

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА
Навчально-науковий інститут енергетичної, інформаційної
та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Пояснювальна записка
до кваліфікаційної роботи бакалавра

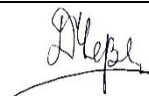
на тему: «Створення вебдодатку для рецензування кінострічок»

Виконав: здобувач вищої освіти,
групи КН 2021-1
спеціальності
122 Комп'ютерні науки

(шифр і назва спеціальності)

Денис ЧЕРНИШ

(ім'я та прізвище)



Керівник: Марина НОВОЖИЛОВА

(ім'я та прізвище)



Рецензент: Микола КАРПЕНКО

(ім'я та прізвище)



м. Харків – 2025 рік

Харківський національний університет міського господарства імені О. М. Бекетова

(повне найменування закладу вищої освіти)

Навчально-науковий Інститут енергетичної, інформаційної

та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КНтаІТ



Марина НОВОЖИЛОВА

« 25 » 06 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Черниша Дениса Олександровича

(прізвище, ім'я, по батькові)

1. Тема роботи «Створення вебдодатку для рецензування кінострічок»

керівник роботи Новожилова Марина Володимирівна

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

затверджені наказом закладу вищої освіти від 09 травня 2025 р. №341-03

2. Термін подання студентом роботи 27 червня 2025 р.









3. Вихідні дані до роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): 1. Аналітична частина 2. Інформаційно-математична частина 3. Програмно-технічна частина 4. Охорона праці

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

Презентація – 16 слайдів


6. Консультанти розділів роботи

Розділ	Ім'я та Прізвище, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Марина НОВОЖИЛОВА, д.ф.-м.н., професор кафедри КН та ІТ	12.05.2025 	20.05.2025 
2	Марина НОВОЖИЛОВА, д.ф.-м.н., професор кафедри КН та ІТ	20.05.2025 	28.05.2025 
3	Марина НОВОЖИЛОВА, д.ф.-м.н., професор кафедри КН та ІТ	28.05.2025 	05.06.2025 
4	Вікторія МАЛИШЕВА, к. т. н., доцент кафедри ОП та БЖ	21.05.2025 	03.06.2025 

7. Дата видачі завдання 12.05.2025**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми дипломної роботи	05.05.2025	виконано
2	Затвердження тем, наукових керівників, завдань та календарного плану підготовки кваліфікаційної роботи	09.05.2025	виконано
3	Написання I розділу	20.05.2025	виконано
4	Написання II розділу	28.05.2025	виконано
5	Написання III розділу	05.06.2025	виконано
6	Написання IV розділу	09.06.2025	виконано
7	Подання дипломної роботи керівнику	10.06.2025	виконано
8	Робота по усуненню зауважень керівника, уточнення і доповнення практичного матеріалу, оформлення додатків до роботи	16.06.2025	виконано
9	Подання доопрацьованого варіанту роботи керівнику	18.06.2025	виконано
10	Захист матеріалів дипломної роботи на засіданні кафедри	20.06.2025	виконано
11	Офіційний захист матеріалів дипломної роботи на засіданні екзаменаційної комісії	28.06.2025	виконано


Студент


 (підпис)

Денис ЧЕРНИШ

(прізвище та ініціали)

Керівник роботи


 (підпис)

Марина НОВОЖИЛОВА

(прізвище та ініціали)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка кваліфікаційної роботи бакалавра студента групи КН 2021-1 спеціальності 122 «Комп'ютерні науки» Черниша Дениса Олександровича за темою «Створення веб-додатку для рецензування кінострічок» складається з 4 розділів, містить 75 сторінок тексту, 57 рисунків, 1 таблицю, 26 джерел.

Дана робота присвячена актуальній темі – створенню веб-додатку для рецензування фільмів, що дозволяє користувачам залишати відгуки та оцінки, формувати персоналізовані добірки.

Мета дослідження – розробка функціонального веб-додатку для рецензування та оцінювання кінострічок.

Об'єктом дослідження є процеси взаємодії користувачів з системами онлайн-рецензування фільмів.

Предметом дослідження є побудова веб-додатку, що автоматизує процес створення, перегляду та управління рецензіями.

У першому розділі виконано постановку задачі та проведено аналітичний огляд існуючих платформ для рецензування фільмів.

У другому розділі описано архітектуру веб-додатку, обґрунтовано вибір технологій і засобів розробки.

У третьому розділі реалізовано інтерфейс користувача, функціонал створення рецензій, реєстрацію, авторизацію та систему оцінювання.

У четвертому розділі розглянуто питання забезпечення інформаційної безпеки та ергономіки веб-додатку.

Ключові слова: ВЕБДОДАТОК, РЕЦЕНЗІЯ, КОРИСТУВАЧ, ІНТЕРФЕЙС, ОЦІНКА.

ANNOTATION

Structure and scope of the work. The explanatory note of the bachelor's qualification work by student Denys Oleksandrovysh Chernysh, group KN 2021-1, specialty 122 "Computer Science", on the topic "Development of a Web Application for Movie Review" consists of 4 chapters and contains 75 pages of text, 57 figures, 1 table, 26 sources.

This work is dedicated to a relevant topic – the development of a web application for movie reviews that allows users to leave feedback and ratings, as well as create personalized selections.

The goal of the research is to develop a functional web application for reviewing and evaluating movies.

The object of the research is the interaction processes between users and online film review systems.

The subject of the research is the construction of a web application that automates the process of creating, viewing, and managing reviews.

The first chapter sets the task and presents an analytical overview of existing movie review platforms.

The second chapter describes the architecture of the web application and justifies the choice of development technologies and tools.

The third chapter implements the user interface, the functionality for writing reviews, registration, authentication, and the rating system.

The fourth chapter covers issues of information security and the ergonomics of the web application.

Keywords: WEB APPLICATION, REVIEW, USER, INTERFACE, RATING.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	10
1.1 Опис предметного середовища.....	10
1.1.1 Опис процесу діяльності	11
1.1.2 Опис функціональної моделі	12
1.2 Огляд наявних аналогів	13
1.3 Постановка задачі	19
Висновки до розділу	20
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	22
2.1 Аналіз предметної області	22
2.1.1 Вхідні дані	23
2.1.2 Вихідні дані	23
2.2 Проектування системи.....	24
2.2.1 Проектування бази даних.....	24
2.2.2 Побудова об'єктно-орієнтованої моделі	26
2.3 Математичне та алгоритмічне забезпечення	31
Висновки до розділу	35
РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	37
3.1 Засоби розробки	37
3.2 Вимоги до технічного та програмного забезпечення.....	39
3.3 Опис програмної реалізації.....	40
3.3.1 Середовище виконання	41
3.3.2 Безпека доступу.....	42

	7
3.3.3 Реалізація бази даних та взаємодії між компонентами проєкту	44
3.3.4 Деплой сайту та підключення AWS для збереження медіа	51
3.4 Керівництво користувача	56
Висновки до розділу	60
РОЗДІЛ 4 ОХОРОНА ПРАЦІ	62
4.1 Регулювання питань охорони праці на законодавчому рівні	62
4.2 Виявлення потенційних небезпек стосовно об'єкту проєктування.....	63
4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проєктування та розробка заходів щодо їх попередження	66
Висновки до розділу	69
ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72

ВСТУП

У зв'язку з постійним зростанням обсягів відеоконтенту та популярністю потокових сервісів, потреба в якісних онлайн-платформах для рецензування фільмів стає дедалі актуальнішою. Користувачі прагнуть не лише переглядати кінострічки, а й ділитися враженнями, оцінками, читати думки інших. Наявні сервіси не завжди забезпечують зручність, відкритість до локальної аудиторії чи сучасний інтерфейс. Тому створення спеціалізованого вебдодатку для рецензування кінострічок є актуальним кроком до розвитку глядацької культури та цифрової взаємодії.

Мета дослідження – розробка функціонального вебдодатку для рецензування та оцінювання кінострічок.

Для досягнення поставленої мети необхідно вирішити такі основні задачі:

- провести аналіз предметної області з метою виявлення ключових функціональних вимог до системи;
- спроектувати архітектуру інформаційної системи;
- обґрунтувати вибір засобів реалізації програмного продукту;
- реалізувати інформаційну систему згідно з технічними вимогами;
- провести тестування програмного продукту для виявлення та усунення помилок;
- оцінити ефективність розробленого рішення та можливості його масштабування або адаптації до інших задач.

Об'єктом дослідження є процеси взаємодії користувачів з системами онлайн-рецензування фільмів.

Предметом дослідження є побудова вебдодатку, що автоматизує процес створення, перегляду та управління рецензіями.

У дослідженні використано методи аналізу вимог до програмного забезпечення, структурного та об'єктно-орієнтованого проектування,

розробки вебзастосунків із використанням сучасних фреймворків, а також функціонального тестування.

Теоретична цінність роботи полягає у формалізації вимог до інформаційних систем такого типу та дослідженні принципів створення зручного і масштабованого інтерфейсу користувача.

Прикладна значущість результатів полягає в можливості використання розробленого вебдодатку як готового рішення або основи для подальших удосконалень та адаптацій в галузі онлайн-рецензування й спільнот користувачів.

Результатом роботи стане працездатний вебдодаток, що забезпечує користувачам можливість створювати, переглядати та оцінювати рецензії на фільми, а також взаємодіяти через коментарі та рейтингову систему. Реалізований функціонал дозволяє не лише покращити досвід користувача, але й створює підґрунтя для подальшого розвитку системи як повноцінної платформи для кінокритики та обміну думками в онлайн-середовищі.

Результати дослідження пройшли апробацію на II Міжнародній науково-практичній конференції «ІТ: теорія та практика» [1].

РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Сучасний розвиток інформаційних технологій суттєво трансформував підходи до споживання та обговорення кіноіндустріального контенту. Все більше користувачів не лише переглядають фільми, а й активно висловлюють свої думки про побачене у вигляді рецензій, оцінок, коментарів та рекомендацій. Позитивні рецензії впливають на стимулювання продажу квитків і сприяють комерційному успіху фільму, тоді як негативні відгуки можуть відлякувати потенційних глядачів [2, с. 3534]. Це сформувало сталу потребу у створенні спеціалізованих веб-додатків, які дозволяють структурувати, систематизувати та зручно представляти думки користувачів щодо кінострічок.

Рецензування та обговорення фільмів на онлайн-платформах є важливою складовою сучасної кіноіндустрії. Щомісячна аудиторія таких платформ, як IMDb, становить близько 250 мільйонів активних користувачів [3], що свідчить про високий рівень інтересу до можливостей оцінювання, коментування і персоналізації перегляду кінострічок. Водночас сучасні платформи стикаються з викликами щодо підвищення зручності інтерфейсів, розширення соціальної взаємодії між користувачами, а також забезпечення достовірності та якості контенту. Врахування цих аспектів є необхідним для створення конкурентоспроможного продукту, який відповідатиме очікуванням аудиторії.

Незважаючи на широку популярність платформ для рецензування фільмів, існують певні проблеми, які залишаються невирішеними. Зокрема, складний та неінтуїтивний інтерфейс ускладнює процес написання та пошуку рецензій. Крім того, обмежена гнучкість у персоналізації

користувацького досвіду не дає змоги враховувати індивідуальні вподобання при формуванні рекомендацій, а недостатній рівень соціальної взаємодії між користувачами призводить до зниження їх залученості на платформах. Водночас проблеми з модерацією контенту та можливість появи фейкових відгуків ставлять під сумнів довіру до наявної інформації.

1.1.1 Опис процесу діяльності

На сучасних платформах для рецензування кінострічок користувачі зазвичай очікують інтуїтивний та швидкий доступ до основних функцій, таких як реєстрація, авторизація, перегляд інформації про фільми, написання рецензій, оцінювання, створення персональних списків перегляду та отримання рекомендацій. Важливим аспектом є також можливість взаємодії між користувачами через коментарі та інші соціальні функції, а також надійне збереження особистих даних і медіаконтенту.

Процес діяльності користувача у розроблюваному вебдодатку для рецензування кінострічок розпочинається з авторизації, що включає реєстрацію нового користувача або вхід за існуючими обліковими даними. Авторизація забезпечує безпечний доступ до персоналізованих функцій платформи, гарантуючи захист особистих даних і підтримку сесій користувача.

Після авторизації користувачеві стають доступними такі дії:

- створення рецензій та виставлення оцінок для кінострічок;
- перегляд існуючих рецензій та інформації про фільми;
- керування обліковим записом;
- завантаження медіафайлів, таких як постери, аватари користувачів і трейлери фільмів.

Загалом процес діяльності користувача у вебдодатку можна зобразити у вигляді UML-діаграми діяльності (рис. 1.1):

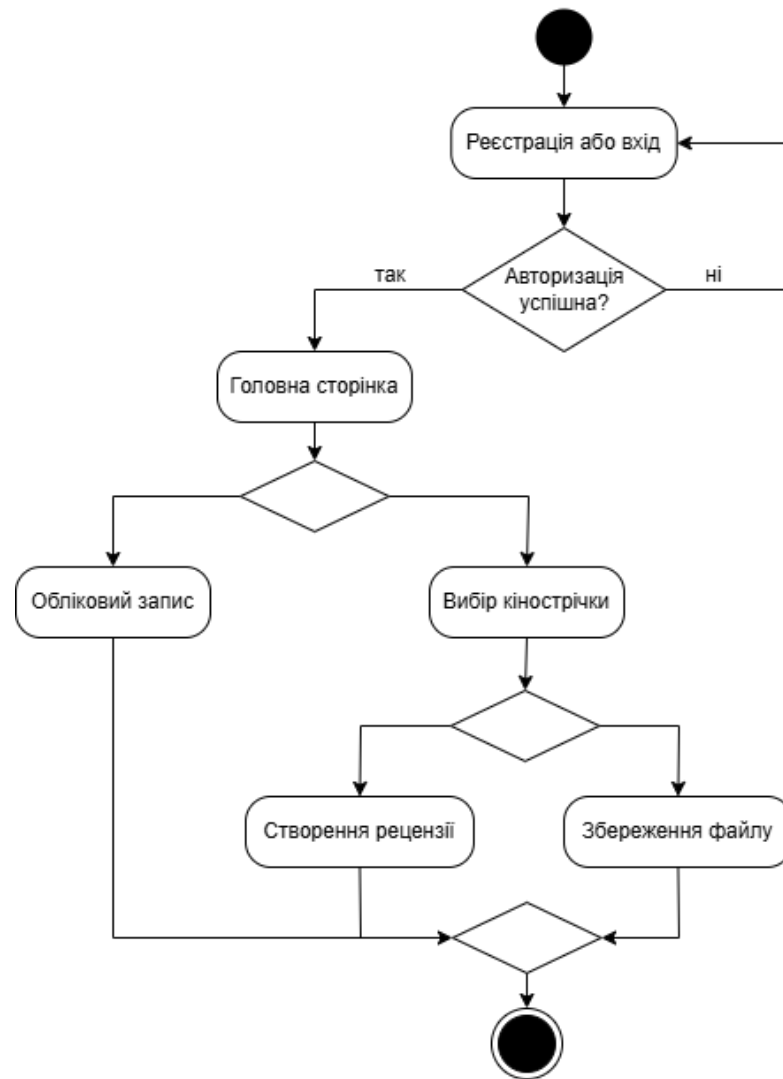


Рисунок 1.1 – UML-діаграма діяльності користувача у вебдодатку

1.1.2 Опис функціональної моделі

Функціональна модель системи відображає основні дії, які можна виконувати у веб-додатку для рецензування кінострічок. Єдиним актором є користувач, який має можливість здійснювати широкий спектр операцій, що забезпечують повноцінну взаємодію із системою.

Користувач може зареєструватися або увійти до системи, після чого йому стають доступні функції перегляду інформації про фільми та існуючих рецензій, створення, редагування та видалення власних рецензій, а також виставлення оцінок кінострічкам. Окрім цього, користувач має змогу керувати своїм обліковим записом, а також завантажувати та переглядати медіафайли, такі як постери та трейлери фільмів.

UML-діаграма використання ілюструє ключові функціональні можливості, доступні користувачеві (рис. 1.2).

