats_new"),
    #редагування
    path("edit_movie/<int:id>/", views.edit_movies, name="edit_movie"),
    path("edit_game/<int:id>/", views.edit_games, name="edit_game"),
    path("edit_album/<int:id>/", views.edit_albums, name="edit_album"),
    path("edit_book/<int:id>/", views.edit_books, name="edit_book"),
    path("edit_service/<int:id>/", views.edit_services, name="edit_service"),
    path("edit_whats_new/<int:id>/", views.edit_whats_new, name="edit_whats_new"),
```

Рисунок 3.16

```

#видалення
path("delete_movie/<int:id>/", views.delete_movies, name="delete_movie"),
path("delete_game/<int:id>/", views.delete_games, name="delete_game"),
path("delete_album/<int:id>/", views.delete_albums, name="delete_album"),
path("delete_book/<int:id>/", views.delete_books, name="delete_book"),
path("delete_service/<int:id>/", views.delete_services, name="delete_service"),
path("delete_whats_new/<int:id>/", views.delete_whats_new, name="delete_whats_new"),
#додавання відгуків
path("add_movie_review/<int:id>/", views.add_movie_review, name="add_movie_review"),
path("add_game_review/<int:id>/", views.add_game_review, name="add_game_review"),
path("add_album_review/<int:id>/", views.add_album_review, name="add_album_review"),
path("add_book_review/<int:id>/", views.add_book_review, name="add_book_review"),
path("add_service_review/<int:id>/", views.add_service_review, name="add_service_review"),
#редагування відгуків
path("edit_movie_review/<int:movie_id>/<int:moviereview_id>/", views.edit_movie_review, name="edit_movie_review"),
path("edit_game_review/<int:game_id>/<int:gamereview_id>/", views.edit_game_review, name="edit_game_review"),
path("edit_album_review/<int:album_id>/<int:albumreview_id>/", views.edit_album_review, name="edit_album_review"),
path("edit_book_review/<int:book_id>/<int:bookreview_id>/", views.edit_book_review, name="edit_book_review"),
path("edit_service_review/<int:service_id>/<int:servicereview_id>/", views.edit_service_review, name="edit_service_review"),
#видалення відгуків
path("delete_movie_review/<int:movie_id>/<int:moviereview_id>/", views.delete_movie_review, name="delete_movie_review"),
path("delete_game_review/<int:game_id>/<int:gamereview_id>/", views.delete_game_review, name="delete_game_review"),
path("delete_album_review/<int:album_id>/<int:albumreview_id>/", views.delete_album_review, name="delete_album_review"),
path("delete_book_review/<int:book_id>/<int:bookreview_id>/", views.delete_book_review, name="delete_book_review"),
path("delete_service_review/<int:service_id>/<int:servicereview_id>/", views.delete_service_review, name="delete_service_review"),

```

Рисунок 3.17

3.2.2 Додаток accounts

Цей додаток відповідає за базу користувачів.

Файлова структура Backendчастини accounts містить:

- Папку migrations, де зберігаються будь які зміни внесені до бази даних за допомогою моделей в models.py.
- Файл admin.py, що відповідає за відображення БД в адміністративному розділі сайту.
- Файл apps.py, де виділений клас що відповідає за налаштування даного додатку.
- Файл forms.py, де створюються та зберігаються форми що використовує сайт. Всі дані для форм імпортуються з моделей БД із models.py.
- Файл urls.py, де встановлюється та зберігається маршрутизація URL-адрес даного додатку.
- Файл views.py, де прописані усі функції даного додатку.

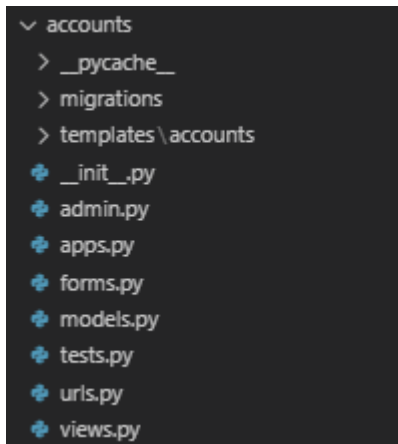


Рисунок 3.18 Файлова структура додатку accounts

Основними файлами в розробці Backend частини даного додатку у сайті є: forms.py, urls.py та звичайно views.py. В даному додатку відсутнє використання файлу models.py оскільки при реєстрації та підтримці персональних кабінетів користувачів ми будемо використовувати вбудовані моделі Django.

forms.py

В даному файлі цього додатку створюються та зберігаються Django-форми для реєстрації та подальшого редагування персонального кабінету користувача сайту.

На рис. 3.19 та 3.20 зображенні класи цих самих форм.