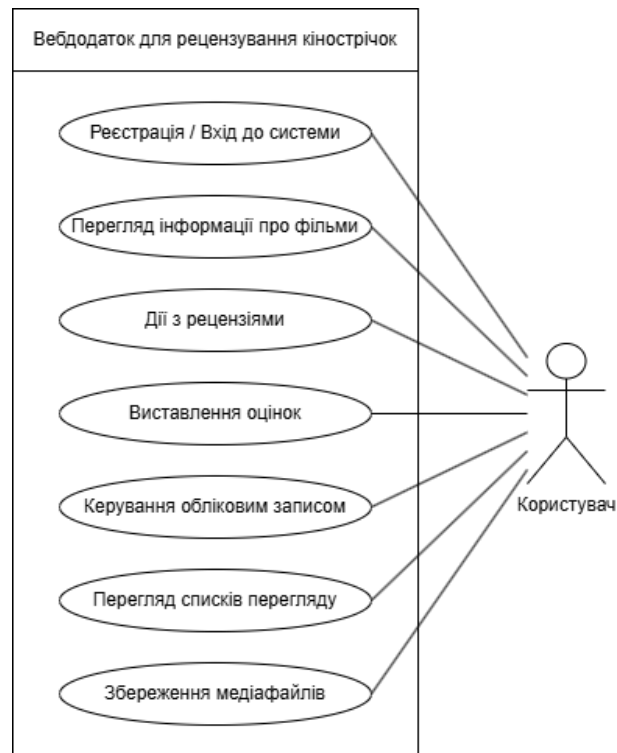


Рисунок 1.2 – UML-діаграма використання функцій у вебдодатку

Обравши фільм, користувач має можливість ознайомитися з інформацією про нього, побачити оцінки та рецензії інших користувачів або ж залишити свої. У власному обліковому записі користувач має можливість створити запис кінострічки чи виконати дії з рецензіями, які передбачають створення нової рецензії та виставлення оцінки або її редагування чи видалення. Крім того, наявна можливість збереження медіафайлів, таких як аватари, постери чи трейлери.

1.2 Огляд наявних аналогів

У рамках дослідження було проведено аналіз наявних веб-платформ, які надають користувачам функціональність для оцінювання, коментування та обговорення фільмів. До розгляду взято такі відомі платформи, як IMDb [4],

Letterboxd [5] та Rotten Tomatoes [6]. Усі вони орієнтовані на масову аудиторію, забезпечують базову функціональність з оцінювання та перегляду інформації про фільми, але відрізняються деталізацією інтерфейсу, гнучкістю персоналізації та можливостями взаємодії між користувачами.

IMDb – одна з найстаріших і найбільших платформ, що забезпечує доступ до структурованої бази даних про кіно- й телевізійну продукцію, включаючи інформацію про фільми, серіали, акторів, режисерів та інші пов’язані елементи.

Функціональні можливості платформи IMDb для рецензування кінострічок включають:

- оцінювання фільмів за 10-бальною шкалою;
- написання користувацьких рецензій з відкритим доступом;
- перегляд оцінок і відгуків інших користувачів;
- базовий інтерфейс для взаємодії з системою оцінювання та рецензування

(рис. 1.3).

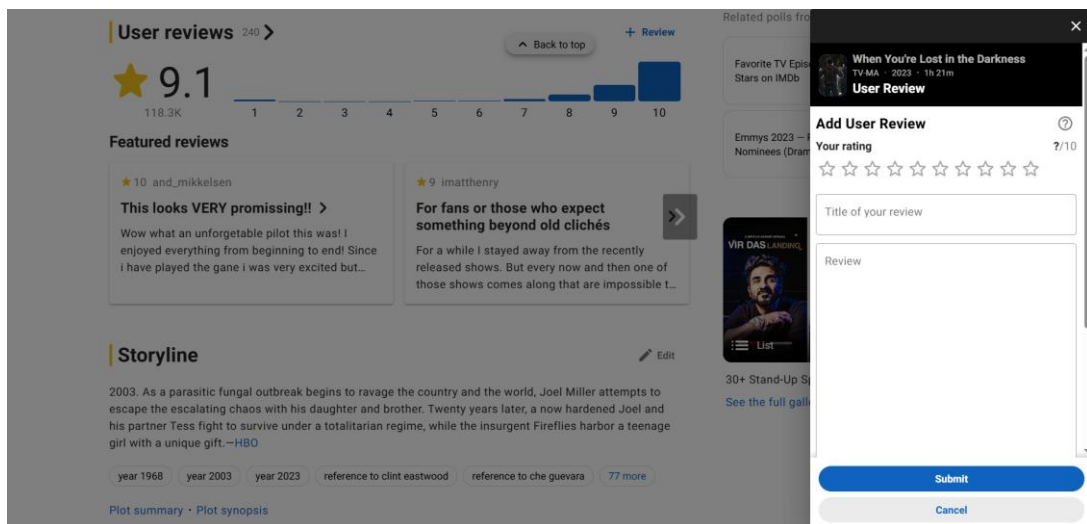


Рисунок 1.3 – Інтерфейс написання відгуку на платформі IMDb

Платформа IMDb вирізняється об’ємною базою даних і активною спільнотою, що сприяє формуванню загальної оцінки кінострічок на основі відгуків користувачів. До переваг можна віднести наявність механізму оцінювання та публікації рецензій, що забезпечує двосторонню взаємодію.

Водночас менш інтуїтивний інтерфейс є недоліком, адже для написання або перегляду відгуків необхідні додаткові кроки.

Letterboxd – платформа, сфокусована на створенні рецензій, веденні щоденників перегляду та списків фільмів, вона має активну спільноту та простий сучасний інтерфейс.

Функціональними можливостями платформи Letterboxd для рецензування кінострічок є:

- оцінювання фільмів за 5-бальною шкалою;
- написання текстових рецензій із можливістю форматування та редагування (рис. 1.4);
- ведення особистого щоденника переглядів із зазначенням дати, вражень і повторних переглядів;
- створення персоналізованих списків фільмів за темами або критеріями на власний розсуд;
- перегляд відгуків, оцінок і списків інших користувачів із можливістю їх коментування чи додавання в друзі.

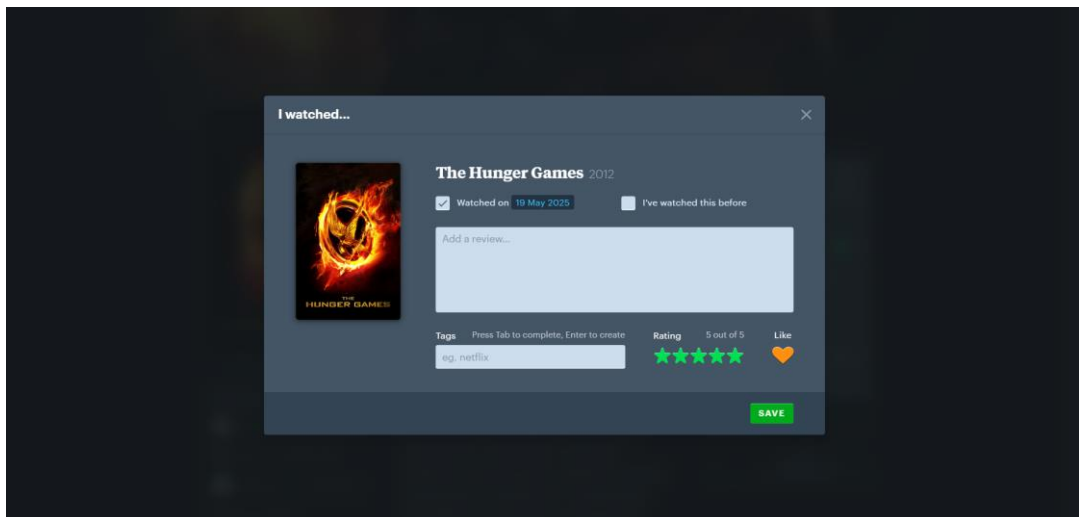


Рисунок 1.4 – Інтерфейс написання відгуку на платформі Letterboxd

Платформа Letterboxd є привабливою для користувачів завдяки сучасному мінімалістичному дизайну, активній спільноті та інструментам для ведення персоналізованої кіноісторії. Її перевагами є зручний механізм

написання рецензій і систематизація переглядів, що сприяє особистій аналітиці переглянутого контенту. Недоліком можна вважати обмежену структурованість бази фільмів порівняно з іншими платформами, а також орієнтацію переважно на кіноманів, що може зменшити привабливість для ширшої аудиторії.

Rotten Tomatoes – платформа, яка спеціалізується на агрегуванні рецензій від професійних кінокритиків і звичайних глядачів. Основною метою ресурсу є формування загального рейтингу фільмів, що відображає як професійну, так і масову думку.

Функціональними можливостями платформи Rotten Tomatoes для рецензування кінострічок є:

- публікація текстових рецензій із зазначенням оцінки за 5-бальною шкалою (рис. 1.5);
- формування рейтингу фільму на основі відгуків сертифікованих критиків;
- окремий рейтинг глядачів, який відображає загальне сприйняття фільму масовою аудиторією;
- можливість ознайомлення з короткими цитатами з рецензій критиків, посиланнями на повні версії, а також фільтрація за джерелом чи оцінкою.

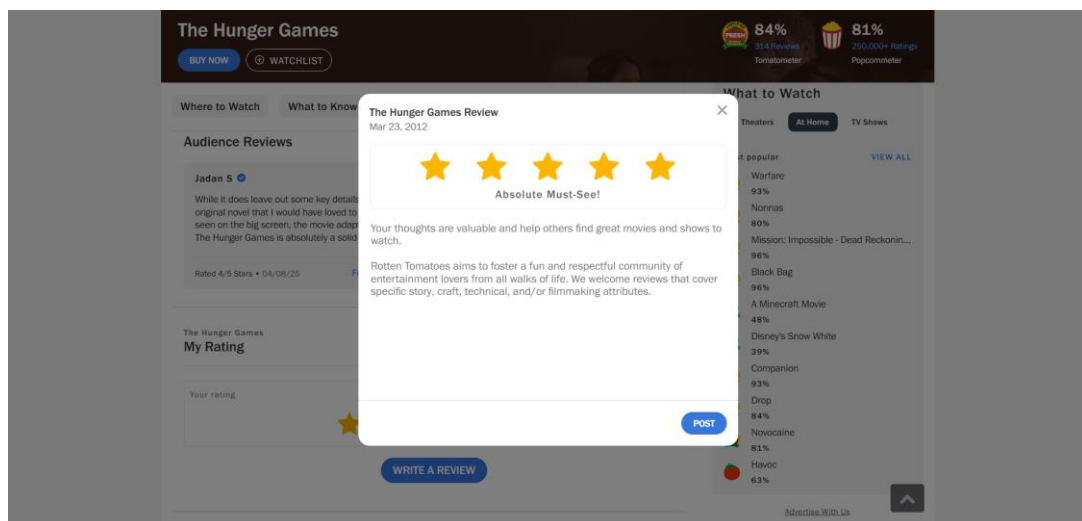


Рисунок 1.5 – Інтерфейс написання відгуку на платформі Rotten Tomatoes

Платформа Rotten Tomatoes є авторитетним джерелом з точки зору агрегування професійної критики, що дозволяє швидко зорієнтуватися в оцінці фільму експертами. До переваг належать поділ рейтингу критиків і глядачів, чітка візуальна подача оцінок і високий рівень довіри серед аудиторії. Водночас функціонал для звичайного користувача обмежений у порівнянні з платформами, орієнтованими на активне рецензування, а система оцінювання є менш гнучкою через двійкову шкалу, оскільки для складання загального рейтингу критиків враховується лише поділ відгуків на позитивні й негативні.

Порівняння функціональних можливостей та особливостей розглянутих платформ наведено у табл. 1.1:

Таблиця 1.1 – Порівняння платформ-аналогів для рецензування кінострічок

Функціональні можливості	IMDb	Letterboxd	Rotten Tomatoes
Оцінювання фільмів	10-бальна шкала	5-бальна шкала	5-бальна шкала оцінки під відгуками, агрегований рейтинг критиків
Написання рецензій	Так	Так	Так
Перегляд оцінок інших користувачів	Так	Так	Так
Спільнота користувачів	Велика, активна	Активна, орієнтована на рецензування	Акцент на професійних критиках, користувацькі оцінки окремо
Інтерфейс	Менш інтуїтивний	Сучасний, простий	Інформаційний, але обмежений для користувачів
Збереження медіаконтенту	Ні (переважно текст і оцінки)	Обмежено (постери, трейлери)	Ні
Механізм рейтингу	Оцінка за 10-бальною шкалою	Оцінка у зірках	Двійкова класифікація критиків та бальна оцінка в рецензіях

З метою оцінки масштабу аудиторії було зібрано та систематизовано дані щодо кількості щомісячних активних користувачів (MAU) для кожної платформи у розрізі кварталів за 2024 рік:

- IMDb – стабільно близько 250 млн MAU;
- Rotten Tomatoes – приблизно 108 млн MAU;
- Letterboxd – близько 14 млн MAU;

Ці дані наочно представлені у вигляді графіка (рис. 1.6):



Рисунок 1.6 – MAU представників предметної області за 2024 рік (поквартально)

Отримані результати свідчать про значне домінування IMDb на ринку за обсягом користувачів, однак це не обов'язково відображає зручність або якість взаємодії з платформою, що є важливим при проєктуванні нового продукту.

Таким чином, аналіз наявних платформ для оцінювання та рецензування фільмів показав, що кожне рішення має свої сильні та слабкі сторони. IMDb вирізняється найбільшою базою користувачів і об'ємною структурованою інформацією, проте менш інтуїтивним інтерфейсом. Letterboxd приваблює сучасним дизайном і зручними інструментами для ведення персональних

щоденників переглядів, але має меншу аудиторію та обмежену структурованість даних. Rotten Tomatoes відзначається високою довірою завдяки агрегуванню оцінок професійних критиків і розділенню рейтингів аудиторії, але обмежує гнучкість оцінювання через спрощену двійкову шкалу у формуванні основного рейтингу.

Враховуючи ці особливості, при розробці нового продукту доцільно поєднати переваги великих баз даних і активної спільноти із сучасним і зручним інтерфейсом, а також забезпечити більш гнучкий і деталізований механізм оцінювання, що дозволить точніше відображати різноманіття користувацьких вражень.

1.3 Постановка задачі

Проектування та реалізація веб-додатку для рецензування кінострічок передбачає створення повноцінної програмної системи з клієнтською, серверною та інфраструктурною складовими. Основний акцент у даній роботі зроблено не на графічному оформленні або дизайні інтерфейсу, а саме на логіці роботи серверної частини, структурі бази даних, обробці запитів, роботі з медіаконтентом та налаштуванні інфраструктури для деплойменту та хостингу.

Метою роботи є реалізація вебдодатку, який забезпечує ключовий функціонал платформи рецензування:

- створення, збереження та перегляд рецензій на кінострічки;
- виставлення оцінок користувачами;
- керування обліковими записами;
- збереження медіафайлів (постери, аватари, трейлери);
- швидке й масштабоване розгортання додатку у продакшн-середовищі.

Для досягнення мети було сформульовано низку технічних задач:

а) створити бекенд-застосунок, що обробляє HTTP-запити, реалізує REST API для клієнтської частини та взаємодіє з базою даних;

- б) розробити структуру реляційної бази даних на PostgreSQL з використанням сервісу Neon як віддаленого хмарного сховища;
- в) забезпечити аутентифікацію та авторизацію користувачів, включаючи підтримку JWT-токенів;
- г) реалізувати CRUD-функціонал для рецензій, фільмів, оцінок, списків користувачів;
- г) інтегрувати зберігання медіафайлів через Amazon Web Services (AWS), використовуючи S3 або інші сервіси для зберігання зображень фільмів і аватарів;
- д) забезпечити деплоймент сайту через платформу Render.com, включно з налаштуванням постійного підключення до бази даних Neon та хмарного зберігання AWS;
- е) оптимізувати запити та серверну логіку, щоб забезпечити стабільну роботу при зростанні кількості користувачів;
- є) передбачити розширювану архітектуру, яка дозволить у майбутньому додати функціонал персоналізованих рекомендацій, коментарів, лайків та взаємодії між користувачами.

Таким чином, поставлена задача охоплює як проектування прикладного рівня додатку, так і реалізацію хмарної інфраструктури. Усі етапи виконання спрямовані на побудову працездатного веб-сервісу, готового до розгортання у реальному середовищі.

Висновки до розділу

У першому розділі було розглянуто загальні положення предметної області веб-додатків для рецензування кінострічок, що є актуальними в умовах зростаючої популярності онлайн-платформ для оцінювання та обговорення фільмів. Проведений аналіз предметного середовища показав, що сучасні користувачі не лише споживають кіноіндустріальний контент, а й активно беруть участь у формуванні думок про нього, що підсилює потребу у зручних

та функціональних інструментах для створення, пошуку та перегляду рецензій. Водночас були визначені ключові проблеми існуючих систем, серед яких – недостатня інтуїтивність інтерфейсів, обмежені можливості соціальної взаємодії, а також питання достовірності контенту.

Також у розділі наведено опис процесу діяльності користувача у веб-додатку, що включає етапи авторизації, взаємодії з рецензіями та медіафайлами, а також управління особистим кабінетом. На основі функціональної моделі було деталізовано ключові операції, які забезпечують повноцінну взаємодію користувача із системою. Огляд існуючих аналогів – IMDb, Letterboxd та Rotten Tomatoes – дозволив виділити переваги та недоліки кожної платформи, що стало основою для визначення вимог і напрямків удосконалення розроблюваного вебдодатку з урахуванням актуальних тенденцій і очікувань користувачів.

РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз предметної області

Розроблена інформаційна система призначена для автоматизації процесу оцінювання фільмів користувачами. Основна мета – надати зручний вебінтерфейс для взаємодії з кіноконтентом: переглядом фільмів, залишенням відгуків, виставленням рейтингу, додаванням до списку бажаного перегляду (wishlist), керуванням профілем користувача.

Система реалізує такі ключові функції:

- перегляд каталогу фільмів із описом, постером, роком випуску та жанрами;
- виставлення оцінок та написання описового коментаря;
- додавання фільмів у персональний wishlist;
- реєстрація користувача, вхід до особистого кабінету, зміна аватара;
- доступ до системи адміністрування для керування вмістом.

Проект орієнтований на кіноманів, які хочуть вести власну фільмотеку з оцінюванням контенту, а також розробників, що досліджують реалізацію подібних сервісів на базі Django.

Додатково, система є адаптованою для подальшого масштабування – наприклад, додавання рекомендаційної системи або функцій соціальної взаємодії (лайки, коментарі до відгуків тощо).

У межах предметної області розглядається процес взаємодії користувача з вебсервісом для оцінювання фільмів, основними учасниками якого є зареєстровані користувачі та адміністратор.

Основні поняття предметної області включають:

- фільм – об’єкт, який має назву, опис, рік випуску, жанри, постер, кожен фільм може мати багато оцінок від різних користувачів;

- користувач – зареєстрована особа, яка має можливість оцінювати фільми, залишати відгуки, додавати фільми до списку бажаного перегляду та редагувати персональні дані;

- оцінка – числове значення в діапазоні від 0 до 10, яке задається фільму разом з описовим коментарем;

- вибране – список фільмів, які користувач позначив як бажані до перегляду;

- жанр – категорія, до якої належить фільм (драма, комедія тощо);

- адміністратор – користувач, що має доступ до Django admin-панелі для керування контентом: додавання, редагування та видалення фільмів, жанрів, користувачів.

2.1.1 Вхідні дані

Вхідними даними є:

- логін, пароль, e-mail та ім'я, які вводяться у процесі реєстрації та авторизації;

- зображення аватара (типи .jpg, .png), які завантажуються під час формування профілю;

- текстові поля, які заповнюються під час додавання фільмів, зокрема: назва, опис, жанр, файл постеру;

- відгуки на фільми, які включають оцінку від 0 до 10 та текстовий коментар;

- фільтрація за жанром чи назвою;

- взаємодія з wishlist – натискання кнопки додавання/видалення з переліку бажаного до перегляду.

2.1.2 Вихідні дані

Вихідні дані включають:

- список фільмів, відображення постерів, опису, рейтингу;

- персональні дані, аватар, wishlist в особистому профілі;

- повідомлення про успішне створення або редагування записів, помилки авторизації;
- динамічні списки – результати фільтрації;
- рейтинг фільмів у вигляді зірок та залишених коментарів;
- доступ до функціоналу – відображення кнопок редагування, якщо користувач має відповідні права.

2.2 Проєктування системи

2.2.1 Проєктування бази даних

Проєктування бази даних відбувається на основі визначених сутностей системи, які відображають ключові об'єкти предметної області, за ними формуються таблиці наявними полями, які представляють атрибути. На діаграмі можна побачити структуру бази даних вебдодатку з наявними зв'язками (рис. 2.1).

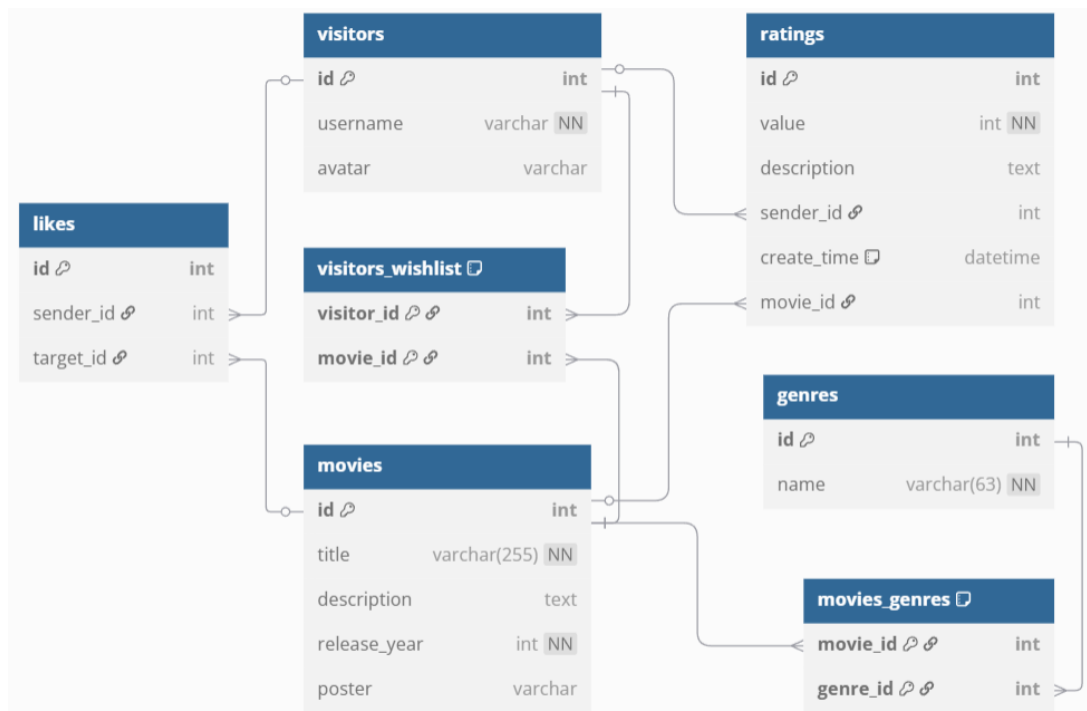


Рисунок 2.1 – ER-діаграма зв'язків між основними сутностями системи

Таблиця `visitors` зберігає інформацію про користувача платформи, де `id` – це первинний ключ (унікальний ідентифікатор кожного користувача), `username` – логін, створений під час реєстрації, а `avatar` – зображення особистого профілю, збережене як шлях до медіафайлу.

Таблиця `visitors_wishlist` призначена для збереження зв'язків між користувачами та фільмами, які вони додали до списку бажаного, її полями є `visitor_id` – ідентифікатор користувача – і `movie_id` – ідентифікатор фільму.

Дані таблиці поєднуються через зовнішні ключі, типом зв'язку є багато-до-багатьох.

Таблиця `likes` являє собою позначку вподобання фільму користувачем, вона містить наступні поля:

- `id` – первинний ключ;
- `sender_id` – зовнішній ключ до таблиці `visitors`, що вказує, хто вподобав фільм;
- `target_id` – вказує, якому фільму поставлено лайк.

Таблиця `movies` – це основна таблиця для збереження інформації про фільми, вона налічує такі поля:

- `id` – унікальний ідентифікатор фільму;
- `title` – назва фільму;
- `description` – короткий опис сюжету кінострічки;
- `release_year` – рік виходу;
- `poster` – шлях до зображення постера.

За полем `target_id` реалізується зв'язок один-до-багатьох між `movies` та `likes` (один фільм може мати багато вподобань). Таблиця `visitors_wishlist` також пов'язана з `movies` через через поле `movie_id` типом зв'язку багато-до-багатьох.

Для збереження наявних жанрів використовується таблиця `genres`, в якій `id` представляє унікальний ідентифікатор жанру, а `name` – його назву.

Таблиця `movies_genres` створена для збереження зв'язку між фільмами та жанрами, її полями є `movie_id`, тобто зовнішній ключ на таблицю `movies`, за

яким реалізується зв'язок типу багато-до-багатьох, та `genre_id` – зовнішній ключ на таблицю `genres`, який також реалізує зв'язок багато-до-багатьох.

Таблиця `ratings` зберігає всі дані про відгуки користувачів, її полями є:

- `id` – унікальний ідентифікатор відгуку;
- `value` – числова оцінка (від 0 до 10);
- `description` – текстовий коментар до оцінки;
- `sender_id` – зовнішній ключ на `visitors`, що вказує, хто залишив оцінку (зв'язок типу один-до-багатьох);
- `create_time` – дата й час створення відгуку;
- `movie_id` – зовнішній ключ на `movies`, який вказує, якому фільму належить оцінка (зв'язок типу один-до-багатьох).

Таким чином, база даних підтримує логічну, нормалізовану структуру зі збереженням зв'язків між основними сутностями системи: користувачами, фільмами, жанрами, оцінками та вподобаннями.

2.2.2 Побудова об'єктно-орієнтованої моделі

Структура застосунку включає такі логічні шари:

- моделі (`models`), які відповідають за опис структури даних та зв'язків;
- шаблони (`templates`), створені для HTML-візуалізації інтерфейсу;
- представлення (`views`), що реалізують логіку обробки HTTP-запитів;
- форми (`forms`), створені для обробки введення з форм;
- адміністративна панель для керування контентом.

У проєкті реалізована класична схема розділення логіки за принципом `Model–Template–View` (MTV), яка забезпечує ефективну взаємодію між даними, бізнес-логікою та відображенням інтерфейсу.

Моделі визначають структуру даних, з якою працює застосунок:

- `Visitor` – наслідує `AbstractUser`, являє собою користувача платформи, має персональні дані, `wishlist`, аватар, ролі;
- `Movie` – фільм із назвою, описом, роком виходу, постером і жанрами;

- Genre – жанр, до якого належить фільм (їх може бути кілька);
 - Rating – оцінка та опис від користувача для певного фільму;
 - Like – відмітка «подобається» від користувача для певного фільму;
- Дану структуру можна відобразити у вигляді діаграми (рис. 2.2).

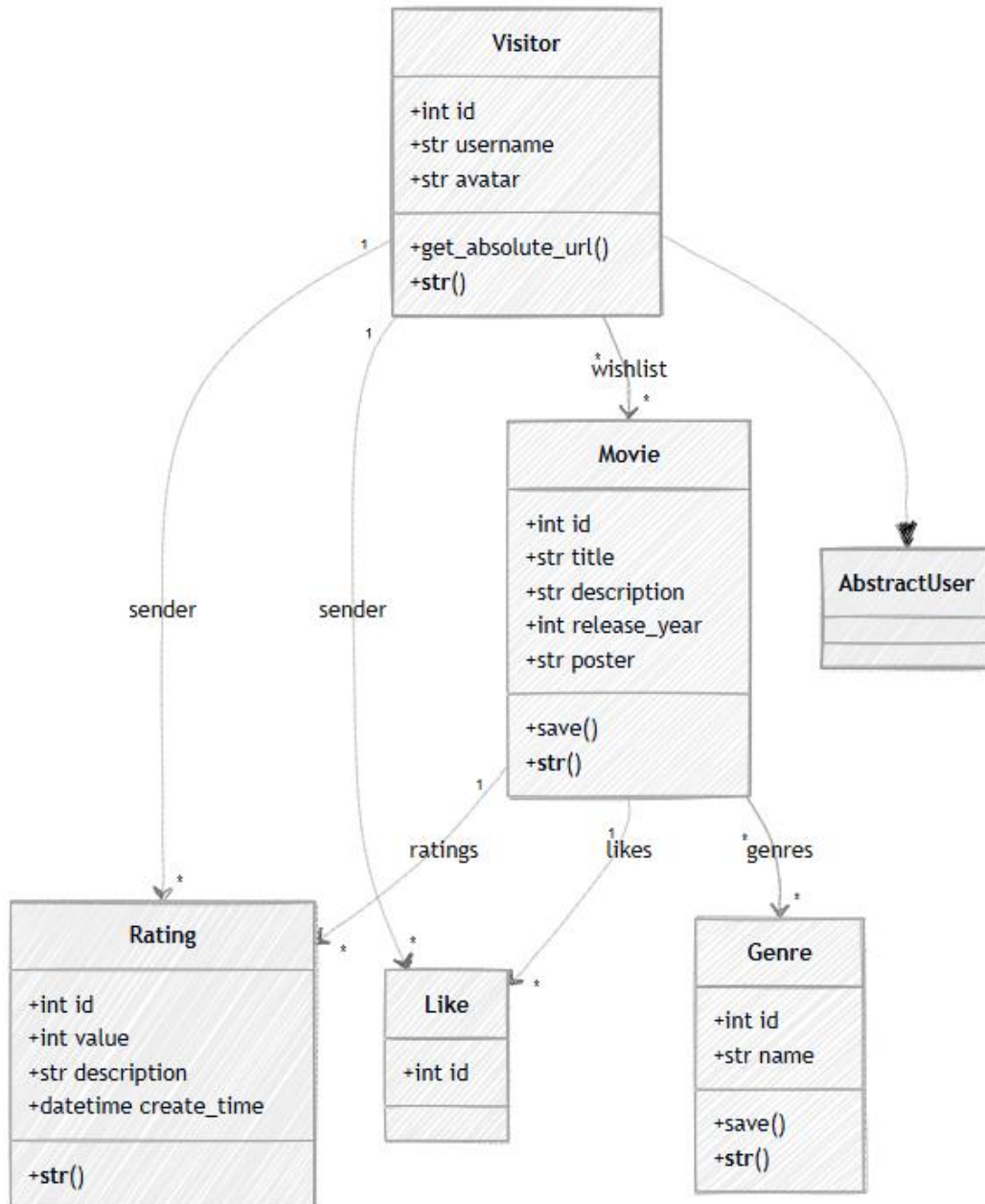


Рисунок 2.2 – UML-діаграма класів моделей

Також в роботі наявні класи представлень, які відображено на діаграмі на рис. 2.3.

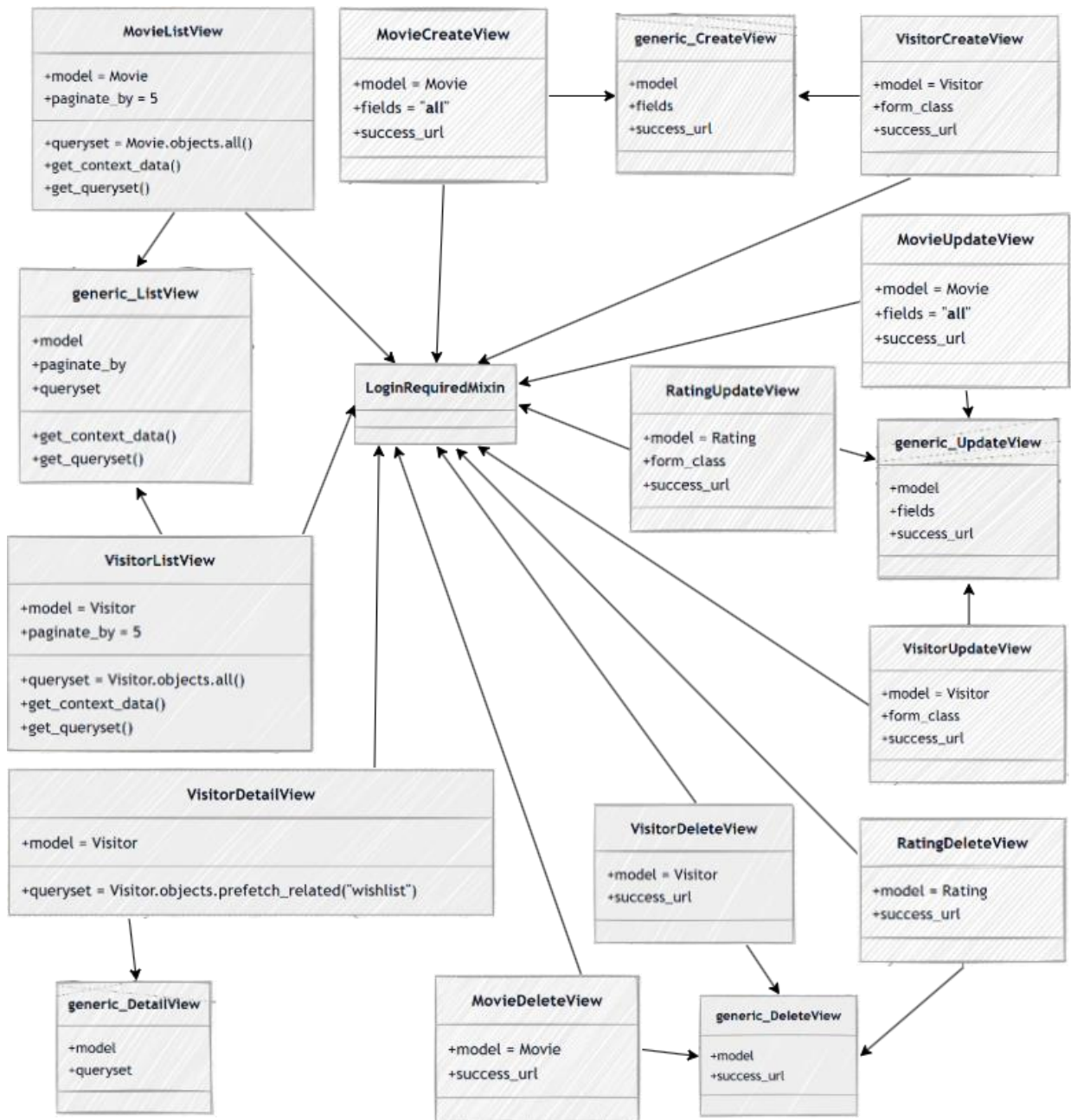


Рисунок 2.3 – UML-діаграма класів представлень

Призначенням представлень є обробка HTTP-запитів й повернення відповідних відповідей.