```
class RegistrationForm(UserCreationForm):
    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'email', 'password1', 'password2')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['username'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['first_name'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['last_name'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['email'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['password1'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['password2'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
```

Рисунок 3.19 Клас форми (RegistrationForm) для реєстрування нових користувачів у системі та занесення їх у БД SQLite.


```
class Edit_accountForm(forms.ModelForm):
    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'email')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['username'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['first_name'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['last_name'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
        self.fields['email'].widget.attrs.update({'style': 'background-color: #262e35; color: white;'})
```

Рисунок 3.20 Клас форми (Edit_accountForm) для редагування персонального кабінету користувача.

Поля для цих форм імпортуються із уже існуючих класів моделей що знаходяться в бібліотеці фреймворка Django.

views.py

В даному файлі, розписані усі функції для даного додатку сайту.

У цілому в даному додатку описані такі функції як:

- Реєстрація користувача
- Вхід в особистий кабінет
- Вихід із особистого кабінету
- Деталі особистого кабінету
- Редагування особистого кабінету
- Видалення особистого кабінету

На наступних зображеннях зображенні приклади опису даних функцій.

Майже усі функції використовують форми із forms.py для формування їх основного функціоналу у виді інтерактивного інтерфейсу.

```
#Реєстрація користувачів

def register(request):
    if request.user.is_authenticated:
        return redirect("main:home")
    else:
        if request.method == "POST":
            form = RegistrationForm(request.POST or None)

            if form.is_valid():
                user = form.save()

                raw_password = form.cleaned_data.get('password1')

                user = authenticate(username=user.username, password=raw_password)

                login(request, user)

                return redirect("main:home")
        else:
            form = RegistrationForm()
            return render(request, "accounts/register.html", {"form": form})
```

Рисунок 3.21 Функція реєстрації користувача

```
#Вхід в аккаунт
def login_user(request):
    if request.user.is_authenticated:
        return redirect("main:home")
    else:
        if request.method == "POST":
            username = request.POST['username']
            password = request.POST['password']

            user = authenticate(username=username, password=password)

            if user is not None:
                if user.is_active:
                    login(request, user)
                    return redirect("main:home")
                else:
                    return render(request, 'accounts/login.html', {"error": "Ваш аккаунт був відключений."})
            else:
                return render(request, 'accounts/login.html', {"error": "Неправильне ім'я користувача або пароль. Спробуйте знову."})
    return render(request, 'accounts/login.html')
```

Рисунок 3.21 Функція входу в особистий кабінет

```
#Вихід з аккаунта

def logout_user(request):
    if request.user.is_authenticated:
        logout(request)
        return redirect("accounts:login")
    else:
        return redirect("accounts:login")
```

Рисунок 3.22 Функція виходу із особистого кабінету.

```
def detail_account(request, id):
    user = User.objects.get(id=id)

    if request.user.is_authenticated:
        if request.user.id == user.id:

            context = {
                "user": user,
            }

            return render(request, 'accounts/details_account.html', context)
        else:
            return redirect('main:home')
    else:
        return redirect('accounts:login')
```

Рисунок 3.23 Функція деталей особистого кабінету