Класи для фільмів включають наступні:

– `MovieListView` (`LoginRequiredMixin`, `generic.ListView`) – представлення для відображення списку фільмів із можливістю пошуку та фільтрації, успадковує захист і функціонал списку;

– `MovieCreateView` (`LoginRequiredMixin`, `generic.CreateView`) – дозволяє автентифікованим користувачам створювати нові записи фільмів, успадковує створення об'єктів із перевіркою доступу;

– `MovieUpdateView` (`LoginRequiredMixin`, `generic.UpdateView`) – дає змогу редагувати існуючі фільми, наслідує механізм оновлення даних з обмеженням доступу;

– `MovieDeleteView` (`LoginRequiredMixin`, `generic.DeleteView`) – представлення для видалення фільмів зі списку, наслідує логіку видалення об'єктів.

Класами для користувачів є:

– `VisitorListView` (`LoginRequiredMixin`, `generic.ListView`) – список відвідувачів з функцією пошуку за іменем, наслідує доступ лише для авторизованих користувачів і логіку перегляду списку;

– `VisitorDetailView` (`LoginRequiredMixin`, `generic.DetailView`) – детальна інформація про список бажаних фільмів користувача, забезпечує перегляд одного об'єкта;

– `VisitorCreateView` (`LoginRequiredMixin`, `generic.CreateView`) – форма створення нового користувача системи, використовується адміністратором або системою;

– `VisitorUpdateView` (`LoginRequiredMixin`, `generic.UpdateView`) – дозволяє редагувати дані користувача, успадковує оновлення моделей з перевіркою прав;

– `VisitorDeleteView` (`LoginRequiredMixin`, `generic.DeleteView`) – видаляє користувача із бази даних, використовує стандартну логіку видалення об'єкта.

Класи для оцінювання фільмів налічують:

– `RatingUpdateView` (`LoginRequiredMixin`, `generic.UpdateView`) – дає змогу користувачу змінити свою оцінку фільму, наслідує логіку оновлення, застосовану до оцінок;

– RatingDeleteView (LoginRequiredMixin, generic.DeleteView) – видаляє оцінку, залишену користувачем, використовує стандартне видалення з перевіркою прав доступу.

Для реалізації вебзастосунку важливими є й класи форм, які створюються для забезпечення введення та валідації даних користувачів, що надходять із HTML-форм під час взаємодії з інтерфейсом (рис. 2.4).

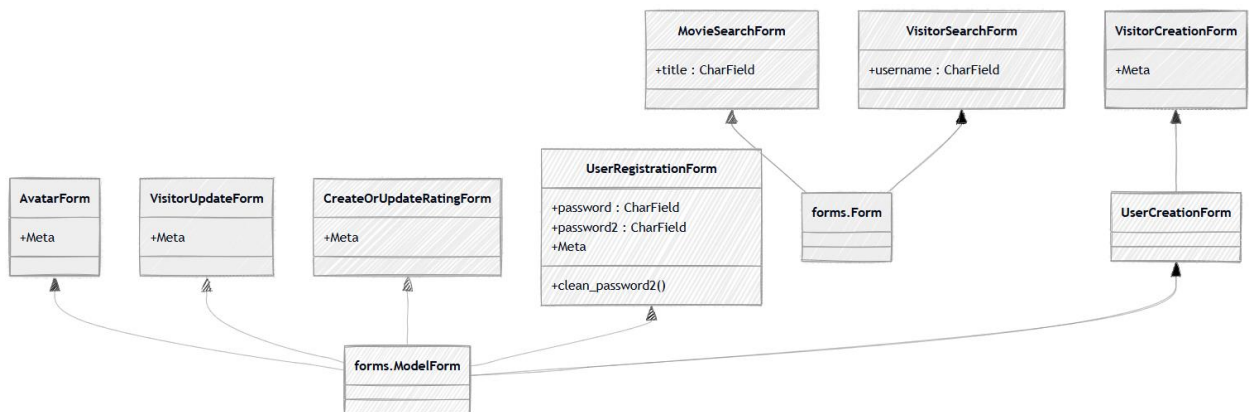


Рисунок 2.4 – UML-діаграма класів форм

У системі передбачено такі класи форм:

- AvatarForm (наслідує від forms.ModelForm) – модельна форма для зміни аватара користувача (на основі моделі Visitor);
- MovieSearchForm (наслідує від forms.Form) – звичайна форма для пошуку фільмів за назвою;
- VisitorSearchForm (наслідує від forms.Form) – звичайна форма для пошуку користувачів за іменем користувача;
- VisitorCreationForm (наслідує від UserCreationForm, forms.ModelForm) – розширена форма реєстрації нового користувача на базі UserCreationForm, з додатковими полями (електронна пошта, ім'я, прізвище, список бажаного);
- VisitorUpdateForm (наслідує від forms.ModelForm) – модельна форма для редагування профілю користувача;
- CreateOrUpdateRatingForm (наслідує від forms.ModelForm) – форма для створення або оновлення оцінки, включає поля значення оцінки та опис;

– `UserRegistrationForm` (наслідує від `forms.ModelForm`) – окрема форма реєстрації, яка забезпечує перевірку збігу паролів та створення облікового запису користувача.

2.3 Математичне та алгоритмічне забезпечення

Система передбачає кілька типів ролей, до яких відноситься користувач, який не був авторизований, авторизований користувач та адміністратор. Для кожного з них є перелік можливих дій у застосунку, зокрема, гостю (неавторизованому користувачу) дозволено:

- перегляд списку фільмів;
- перегляд деталей фільму;
- доступ до сторінки реєстрації/входу.

Зареєстрований користувач має доступ до наступних дій:

- створення особистого кабінету;
- зміни персональних даних (ім'я, прізвище, email, username);
- зміни аватара (вибір із набору, завантаженого в AWS S3);
- виставлення оцінки фільму (одна на користувача);
- додавання або видалення фільму зі списку бажаного (wishlist);
- перегляду власних оцінок;
- видалення або редагування своїх оцінок;
- фільтрації фільмів за назвою або жанром.

Останньою категорією є адміністратор, який може:

- керувати фільмами, жанрами, користувачами, оцінками через стандартну панель адміністратора;
- масово додавати або редагувати записи через інтерфейс `admin.site`.

У системі реалізовано низку типових сценаріїв взаємодії користувача з функціоналом платформи через вебінтерфейс. Один із них – перегляд каталогу фільмів, доступний для всіх користувачів (рис. 2.5).



Рисунок 2.5 – UML-діаграма варіантів використання системи для перегляду каталогу фільмів

Наступний сценарій – перегляд сторінки фільму, що включає огляд наявної інформації, оцінок та відгуків від інших користувачів та взаємодію з обраною стрічкою у разі авторизації (рис. 2.6).



Рисунок 2.6 – UML-діаграма варіантів використання системи для взаємодії зі сторінкою фільму

Авторизований користувач, переглядаючи обраний фільм, має можливість залишити власний відгук, який включає оцінку за 10-бальною шкалою і короткий опис, що відображено на діаграмі на рис. 2.7.

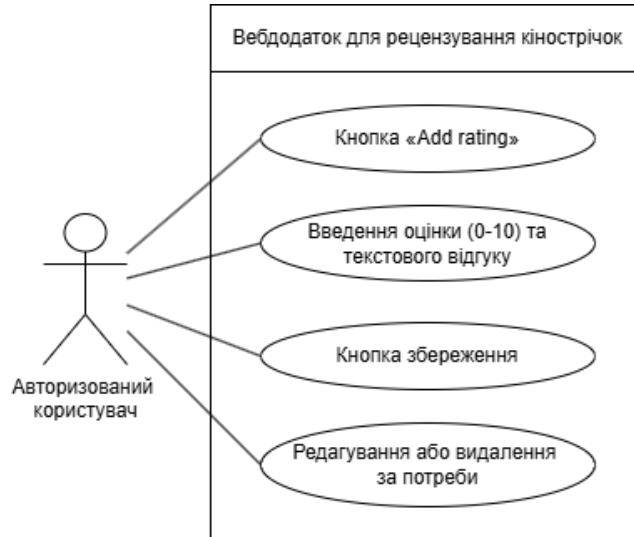


Рисунок 2.7 – UML-діаграма варіантів використання системи для залишення оцінки

Ще одним зі сценаріїв, доступних для користувача, є керування особистим профілем, що включає зміну залишених під час реєстрації даних, вибір аватара та ознайомлення зі списком фільмів, бажаних до перегляду (рис. 2.8).

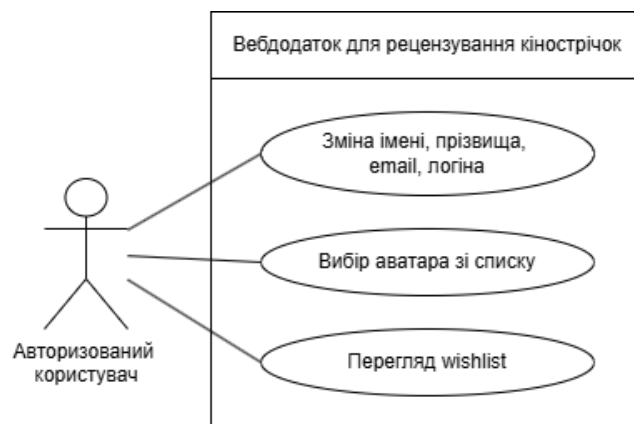


Рисунок 2.8 – UML-діаграма варіантів використання системи для керування профілем

Також важливо відобразити й процес створення облікового запису, для якого користувачу необхідно вводити особисті дані. Можливі варіанти показано на діаграмі на рис. 2.9.

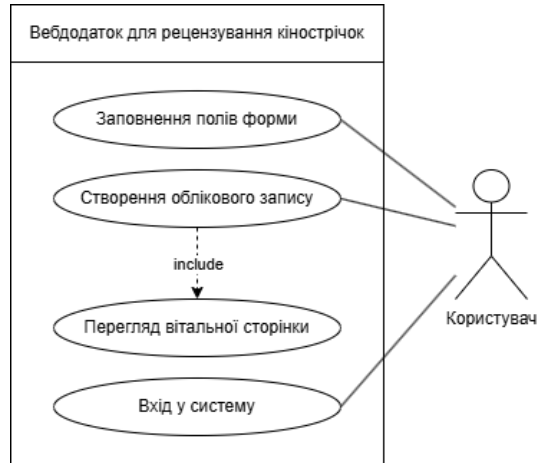


Рисунок 2.9 – UML-діаграма варіантів використання системи для

На рис. 2.10 зображено загальну схему взаємодії компонентів проекту.

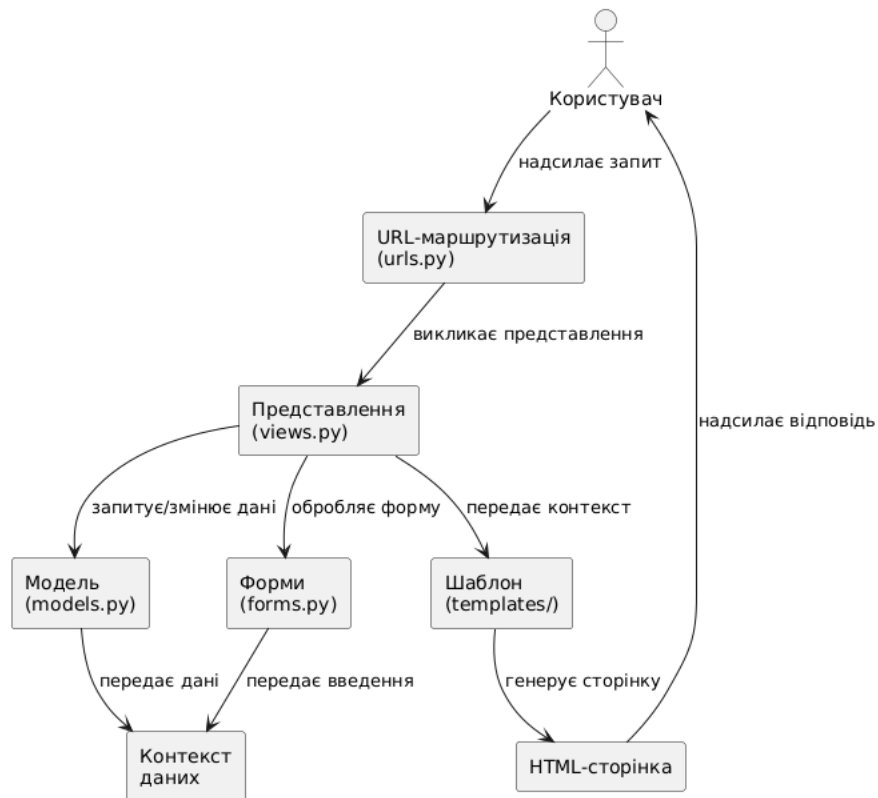


Рисунок 2.10 – Схема взаємодії компонентів системи

Запит користувача спочатку обробляється системою маршрутизації, що викликає відповідне представлення, яке звертається до моделей для отримання даних і/або до форм для обробки введення. Сформований контекст передається у відповідний шаблон, який генерує HTML-сторінку для користувача.

Послідовність взаємодії користувача з застосунком відображено на діаграмі рис. 2.11.

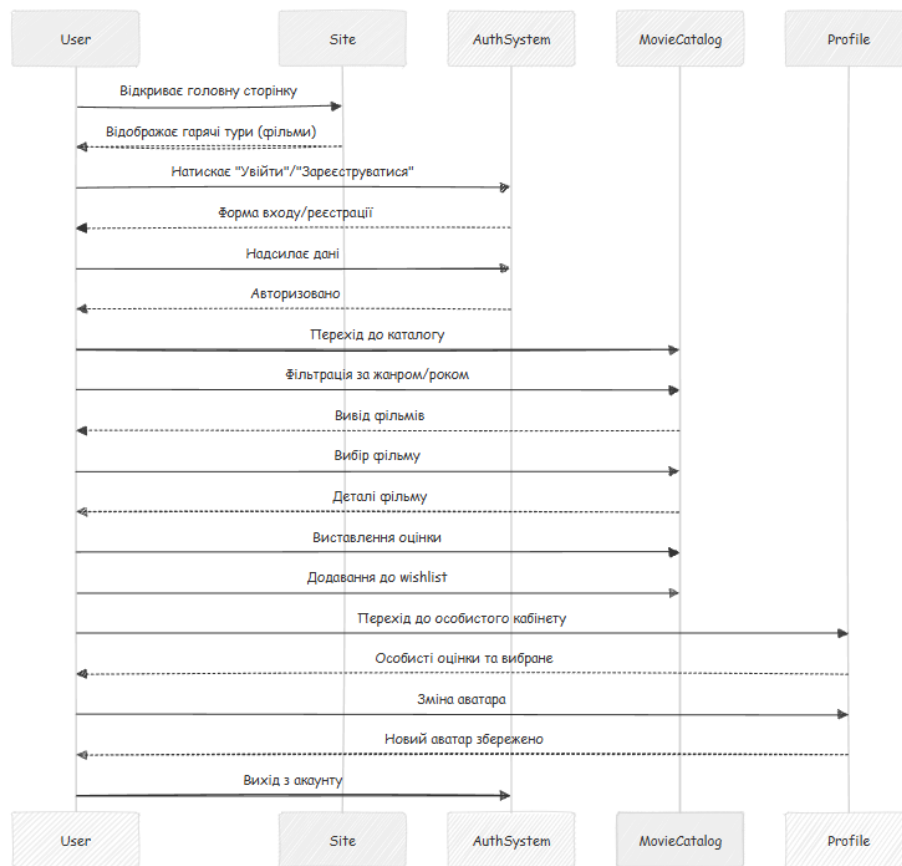


Рисунок 2.11 – Діаграма моделі поведінки користувача

Висновки до розділу

У розділі було проаналізовано предметну область, що включало виокремлення ключових функцій та понять системи, зокрема таких, як фільм, жанр, оцінка, вибране, користувач, адміністратор. Також було визначено вхідні і вихідні дані.

У ході проектування бази даних було розглянуто створення таблиць за ключовими сутностями системи, які включають фільми, жанри, користувачів, рейтинги, вподобання та списки бажаного до перегляду. За кожною сутністю в таблицях було відображено відповідні до атрибутів поля.

Побудова об'єктно-орієнтованої моделі дозволила визначити основні класи системи, зокрема класи моделей, представлень і форм, згідно з якими було побудовано відповідні UML-діаграми.

Алгоритмічне забезпечення включало розгляд типів ролей та відповідних доступів, а також відображення ключових сценаріїв дій користувачів у системі. Крім того, було визначено алгоритм взаємодії таких компонентів проєкту, як URL-маршрутизація, представлення, форми, моделі й шаблони, у процесі надсилання запиту.

Побудовані діаграми бази даних і класів, а також виокремлені алгоритми системи стануть основою для подальшої програмної реалізації вебдодатку для рецензування кінострічок.

РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Для реалізації проєкту було обрано мову Python [7] та фреймворк Django [8], які забезпечують високу продуктивність розробки, надійність та структурованість системи.

Python – сучасна мова програмування, що має простий синтаксис, багатий набір бібліотек і широке ком'юніті. Її вибір зумовлено тим, що вона підходить для реалізації вебзастосунків завдяки високій читабельності коду й швидкому розгортанню.

Python посідає перше місце в рейтингу TIOBE Index станом на червень 2025 року [9], що свідчить про її широке застосування у різних галузях програмування та актуальність для вирішення сучасних завдань (рис. 3.1).











Jun 2025	Jun 2024	Change	Programming Language	Ratings	Change
1	1		 Python	25.87%	+10.48%
2	2		 C++	10.68%	+0.65%
3	3		 C	9.47%	+0.24%
4	4		 Java	8.84%	+0.44%
5	5		 C#	4.69%	-1.96%
6	6		 JavaScript	3.21%	-0.11%
7	7		 Go	2.28%	+0.35%
8	9	▲	 Visual Basic	2.20%	+0.54%
9	11	▲	 Delphi/Object Pascal	2.15%	+0.62%
10	10		 Fortran	1.86%	+0.33%

Рисунок 3.1 – Рейтинг мов програмування за версією TIOBE (червень 2025 року)

Фреймворк Python Django пропонує повний набір інструментів і бібліотек для швидкого й ефективного створення вебдодатків [10, с. 124].

Django реалізує концепцію «DRY» (Don't Repeat Yourself) та MVC (Model–View–Controller) через власну архітектуру MTV (Model–Template–View). Його перевагами є:

- вбудована ORM для роботи з базами даних;
- підтримка шаблонів HTML через template engine;
- адміністративна панель «з коробки»;
- система авторизації та реєстрації;
- зручна маршрутизація URL;
- підтримка middleware, signals, forms;
- велика кількість готових бібліотек (crispy-forms, debug-toolbar, django-storages тощо).

Проект реалізовано з використанням Django 4.2.5 та Python 3.12.6, що є стабільними та активно підтримуються спільнотою, а також з застосуванням бази даних PostgreSQL [11].

Основним середовищем розробки є PyCharm [12] – потужна IDE для Python, його інтерфейс відображено на рис. 3.2.

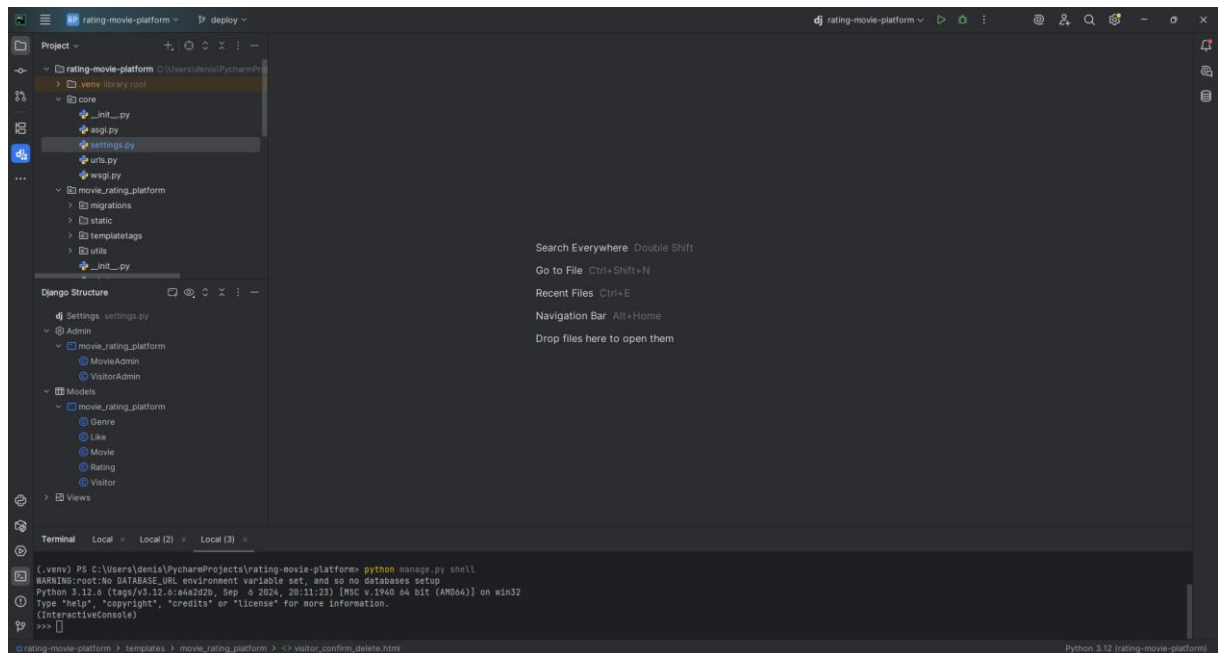


Рисунок 3.2 – Інтерфейс середовища розробки PyCharm з відкритим проектом

PyCharm забезпечує:

- автоматичне завершення коду та підсвічування синтаксису;
- інтеграцію з Git для керування версіями;
- зручну навігацію між файлами та класами;
- інтеграцію з Django через спеціальний плагін;
- підтримку запуску серверу розробки та управління БД.

Бібліотеки, що використовувалися в проєкті, включають наступні:

- `crispy-forms` [13] – для зручної побудови та стилізації HTML-форм;
- `debug-toolbar` [14] – для налагодження запитів, відстеження ORM-операцій, перегляду змінних запиту;
- `django-storages` [15] + `boto3` [16] – для інтеграції з AWS S3 (медіафайли);
- `dj-database-url` [17] – для зчитування параметрів підключення до PostgreSQL через змінні середовища.

3.2 Вимоги до технічного та програмного забезпечення

Системні вимоги для клієнта (користувача сайту):

- ОС: будь-яка сучасна (Windows 7+, macOS 10+, Linux);
- браузер: Google Chrome, Mozilla Firefox, Microsoft Edge або Safari (останні версії);
- роздільна здатність екрану: мін. 1024×768;
- підключення до Інтернету: стабільне, не менше 1 Мбіт/с.

Системні вимоги для розробника:

- ОС: Windows 10+, macOS 11+, Ubuntu 20.04+;
- Python: версія 3.10+;
- браузер: будь-який серед сучасних для перевірки працездатності;
- встановлені компоненти: Python, pip, Git, PostgreSQL (або доступ до віддаленої бази на NeonDB), Docker (опціонально, для тестового деплою).

Вимоги до серверу (виробниче середовище):

- хостинг: Render.com [18];
- база даних: PostgreSQL (через Neon.tech [19]);
- сховище файлів: Amazon S3 (через boto3);
- середовище виконання: Python 3.10+, Django 4.2.5, Gunicorn / WhiteNoise (для обробки запитів);
- підтримка змінних середовища (.env або панель керування Render).

3.3 Опис програмної реалізації

У процесі програмної реалізації застосовувалися різні інструменти для ефективної організації роботи, спрощення розробки та налагодження, включно зі вбудованою адміністративною панеллю Django (/admin/), відображеною на рис. 3.3, її було застосовано для:

- керування користувачами (авторизація, видалення);
- створення, редагування та видалення об'єктів моделей (фільми, рецензії, профілі);
- попереднього наповнення бази даних без використання SQL;
- тестування моделей та перевірки взаємозв'язків.

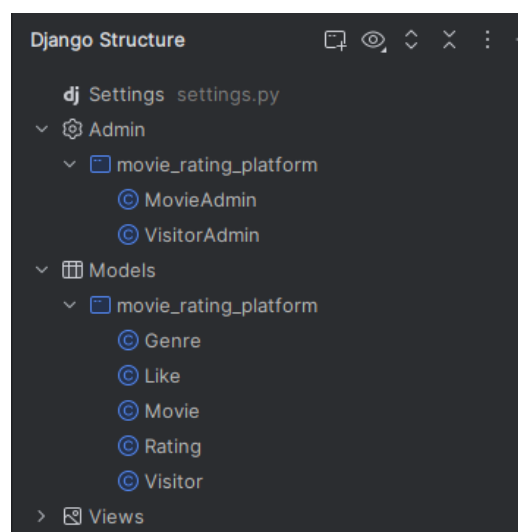


Рисунок 3.3 – Адміністративна панель Django з реєстрацією моделей

Важливою є й наявність усіх необхідних усіх необхідних бібліотек і локальних застосунків у списку `INSTALLED_APPS`, адже саме їх підключення визначає доступність функціоналу в усьому проєкті (рис. 3.4).

```
INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "movie_rating_platform",
    "debug_toolbar",
    "crispy_forms",
    "crispy_bootstrap4",
    "storages"
]
```

Рисунок 3.4 – Вміст `INSTALLED_APPS` у файлі `settings.py`

3.3.1 Середовище виконання

Вебзастосунок `RateFlix` реалізований з використанням фреймворку `Django` і вимагає налаштованого середовища для правильного функціонування, як у розробницькому, так і у виробничому режимі.

Для гарантування безпеки та конфігурації застосунку використовуються змінні середовища:

- `DJANGO_SECRET_KEY` – секретний ключ застосунку;
- `DEBUG` – логічний прапорець для режиму налагодження (у продакшені має бути `False`);
- `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_STORAGE_BUCKET_NAME` – доступ до сховища медіафайлів `Amazon S3`;
- `DATABASE_URL` – рядок підключення до `PostgreSQL` через `NeonDB`.

На платформі `Render.com` середовище виконує такі функції:

- автоматичне розгортання з `GitHub`;
- підтримка команд `build` та `start` (наприклад, `python manage.py migrate`, `gunicorn core.wsgi`);

- налаштування змінних середовища через UI (рис. 3.5);
- автоматичне SSL.

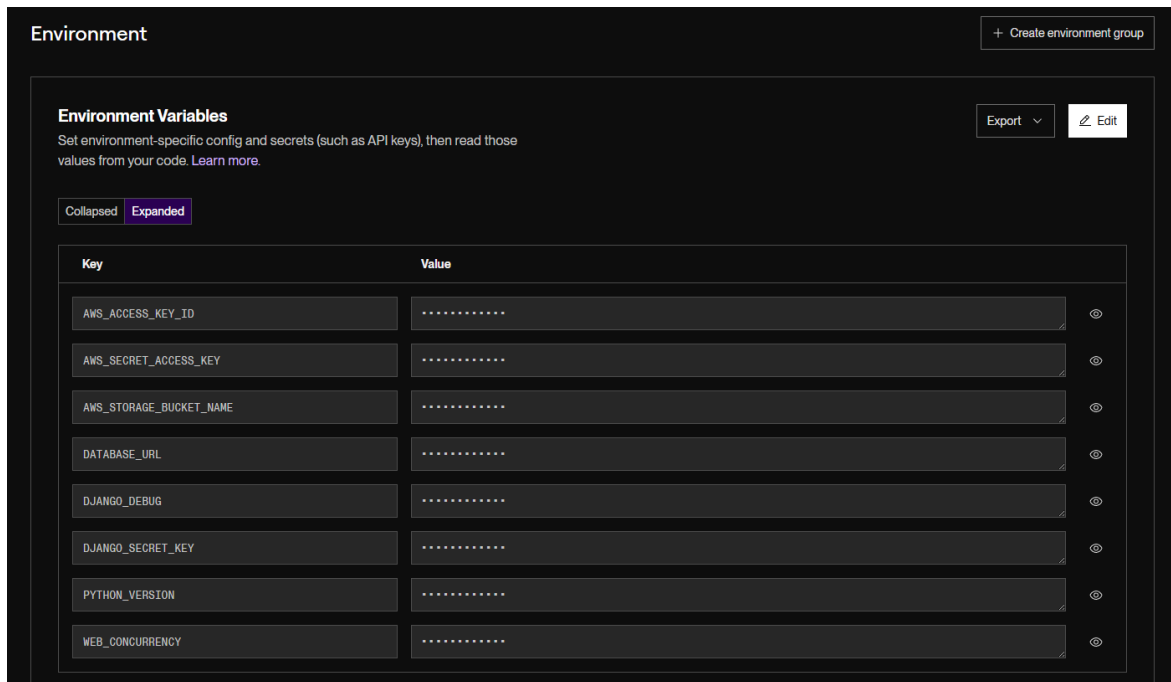


Рисунок 3.5 – Список змінних середовища в панелі керування Render.com

Підключення до віддаленої PostgreSQL бази даних на Neon.tech відбувається з використанням SSL, збереженням параметрів в DATABASE_URL:

```
perl
```

```
postgres://username:password@host:port/dbname?sslmode=require
```

У проєкті використано бібліотеку dj-database-url, яка зчитує параметри з середовища. Статичні файли обслуговуються через WhiteNoise, а медіафайли зберігаються на Amazon S3 через бібліотеку boto3 та django-storages.

3.3.2 Безпека доступу

Вебзастосунок RateFlix реалізує базові засоби захисту даних користувачів шляхом використання вбудованої системи аутентифікації Django, а також розмежування доступу за допомогою прав користувачів.

Аутентифікація реалізована через Django Auth (django.contrib.auth), вона підтримує такі операції:

- реєстрація – створення облікового запису;
- вхід та вихід через стандартні шаблони login/logout;
- редагування профілю.

Після входу користувач перенаправляється на головну сторінку (рис. 3.6).

```
LOGIN_REDIRECT_URL = "/"
```

Рисунок 3.6 – Реалізація переходу на головну сторінку після входу в додаток

Обмеження прав доступу реалізовується через шаблони залежно від того, чи увійшов користувач до системи, або ж це є адміністратор (рис. 3.7).

```
{% if user.is_authenticated %}
<li class="nav-item"><a class="nav-link nav-link-a" href="{{ user.get_absolute_url }}">Profile</a></li>
<li class="nav-item"><a class="nav-link bg-success rounded active nav-link-a" href="{% url 'logout' %}?next={{request.path}}">Logout</a></li>
{% else %}
<li class="nav-item"><a class="nav-link bg-success rounded active nav-link-a" href="{% url 'login' %}?next={{request.path}}">Login</a></li>
{% endif %}
```

Рисунок 3.7 – Обмеження доступу в шаблонах

Користувач, який пройшов аутентифікацію, має такі права доступу:

- доступ до свого профілю;
- можливість ставити оцінки;
- додавання фільмів у список бажаного.

Для користувача, який не увійшов до особистого профілю, передбачено:

- доступ лише до перегляду фільмів;
- перенаправлення до сторінки входу у разі спроби здійснення дій, що потребують аутентифікації.

Адміністратор (is_staff) має доступ до панелі Django Admin та може редагувати або видаляти всі фільми, відгуки і профілі.

Усі форми захищені CSRF-токенами (рис. 3.8).

```

<div class="centered">
  <h1>Create an account</h1>
  <p>Please, sign up using the following form:</p>
  <form action="." method="post">
    {{ user_form|crispy }}
    {% csrf_token %}
    <p><input class="btn btn-secondary" type="submit" value="Create my account"></p>
  </form>
</div>

```

Рисунок 3.8 – Реалізація форми з використанням csrf_token

Для валідації форм, приклад якої продемонстровано на рис. 3.9, використано Django Forms + crispy-forms.

Please login to see this page.

- Please enter a correct username and password. Note that both fields may be case-sensitive.