```
#Редагування аккаунту

def edit_account(request, id):
    user = User.objects.get(id=id)

    if request.user.is_authenticated:
        if request.user.id == user.id:

            if request.method == "POST":
                form = Edit_accountForm(request.POST or None, instance=user)

                if form.is_valid():
                    data = form.save(commit=False)
                    data.save()
                    return redirect("accounts:details_account", id)
            else:
                form = Edit_accountForm(instance=user)
                return render(request, 'accounts/edit_account.html', {"form": form})
        else:
            return redirect("accounts:details_account", id)
    return redirect("accounts:login")
```

Рисунок 3.24 Функція редагування особистого кабінету

```
#Видалення аккаунту

def delete_account(request, id):
    user = User.objects.get(id=id)

    if request.user.is_authenticated:
        if request.user.id == user.id:
            user.delete()
            return redirect("main:home")
        else:
            return redirect("accounts:details_account", id)
    return redirect(["accounts:login"])
```

Рисунок 3.25 Функція видалення особистого кабінету

urls.py

В даному файлі знаходиться маршрутизація сторінок даного додатку сайту.

Кожна URL-адреса відповідає функції в views.py и активує ці самі функції коли користувач потрапляє на відповідну адресу.

На зображенні 3.26 показано всі дані URL-адреси.

```
app_name = "accounts"

urlpatterns = [
    path("register/", views.register, name="register"),
    path("login/", views.login_user, name="login"),
    path("logout/", views.logout_user, name="logout"),
    path("details_account/<int:id>/", views.detail_account, name="details_account"),
    path("edit_account/<int:id>/", views.edit_account, name="edit_account"),
    path("delete_account/<int:id>/", views.delete_account, name="delete_account"),
]
```

Рисунок 3.26

3.3 Frontend частина розробки сайту

Frontend частина розробки сайту також знаходиться в 2-хосновних папках з додатками (main та accounts).

3.3.1 Додаток main

Цей додаток відповідає за базу категорій та відгуків сайту.

Файлова структура Frontend частини main містить:

- Папки static/css в якій розташований файл main.css, що містить статичні стилі для нашого сайту.
- Папки templates/main, що містить html файли сторінок які виводяться на сайті.

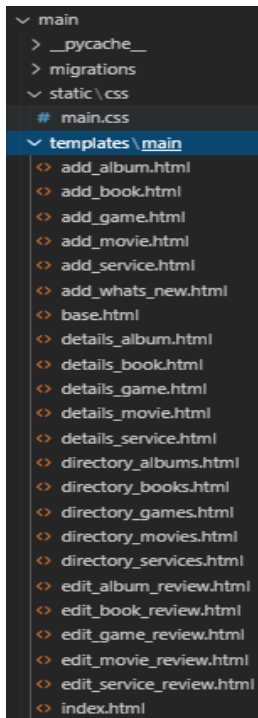


Рисунок 3.27 Файлова структура Frontendчастини main

Розглянемо папку `templates/main`. Кожна з `html` сторінок відповідає функціям описаним в Backend частині цього додатку, що знаходяться в `views.py`.

Оскільки більшість категорії сайту мають майже однакову кодову структуру, то є сенс розглянути тільки одну з них, наприклад розглянемо файли, що мають відношення до категорії фільмів:

- Файл `add_movie.html` відкриває сторінку за допомогою якої адміністратор може додати новий фільм у відповідну категорію.
- Файл `details_movie.html` відкриває сторінку з подробицями про фільм який користувач вибрав.
- Файл `directory_movie.html` відкриває сторінку категорії фільмів, де відображаються усі фільми, що були додані до БД адміністратором.
- Файл `edit_movie_review.html` відкриває сторінку для редагування відгуку але тільки для користувача який його створив.

Окрім цих файлів, є ще два основних інших файли html – index.html та base.html:

- Файл index.html – це головна сторінка сайту, де адміністратор публікує новини щодо сайту.