Username*

Password*

Рисунок 3.9 – Повідомлення про помилку при неправильному вході

Захист медіафайлів забезпечується через зберігання аватарів користувачів у хмарному сховищі Amazon S3, при цьому URL-адреси файлів не містять токенів автентифікації завдяки параметру `AWS_QUERYSTRING_AUTH = False`, що підвищує безпеку доступу.

Щодо захисту середовища, всі секретні ключі зберігаються у змінних середовища на платформі Render.com, при цьому в продакшн-режимі встановлено `DEBUG = False`, а список дозволених хостів (`ALLOWED_HOSTS`) обмежує запити ззовні.

3.3.3 Реалізація бази даних та взаємодії між компонентами проєкту

Архітектура проєкту побудована за принципами MVC (Model–View–Controller), адаптованими під Django Framework. Основна логіка розподілена між такими складовими:

- Models (`models.py`) – визначають структуру даних (фільми, жанри, користувачі, рейтинги, лайки);

- Forms (forms.py) – реалізують логіку створення та обробки форм (реєстрація, пошук, зміна аватара, оцінка);
- Views (views.py) – реалізують контролери для обробки запитів, взаємодіють із моделями та шаблонами;
- Templates (templates/) – HTML-шаблони для відображення інтерфейсу (головна, деталізація фільмів, список користувачів тощо);
- Urls (urls.py) – маршрутизація для кожного типу запиту;
- Admin (admin.py) – підключення моделей до адмін-панелі та кастомізація інтерфейсу.

Допоміжні компоненти включають:

- utils/ – утиліти (наприклад custom_title для форматування назв фільмів);
- static/ – стилі та ресурси;
- media/ – зображення (аватари, постери фільмів), зберігаються в AWS S3.

Проект розділено на додаток movie_rating_platform з централізованим керуванням логікою. Його структуру відображено на рис. 3.10.

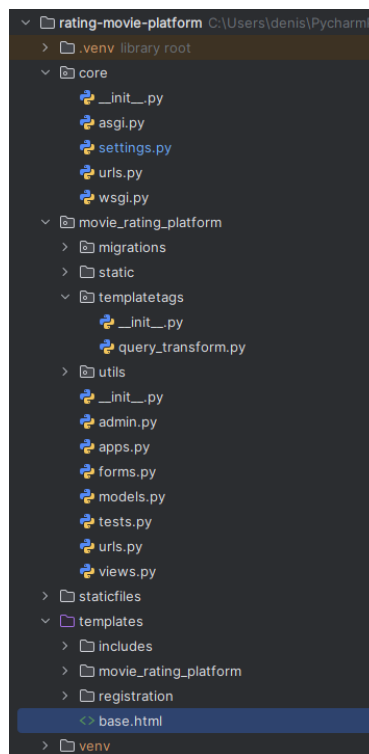


Рисунок 3.10 – Структура директорій проекту

Також реалізовано RESTful-підхід, за яким кожна сторінка (список, деталі, створення, редагування чи видалення) має свій URL та функцію.

Для зберігання інформації про фільми, користувачів та їхні оцінки створено набір моделей, які повністю формують логіку предметної області.

Модель Movie включає назву, опис, рік випуску, зображення (постер), зв'язок із жанрами та оцінками (рис. 3.11).

```
class Movie(models.Model):
    title = models.CharField(max_length=255, unique=True)
    description = models.TextField(null=True, blank=True)
    release_year = models.PositiveIntegerField(
        validators=[
            MinValueValidator(1895),
            MaxValueValidator(datetime.datetime.now().year),
        ]
    )
    poster = models.ImageField(upload_to="posters/", null=True, blank=True)
    genres = models.ManyToManyField(Genre, related_name="movies")

    def save(self, *args, **kwargs):
        self.title = custom_title(self.title)
        super().save(*args, **kwargs)

    def __str__(self):
        return f"{self.title} ({self.release_year})"
```

Рисунок 3.11 – Реалізація класу моделі Movie

Модель Genre призначена для зберігання назви жанру (рис. 3.12).

```
class Genre(models.Model):
    name = models.CharField(
        max_length=63,
        unique=True,
        error_messages={"unique": "This genre already exists"},
    )

    class Meta:
        ordering = ("name",)

    def save(self, *args, **kwargs):
        self.name = self.name.lower()
        super().save(*args, **kwargs)

    def __str__(self):
        return self.name
```

Рисунок 3.12 – Реалізація класу моделі Genre

Модель Rating реалізує зв'язок між користувачем (Visitor) і фільмом, містить числову оцінку та коментар (рис. 3.13).

```
class Rating(models.Model): 9 usages  CaCtUs1k
    value = models.PositiveIntegerField(
        validators=[MinValueValidator(0), MaxValueValidator(10)]
    )
    description = models.TextField(null=True, blank=True)
    sender = models.ForeignKey(
        Visitor, on_delete=models.DO_NOTHING, related_name="ratings"
    )
    create_time = models.DateTimeField(auto_now_add=True)
    movie = models.ForeignKey(
        Movie, on_delete=models.CASCADE, related_name="ratings"
    )

    def __str__(self): CaCtUs1k
        return self.description
```

Рисунок 3.13 – Реалізація класу моделі Rating

Модель Visitor – розширена кастомна модель користувача з аватаром (зображенням) та полем wishlist (рис. 3.14).

```
class Visitor(AbstractUser): 20 usages  CaCtUs1k
    wishlist = models.ManyToManyField(
        Movie,
        related_name="users",
    )
    avatar = models.ImageField(upload_to='avatars/', default='avatars/black.jpg')

    def get_absolute_url(self): CaCtUs1k
        return reverse(viewname="movie_rating:visitor-detail", kwargs={"pk": self.pk})

    def __str__(self): CaCtUs1k
        return f"{self.username}"
```

Рисунок 3.14 – Реалізація класу моделі Visitor

Модель Like створена для відміток про вподобання користувачем певного фільму (рис. 3.15).

```
class Like(models.Model): CaCtUs1k
    sender = models.ForeignKey(
        Visitor, on_delete=models.DO_NOTHING, related_name="likes"
    )
    target = models.ForeignKey(
        Movie, on_delete=models.DO_NOTHING, related_name="likes"
    )
```

Рисунок 3.15 – Реалізація класу моделі Like

Особливостями реалізації передбачено використання `related_name` для зручного доступу до зв'язаних даних. Для аватарів і постерів використовується поле `ImageField`, що забезпечує можливість завантаження файлів, а для `wishlist` застосовується `ManyToManyField`, тобто кожен користувач може зберігати кілька улюблених фільмів.

Всі моделі зареєстровано в адміністративній панелі для ручного керування записами, а також налаштовано виведення ключових полів (`list_display`) та фільтрацію (`list_filter`, `search_fields`).

Для збереження інформації про фільми, жанри, оцінки, вподобання та користувачів у системі реалізовано реляційну базу даних PostgreSQL. Структура бази даних була створена за допомогою ORM Django, яка автоматично генерує таблиці на основі моделей Python-коду.

Схема бази даних містить такі основні таблиці:

- `movie_rating_platform_visitor` – інформація про користувачів (логін, пароль, email, аватар, права доступу тощо);
- `movie_rating_platform_movie` – фільми з полями назви, опису, року виходу та постера;
- `movie_rating_platform_genre` – список жанрів;
- `movie_rating_platform_rating` – оцінки фільмів, що містять значення, опис, дату та посилання на автора і фільм;
- `movie_rating_platform_like` – позначки «вподобано» між користувачами та фільмами;
- `movie_rating_platform_movie_genres` – зв'язок багато-до-багатьох між фільмами та жанрами;
- `movie_rating_platform_visitor_wishlist` – обране користувача (зв'язок багато-до-багатьох між користувачами та фільмами);
- `movie_rating_platform_visitor_groups` / `visitor_user_permissions` – таблиці авторизації Django для роботи з правами доступу.

Схема бази даних демонструє всі сутності, їх атрибути та зв'язки між ними (рис. 3.16).

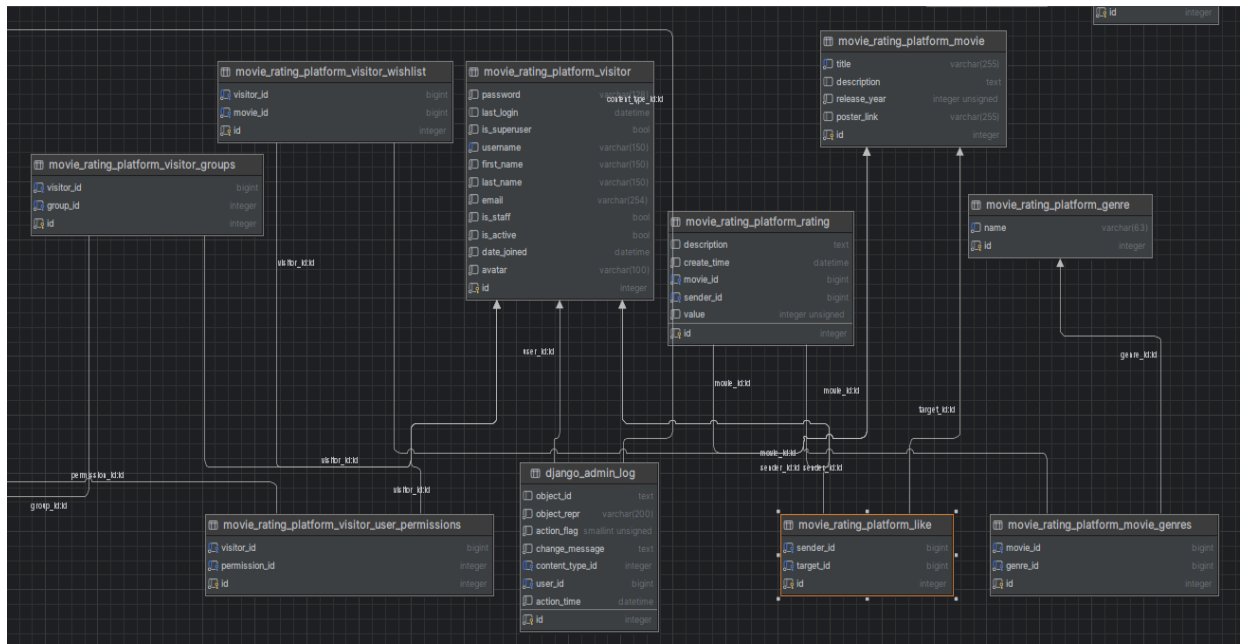


Рисунок 3.16 – Структура бази даних

Після створення структури даних основну логіку взаємодії користувача з системою реалізовано через представлення (views) та шаблони (templates). Це забезпечує повноцінну роботу із фільмами, профілями та рейтингами.

Усі представлення розроблено з використанням класів-нащадків Django CBV (Class-Based Views) – це дозволяє зручно організувати логіку CRUD-операцій. Представлення дають можливість:

- переглядати списки фільмів і користувачів завдяки ListView із можливістю фільтрації за жанром та пошуку за назвою;
- детальних переглядів через DetailView для фільмів та користувачів з виведенням оцінок, описів, кнопок взаємодії;
- створення й редагування через CreateView, UpdateView, DeleteView для фільмів, користувачів та оцінок;
- додаткових дій завдяки кастомним views, що включає, наприклад, додавання фільму до wishlist або зміну аватару.

Для кожного представлення створено окремий HTML-шаблон, який наслідує базовий шаблон base.html, що містить хедер, футер та сітку Bootstrap.

Шаблони включають:

- movie_list.html – список фільмів з фільтрами та пагінацією;

- movie_detail.html – опис фільму, рейтинг, коментарі, кнопки дій;
- visitor_detail.html – профіль користувача, його аватар і wishlist.

Усі форми стилізовані за допомогою crispy-forms.

Функціональність в шаблонах включає фільтрацію фільмів за жанром через GET-запити, відображення кількості зірок згідно з оцінкою (value) через цикл, виведення списку бажаних фільмів кожного користувача та відображення кнопок редагування або видалення лише для власника або адміністратора.

Для взаємодії з базою даних у проєкті використовується ORM Django, яка дозволяє працювати з даними у вигляді Python-об'єктів без написання SQL-запитів.

До прикладу, робота з моделлю Movie з застосуванням запитів та взаємодії з ORM може включати отримання всіх фільмів через Movie.objects.all(). Фільтрація фільмів за роком випуску відбувається завдяки Movie.objects.filter(release_year=2024), а пошук за назвою (часткове співпадіння) – Movie.objects.filter(title__icontains="matrix").

В представленнях (views) ORM-запити використовуються для реалізації логіки роботи додатку: фільтрації фільмів, додавання до обраного, виставлення оцінок тощо (рис. 3.17).

```
@login_required 2 usages CaCtUs1k
def movie_detail_view(request, pk):
    movie = get_object_or_404(Movie, pk=pk)
    max_value = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    movie.ratings.select_related("sender")
    user = request.user
    user_comment = Rating.objects.filter(sender=user, movie=movie).exists()
    context = {"movie": movie, "max_value": max_value, "user_comment": user_comment}
    return render(request, template_name="movie_rating_platform/movie_detail.html", context=context)
```

Рисунок 3.17 – Приклад ORM-запиту в представленні

У файлі urls.py реалізовано маршрутизацію для доступу до представлень (views) за допомогою іменованих шляхів. Для зручності також використовується простір імен namespace="movie_rating", що дозволяє

уникнути конфліктів у випадку підключення декількох додатків. Приклад налаштування наведено на рис. 3.18.

```
urlpatterns = [
    path(route: "register/", register, name="register"),
    path(route: "/", index, name="index"),
    path(route: "movies/", MovieListView.as_view(), name="movie-list"),
    path(route: "movies/create/", MovieCreateView.as_view(), name="movie-create"),
    path(route: "movies/<int:pk>/", movie_detail_view, name="movie-detail"),
```

Рисунок 3.18 – Фрагмент файлу urls.py із прикладами маршрутизації

3.3.4 Деплой сайту та підключення AWS для збереження медіа

Після завершення локальної розробки та тестування вебзастосунків було розгорнуто у продакшн-середовищі з використанням хостингу Render.com (рис. 3.19), що дозволяє швидко та зручно розгорнути Django-додатки.

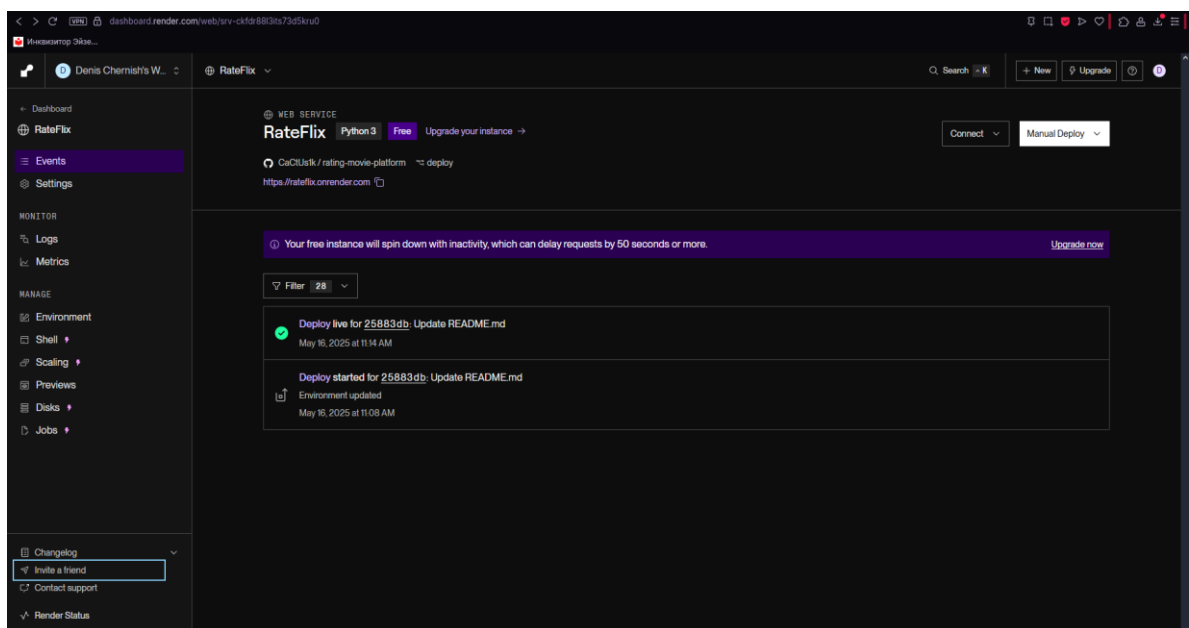


Рисунок 3.19 – Інтерфейс Render.com

Для деплою застосунку були виконано такі кроки:

- створено новий Web Service у Render.com з прив'язкою до GitHub-репозиторію;
- задано середовище виконання Python 3.11;
- задано Start Command: gunicorn core.wsgi:application;

– створено `.build.sh` файл, що містить інструкції для встановлення залежностей;

– додано змінні середовища: `DJANGO_SECRET_KEY`, `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_STORAGE_BUCKET_NAME`.