- Файл base.html – це базова група елементів інтерфейсу сторінок сайту, що входить до будь якої сторінки (наприклад навігаційна панель що присутня зверху кожної сторінки сайту).

3.3.2 Додаток accounts

Цей додаток відповідає за базу користувачів.

Файлова структура Frontend частини accounts містить:

- Папки templates/accounts, що містить html файли сторінок які виводяться на сайті.

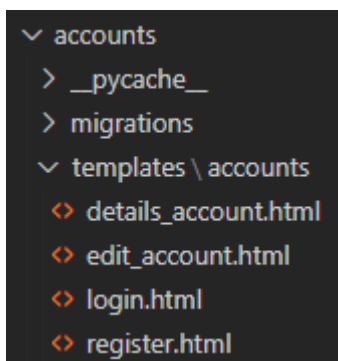


Рисунок 3.28 Файлова структура Frontend частини accounts

Розглянемо папку templates/accounts. Кожна з html сторінок відповідає функціям описаним в Backend частині цього додатку, що знаходяться в views.py.

Розглянемо файли, що мають відношення до користувачів та їх особистих кабінетів:

- Файл details_account.html відкриває сторінку з інформаційними подробицями про особистий кабінет користувача.

- Файл `edit_account.html` відкриває сторінку для редагування подробиць що відображаються в особистому кабінеті.
- Файл `login.html` відкриває сторінку для входу в особистий кабінет користувача.
- Файл `register.html` відкриває сторінку для реєстрації нового користувача.

3.4 Приклад роботи сайту

На наступних зображеннях показано приклад роботи сайту для звичайного користувача.

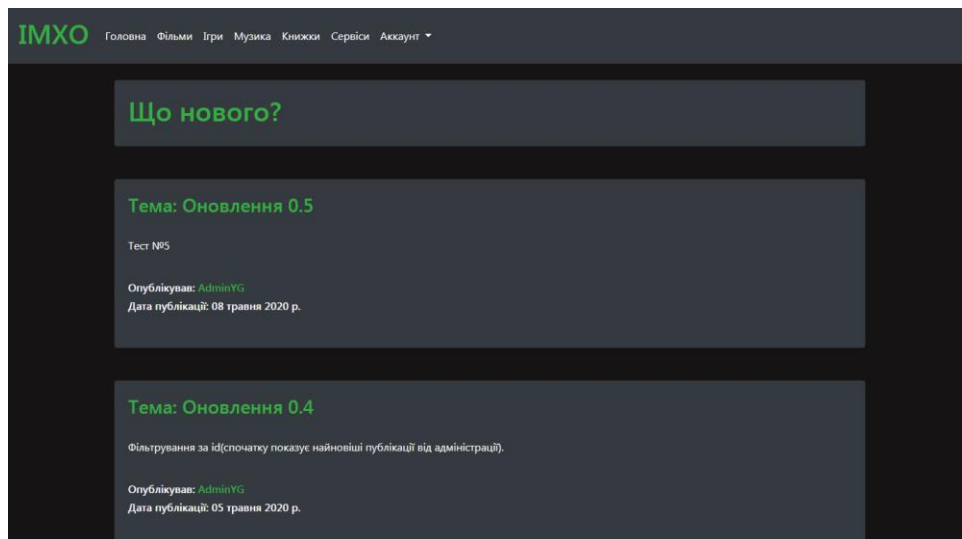


Рисунок 3.29 Головна сторінка сайту

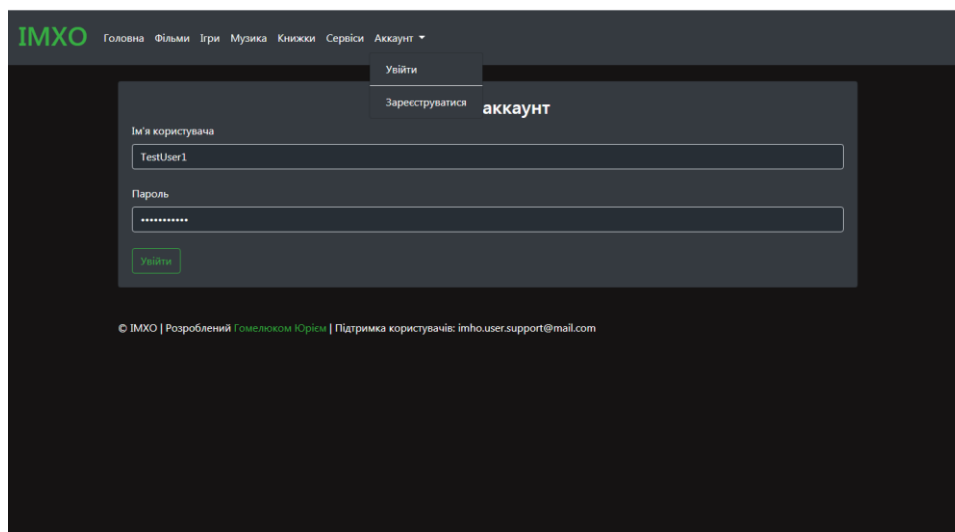


Рисунок 3.30 Вхід в особистий кабінет

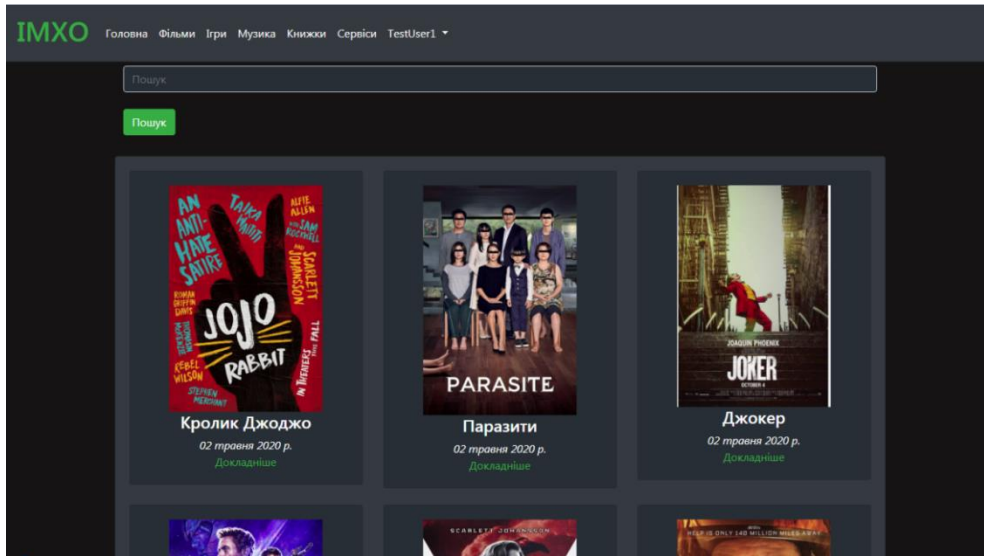


Рисунок 3.31 Сторінка категорії фільмів

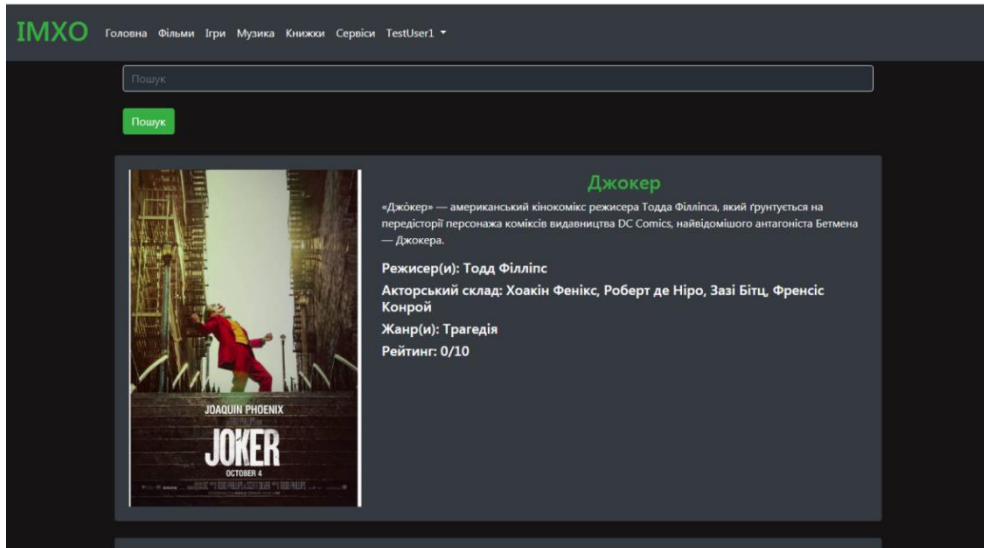


Рисунок 3.32 Сторінка подробиць одного із фільмів

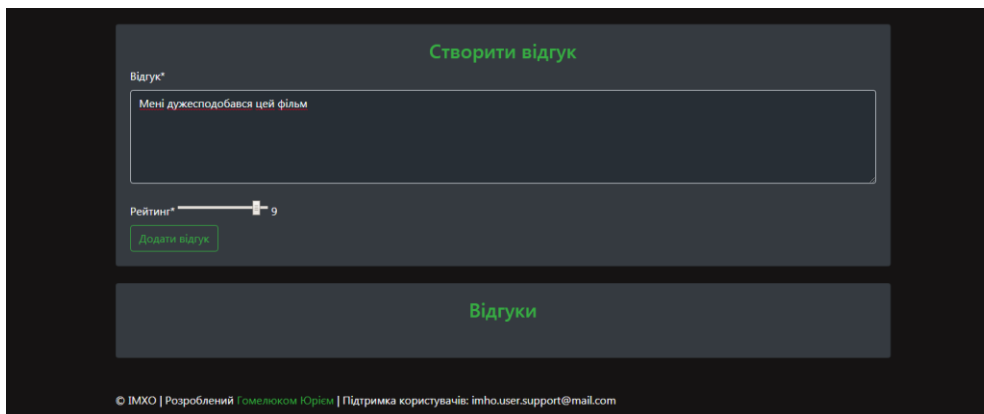


Рисунок 3.33 Створення відгуку на фільм

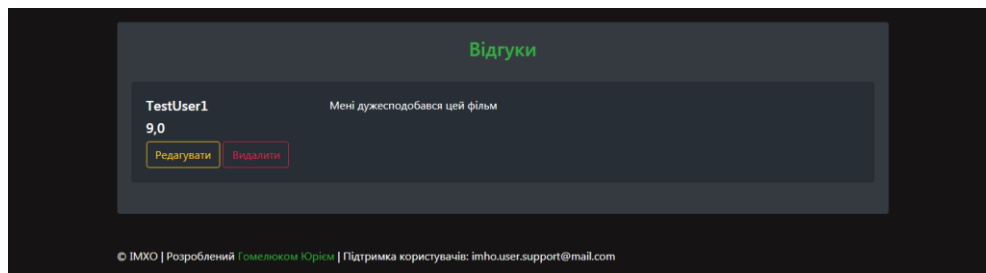


Рисунок 3.34 Відображення створеного відгуку на фільм

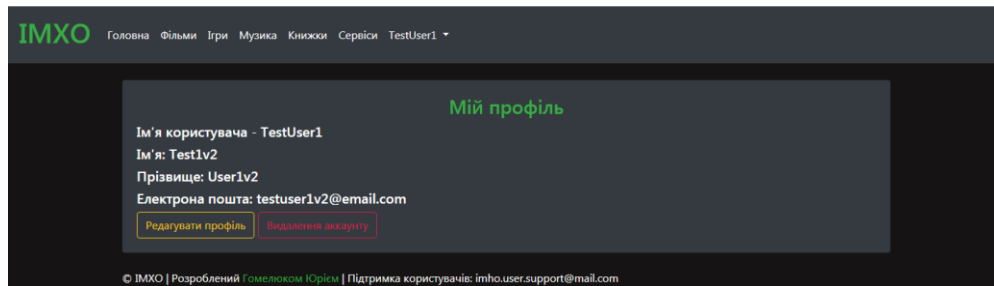


Рисунок 3.35 Сторінка подробиць особового кабінету

ВИСНОВКИ