Render.com надає можливість підключити зовнішню базу даних. У проєкті використано NeonDB – хмарне рішення на базі PostgreSQL (рис. 3.20).

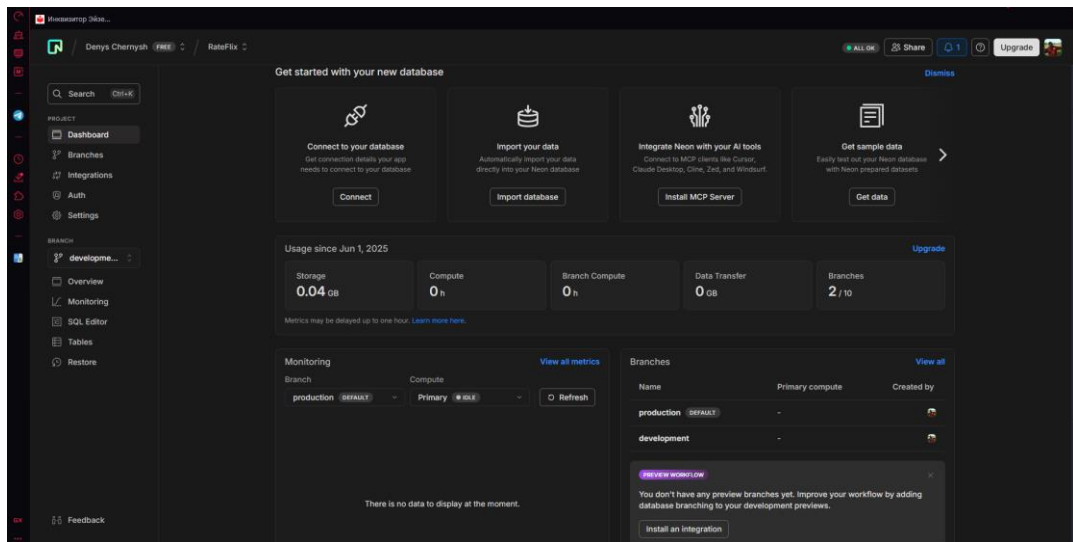


Рисунок 3.20 – Інтерфейс NeonDB

Було вказано параметри підключення в `settings.py` через `django_database_url`, а також активовано SSL-з'єднання (`sslmode=require`) для безпеки.

Ролі в базі даних відображено на рис. 3.21.

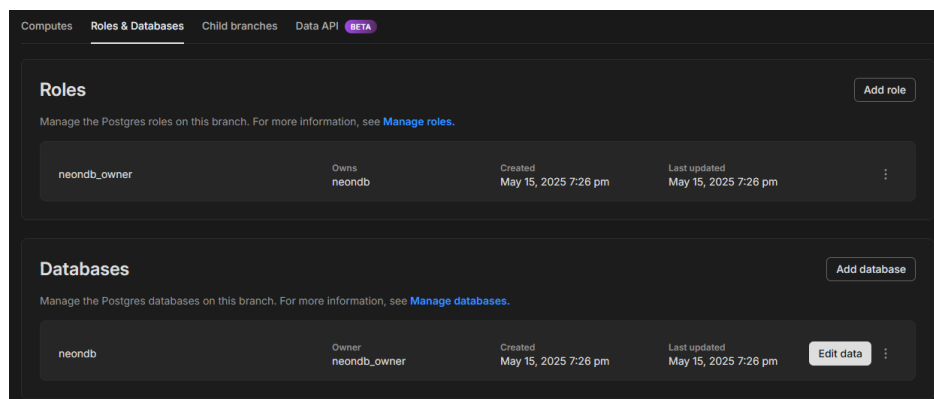


Рисунок 3.21 – Ролі NeonDB

Таблиці бази даних в NeonDB показано на рис. 3.22.

ID	name	id_movie_rating_platform_movie_genres_genre_id
1	Drama	id_movie_rating_platform_movie_genres_genre_id
2	Crime	id_movie_rating_platform_movie_genres_genre_id
3	Action	id_movie_rating_platform_movie_genres_genre_id
4	Adventure	id_movie_rating_platform_movie_genres_genre_id
5	Fantasy	id_movie_rating_platform_movie_genres_genre_id
6	Biography	id_movie_rating_platform_movie_genres_genre_id
7	History	id_movie_rating_platform_movie_genres_genre_id
8	Western	id_movie_rating_platform_movie_genres_genre_id
9	Romance	id_movie_rating_platform_movie_genres_genre_id
10	Thriller	id_movie_rating_platform_movie_genres_genre_id
11	Sci-Fi	id_movie_rating_platform_movie_genres_genre_id
12	Family	id_movie_rating_platform_movie_genres_genre_id
13	Horror	id_movie_rating_platform_movie_genres_genre_id
14	War	id_movie_rating_platform_movie_genres_genre_id
15	Comedy	id_movie_rating_platform_movie_genres_genre_id
16	Animation	id_movie_rating_platform_movie_genres_genre_id
17	Mystery	id_movie_rating_platform_movie_genres_genre_id
18	Music	id_movie_rating_platform_movie_genres_genre_id

Рисунок 3.22 – Відображення таблиць бази даних

Для організації зберігання користувацьких медіафайлів, таких як аватари, було використано хмарне сховище Amazon S3, яке є частиною інфраструктури AWS (рис. 3.23).

Name	Type	Last modified	Size	Storage class
12angry_men.jpg	jpg	May 17, 2025, 11:29:41 (UTC+03:00)	130.8 KB	Standard
7samurai.jpg	jpg	May 17, 2025, 11:29:40 (UTC+03:00)	143.7 KB	Standard
Alien.jpg	jpg	May 17, 2025, 11:29:41 (UTC+03:00)	72.5 KB	Standard
American.jpg	jpg	May 17, 2025, 11:29:41 (UTC+03:00)	121.4 KB	Standard
Apocalypse.jpg	jpg	May 17, 2025, 12:12:41 (UTC+03:00)	85.4 KB	Standard
avatars/	Folder	-	-	-
back_future.jpg	jpg	May 17, 2025, 11:29:41 (UTC+03:00)	148.0 KB	Standard
Casablanca.jpg	jpg	May 17, 2025, 11:29:42 (UTC+03:00)	101.5 KB	Standard
city_of_god.jpg	jpg	May 17, 2025, 11:29:42 (UTC+03:00)	181.0 KB	Standard
CityLights.jpg	jpg	May 17, 2025, 11:29:42 (UTC+03:00)	206.4 KB	Standard
cuckoos.jpg	jpg	May 17, 2025, 11:29:42 (UTC+03:00)	127.3 KB	Standard
darknight.jpg	jpg	May 17, 2025, 11:29:42 (UTC+03:00)	150.0 KB	Standard
Departed.jpg	jpg	May 17, 2025, 11:29:43 (UTC+03:00)	87.1 KB	Standard
Django.jpg	jpg	May 17, 2025, 11:29:43 (UTC+03:00)	156.0 KB	Standard
Dune2.jpg	jpg	May 17, 2025, 12:12:41 (UTC+03:00)	128.6 KB	Standard
fight_club.jpg	jpg	May 17, 2025, 11:29:43 (UTC+03:00)	152.9 KB	Standard
Fireflies.jpg	jpg	May 17, 2025, 11:29:43 (UTC+03:00)	104.4 KB	Standard

Рисунок 3.23 – Інтерфейс AWS

Назву бакета та його базові характеристики наведено на рис. 3.24.

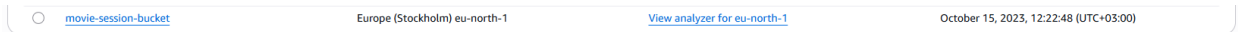


Рисунок 3.24 – Назва бакета та базові характеристики

Для підключення Django-проєкту до Amazon S3 було використано бібліотеку `boto3` та бекенд `S3Boto3Storage`, що входить до складу пакета `django-storages`. Відповідно, `storages` було додано до списку `INSTALLED_APPS`. Змінні доступу до AWS збережено у файлі `settings.py` (рис. 3.25).

```
DEFAULT_FILE_STORAGE = "storages.backends.s3boto3.S3Boto3Storage"
AWS_QUERYSTRING_AUTH = False
```

Рисунок 3.25 – Налаштування змінних середовища для інтеграції AWS S3

Перед розгортанням застосунку було виконано базові налаштування для продакшн-середовища. Зокрема, підключено `WhiteNoise` для обслуговування статичних файлів, встановлено `DEBUG = False` та додано параметр `ALLOWED_HOSTS = ['rateflix.onrender.com']`.

Процес деплою на платформу `Render.com` відбувається автоматично після кожного коміту в `GitHub` і включає такі етапи:

- надсилання змін у репозиторій `GitHub`;
- автоматичне розгортання в `Render`;
- міграції бази даних, які виконуються автоматично або можуть запускатися вручну через консоль;
- перевірка логів роботи сервера, доступна через вебінтерфейс `Render` (рис. 3.26).

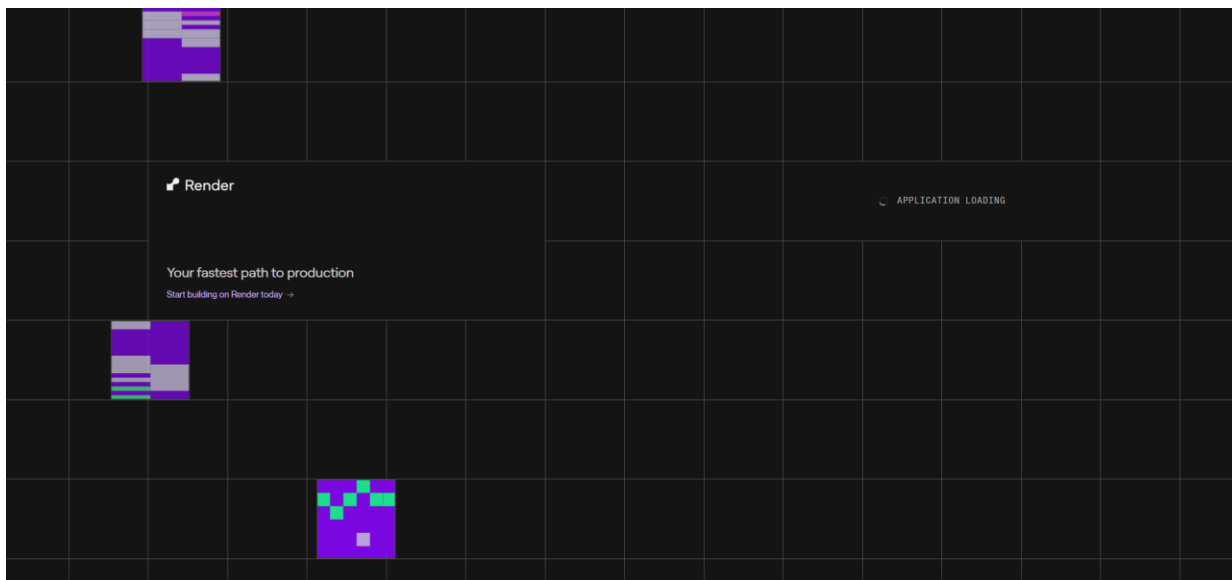


Рисунок 3.26 – Екран завантаження під час розгортання проекту

Після запуску застосунку у консолі можна переглядати логи виконання, помилки, інформаційні повідомлення та інші події, що полегшує діагностику і супровід системи (рис. 3.27).

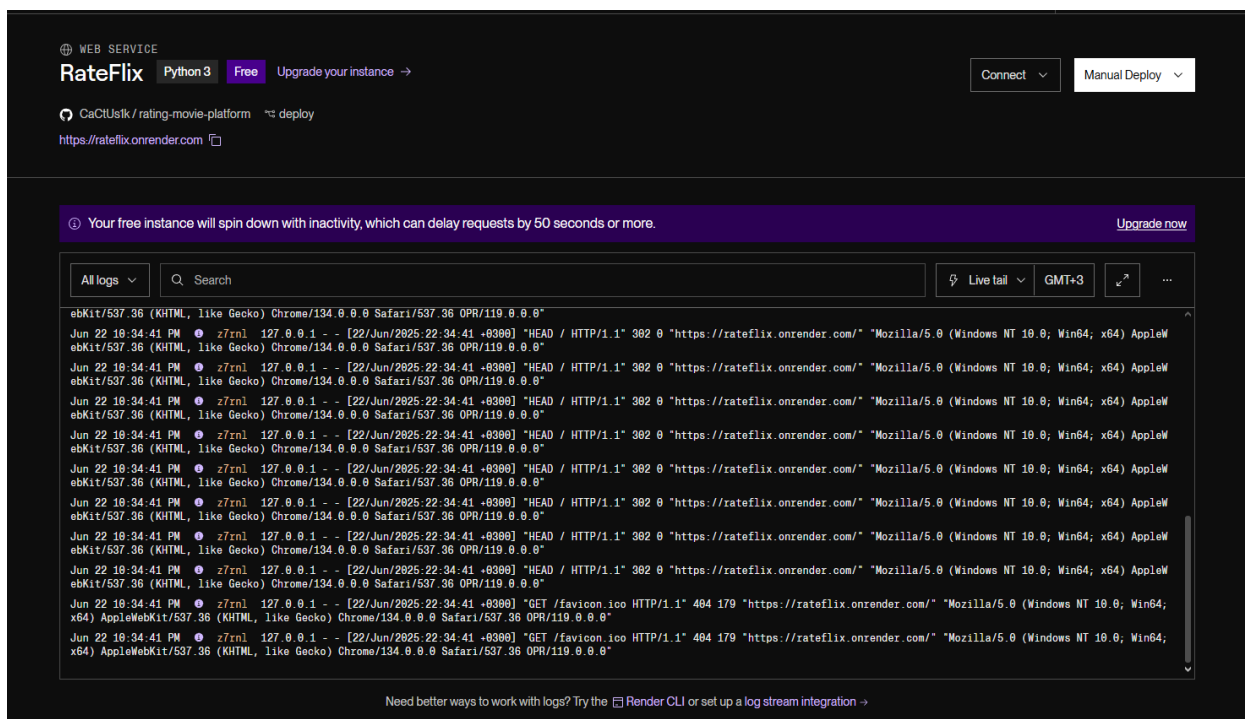


Рисунок 3.27 – Консоль логів після запуску застосунку

3.4 Керівництво користувача

Для входу до вебдодатку для рецензування кінострічок необхідно ввести логін та пароль (рис. 3.28).



Рисунок 3.28 – Сторінка входу

У разі відсутності акаунту користувач може його створити з обов'язковим вказанням логіну й паролю, за бажанням можна ввести ім'я та електронну адресу (рис. 3.29).

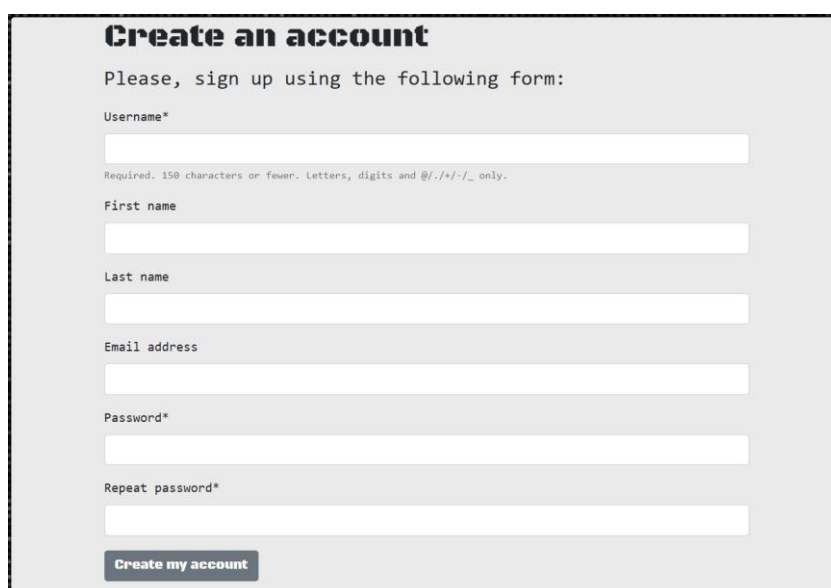


Рисунок 3.29 – Форма реєстрації

У разі успішного створення акаунту користувачу надається можливість увійти до системи (рис. 3.30).