Був запропонований та розроблений веб-сайт для відгуків на базі Pythonта з використанням фреймворкаDjango. Сайт містить такі функціонал як:

- Перегляд 5-ти категорії для відгуків(фільми, ігри, музика, книжки та сервіси) та 1-ї загальної категорій новин сайту.
- Реєстрація користувачів.
- Можливість залишати відгуки на предмети відгуків що знаходяться у відповідних категоріях.
- Можливість для адміністрації сайту додавати, редагувати та видаляти предмети відгуків та самі відгуки.
- Можливість для адміністрації контролювати БД користувачів сайту.
- Розроблений адміністративний розділ та інтерфейс.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cisco 2018 annual cybersecurity report. URL: https://www.cisco.com/c/dam/m/ru_ru/campaigns/security-hub/pdf/acr-2018.pdf (дата звернення: 08.06.2021).
2. Executive summary cisco 2018 asia pacific security capabilities benchmark study. URL: https://www.cisco.com/c/dam/global/en_ru/products/pdfs/executive_summary_cisco_2018_asia_pacific_Security_capabilities_benchmark_study.pdf (дата звернення: 08.06.2021).
3. PandaLabs 2018 annual report. URL: https://partnernews.pandasecurity.com/uk/src/uploads/2018/12/PandaLabs-2018_Annual_Report-uk.pdf (дата звернення: 08.06.2021).
4. Vacca J. R. Computer and information security handbook. Amsterdam; Boston : Elsevier, 2009.
5. Specht D. F. Generation of Polynomial Discriminant Functions for Pattern Recognition. IEEE Transactions on Electronic Computers. 1967. EC-16, № 3. С. 308–319. URL: <https://doi.org/10.1109/pgec.1967.264667> (дата звернення: 09.06.2021).
6. What is windows forms - windows forms .NET. Developer tools, technical documentation and coding examples | Microsoft Docs. URL: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-5.0#:~:text=Windows%20Forms%20is%20a%20UI,easy%20to%20build%20desktop%20apps>. (дата звернення: 02.06.2021).
7. A detailed analysis of the KDD CUP 99 data set / M. Tavallae та ін. 2009 IEEE symposium on computational intelligence for security and defense applications (CISDA), м. Ottawa, ON, Canada, 8—10 лип. 2009 р. 2009. URL: <https://doi.org/10.1109/cisda.2009.5356528> (дата звернення: 08.06.2021).
8. Özgür A., Erdem H. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ preprints. 2016. С. 1. URL: <https://peerj.com/preprints/1954/> (дата звернення: 08.06.2021).

9. Baestaens D. E. Neural network solutions for trading in financial markets. London : Financial Times, 1994. 245 с.
10. Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives / A. Divekar та ін. 2018 IEEE 3rd international conference on computing, communication and security (ICCCS), м. Kathmandu, 25—27 жовт. 2018 р. 2018. URL: <https://doi.org/10.1109/cccs.2018.8586840> (дата звернення: 08.06.2021).
11. Chaudhuri S., Madigan D., Fayyad U. Kdd-99. ACM SIGKDD explorations newsletter. 2000. Т. 1, № 2. С. 49—51. URL: <https://doi.org/10.1145/846183.846194> (дата звернення: 02.06.2021).
12. Vinod Nair, Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines // 27th International Conference on International Conference on Machine Learning. — USA: Omnipress, 2010. — С. 807–814. — (ICML'10).
13. Savchenko A. Probabilistic neural network with complex exponential activation functions in image recognition. IEEE transactions on neural networks and learning systems. 2020. Т. 31, № 2. С. 651—660. URL: <https://doi.org/10.1109/tnnls.2019.2908973> (дата звернення: 08.06.2021).
14. Kyurkchiev N., Markov S. Sigmoidal functions: some computational and modelling aspects. Biomath communications. 2015. Т. 1, № 2. URL: <https://doi.org/10.11145/j.bmc.2015.03.081> (дата звернення: 08.06.2021).
15. Mitchell T. M. Machine learning. New York : McGraw-Hill, 1997. 414 с.
16. Probabilistic logics and probabilistic networks / J. Williamson та ін. Springer, 2013. 155 с.
17. Probabilistic and General Regression Neural Networks. DTREG. URL: <http://www.dtreg.com/pnn.htm> (дата звернення: 07.06.2021).