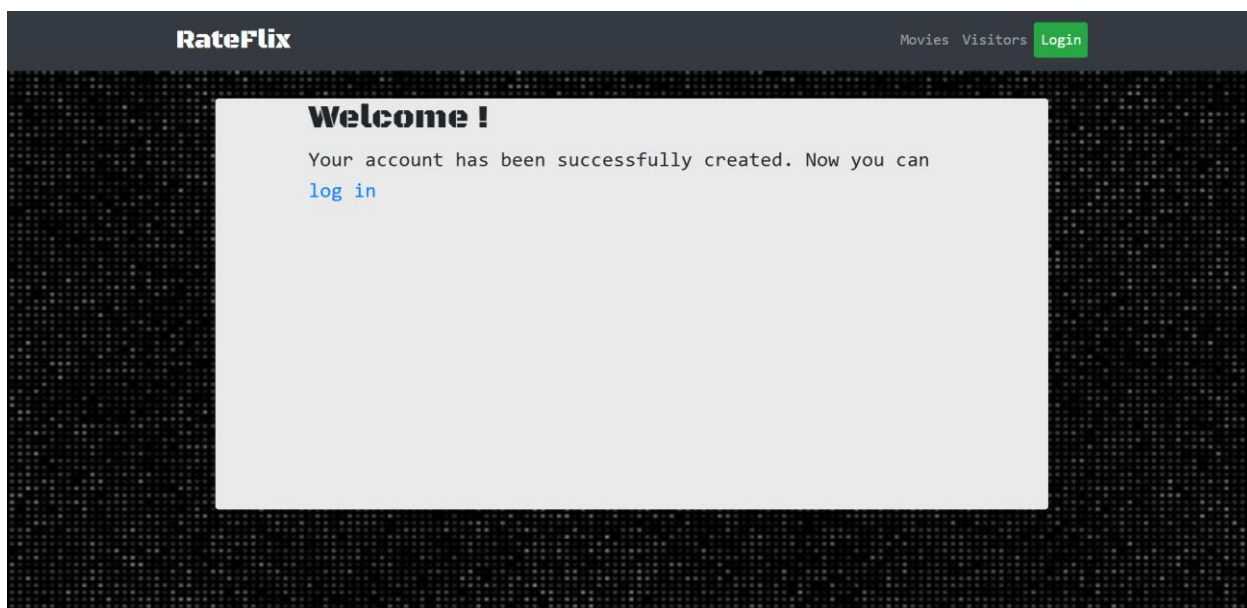


Рисунок 3.30 – Повідомлення про успішне створення акаунту

Головна сторінка відображає коротку інформацію про призначення й особливості вебдодатку, а також кількість користувачів, фільмів і жанрів (рис. 3.31).

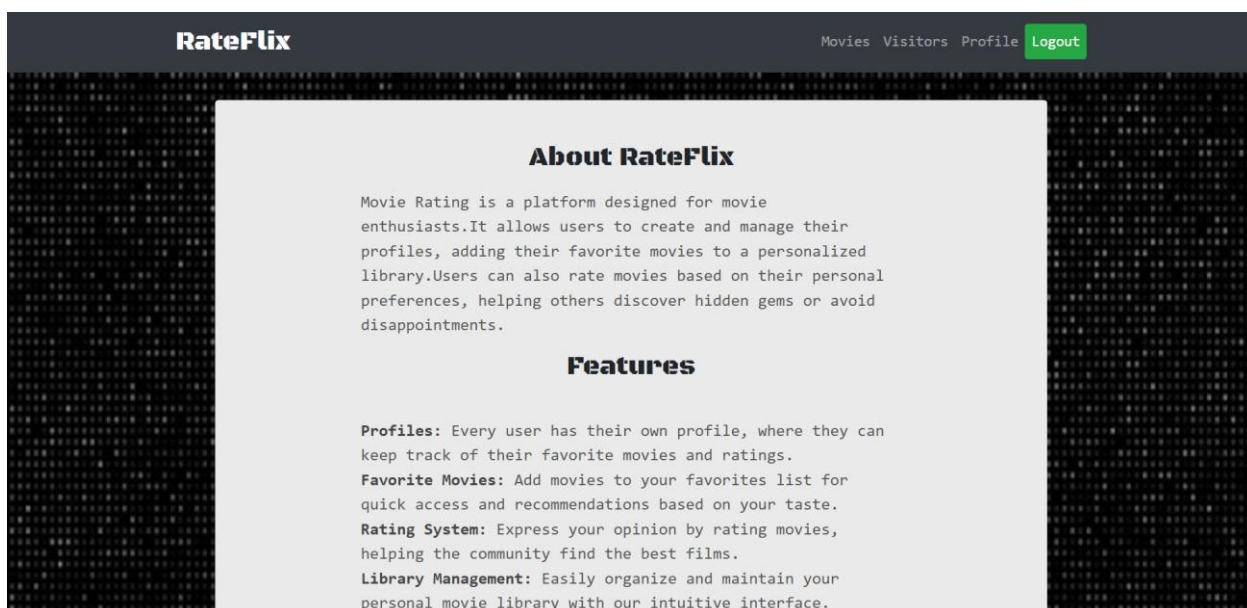


Рисунок 3.31 – Головна сторінка

Особистий профіль відображає дані, які користувач вводив під час реєстрації (за винятком паролю), а також надає можливість додати чи змінити аватар, видалити акаунт, відредагувати дані й переглянути список фільмів, доданих до вішліста (рис. 3.32).

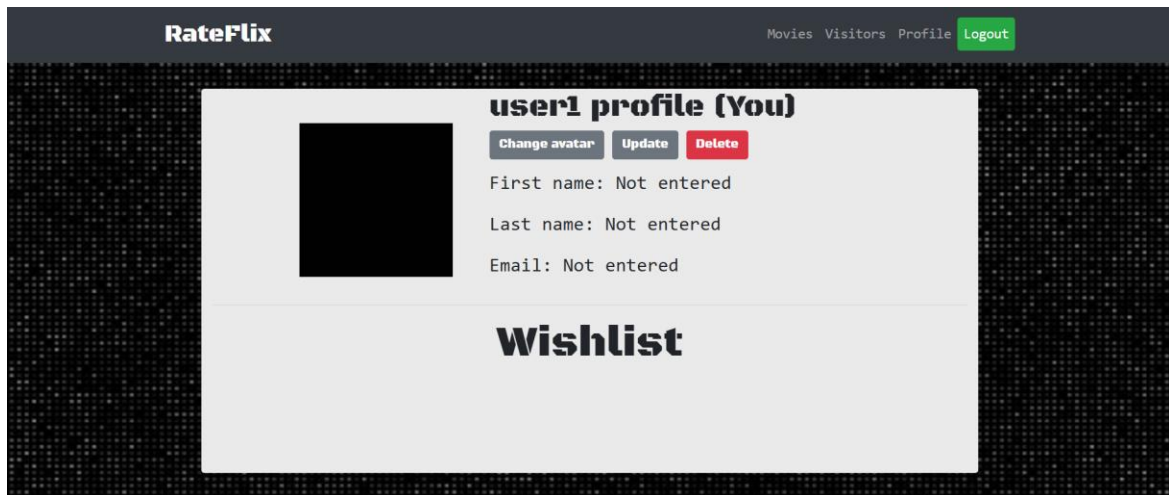


Рисунок 3.32 – Профіль користувача

На сторінці фільмів наявна можливість пошуку кінострічок за назвою або фільтрування за обраним жанром. До прикладу, рис. 3.33 демонструє наявні комедії.



Рисунок 3.33 – Сторінка фільмів за обраним жанром

Також користувач має можливість створити новий запис про фільм, вказавши назву, опис, рік випуску, обравши жанр та завантаживши постер (рис. 3.34).

Create Movie

Title*

Description

Release year*

Poster

Вибрати файл Файл не вибрано

Genres*

Action
Adventure
Animation
Biography

Submit Cancel

Рисунок 3.34 – Форма створення нового запису про фільм

Перейшовши на сторінку фільму, користувач має можливість переглянути інформацію про нього і відгуки інших користувачів (рис. 3.35).

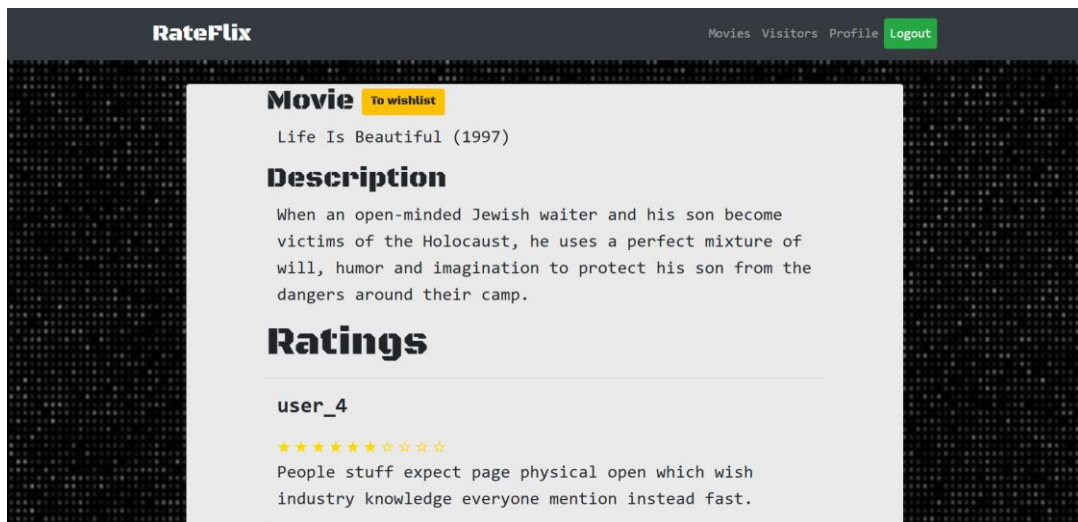


Рисунок 3.35 – Сторінка обраного фільму

На сторінці обраного фільму можна додати власний відгук зі вказанням коментаря й оцінки за 10-бальною шкалою (рис. 3.36).

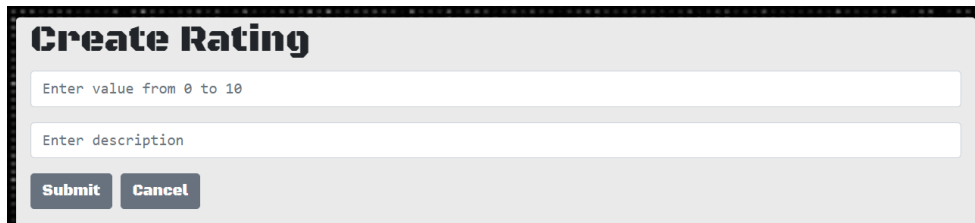


Рисунок 3.36 – Форма додавання відгуку та оцінки на фільм

Сторінка відвідувачів відображає користувачів вебдодатку, їх ім'я та розмір списку фільмів до перегляду. Також наявна можливість пошуку користувача за логіном (рис. 3.37).



Рисунок 3.37 – Сторінка відвідувачів

Висновки до розділу

У розділі було розглянуто засоби розробки, використані для реалізації вебдодатку для рецензування кінострічок, зокрема такі, як мова програмування Python, фреймворк Django, база даних PostgreSQL, розміщена

на хмарному сервісі NeonDB, середовище розробки PyCharm. Такий вибір обумовлено зручністю, гнучкістю, високою продуктивністю розробки й активною підтримкою спільноти. Також було розглянуто використані бібліотеки та вимоги до технічного й програмного забезпечення клієнта і розробника.

Розробка вебдодатку налічувала кілька основних етапів: опис середовища виконання, реалізацію безпеки доступу, бази даних та взаємодії між компонентами проєкту, які включають моделі, представлення, шаблони тощо, а також деплой сайту та підключення AWS для збереження медіа.

Керівництво користувача розкрило ключові моменти користування вебзастосунком для написання рецензій на кінострічки, включно зі створенням та редагуванням особистого профілю, пошуком фільмів та залишенням відгуків.

Результатом виконаної роботи є реалізований вебдодаток, який виконує визначені задачі та доступний онлайн через хмарну платформу Render.com.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ

4.1 Регулювання питань охорони праці на законодавчому рівні

Згідно зі статтею 43 Конституції України, кожен має право на належні, безпечні і здорові умови праці [20, ст. 43]. Це конституційне положення закладає основу для створення комплексної системи охорони праці, яка регулюється низкою нормативно-правових актів на державному рівні.

Основним законодавчим документом у цій сфері є Закон України «Про охорону праці» [21], який визначає загальні принципи, завдання, права та обов'язки сторін, пов'язаних із гарантуванням безпеки на робочих місцях. Згідно з законом, роботодавець зобов'язаний організовувати належні умови праці, які не шкодитимуть життю та здоров'ю людей під час трудової діяльності, своєчасно усувати виявлені загрози та проводити інструктажі з питань безпеки.

Також важливим нормативним документом є Кодекс законів про працю України [22], який регулює трудові відносини, зокрема положення про охорону праці, права працівників на відпочинок, компенсації за роботу в шкідливих або небезпечних умовах, а також механізми контролю за дотриманням норм безпеки.

Діяльність щодо охорони праці на підприємстві мотивується моральною і юридичною відповідальністю, а також зацікавленістю роботодавця в одержанні максимального прибутку, зменшенні витрат на пільги та компенсації за роботу в шкідливих умовах, відшкодування збитків потерпілим, штрафні санкції, ремонт пошкодженого обладнання, зменшення страхових тарифів, підвищення конкурентоспроможності підприємств [23, с. 6]. Таким чином, роботодавець відіграє важливу роль у створенні системи охорони праці в організації, відповідаючи за розробку і впровадження

політики безпеки, забезпечення працівників необхідними засобами захисту, організацію навчання та контроль за дотриманням норм і правил охорони праці.

З іншого боку, працівники також несуть відповідальність за дотримання встановлених правил безпеки, належне використання засобів індивідуального захисту, участь у навчаннях та інструктажах, а також за повідомлення про виявлені небезпечні умови чи порушення норм охорони праці. Активна співпраця між роботодавцем і працівниками є запорукою ефективного функціонування системи охорони праці в організації.

4.2 Виявлення потенційних небезпек стосовно об'єкту проєктування

Офіси ІТ-компаній є робочим простором для фахівців, які займаються розробкою, тестуванням і впровадження програмного забезпечення та іншими аспектами галузі. У таких приміщеннях зосереджена велика кількість людей та комп'ютерної техніки, що створює середовище, в якому можуть виникати різні небезпечні та шкідливі фактори. Це стосується як фізичних і психофізіологічних навантажень, так і можливих хімічних, біологічних та пожежних ризиків, тому організація ефективної системи охорони праці в офісах ІТ-компаній є надзвичайно важливою для збереження здоров'я працівників та підвищення продуктивності праці.

Фізичні небезпечні та шкідливі фактори включають:

- неправильне освітлення, яке може погано впливати на зір;
- шум від техніки, що підвищує втомлюваність та може погіршувати слух;
- невідповідні вологість, температуру та рухливість повітря, які можуть призводити до дискомфорту, зниження працездатності, переохолодження або перегрівання організму;
- випромінювання від моніторів та електромагнітні поля, які спричиняють головний біль, втому чи порушення сну;

– дія електричного струму, яка може спричинити електротравму, опіки, м'язові судоми, зупинку дихання або серця.

До хімічних небезпечних та шкідливих факторів належать:

– накопичення пилу і шкідливих речовин у системах вентиляції, що можуть спричиняти захворювання легень і алергії;

– використання сильних дезінфікуючих засобів, які подразнюють шкіру та слизові оболонки;

– випари і запахи від фарб, клеїв, тонерів для принтерів та інших офісних матеріалів, які можуть викликати подразнення дихальних шляхів, головний біль або алергічні реакції.

Біологічні небезпечні та шкідливі фактори налічують:

– патогенні мікроорганізми (бактерії, віруси, грибки), які можуть поширюватися у замкнених приміщеннях офісу через повітря або контактні поверхні, спричиняючи інфекційні захворювання;

– плісняву, яка може розвиватися у вогких або погано вентильованих зонах, викликаючи алергічні реакції та проблеми з диханням;

– алергени від рослин, що можуть провокувати алергії або астматичні реакції.

Також у робочому середовищі присутні психофізіологічні небезпечні та шкідливі фактори, зокрема:

– статичні фізичні навантаження, пов'язані з тривалим сидінням у неправильній позі, що можуть спричиняти болі в спині, порушення постави та проблеми з опорно-руховим апаратом;

– нервово-психічні перевантаження, зумовлені високим рівнем стресу, дедлайнами, інтенсивною розумовою працею, що можуть призводити до зниження концентрації, тривожності, депресії і вигорання.

Для зменшення негативного впливу розглянутих небезпечних факторів в офісах ІТ-компаній необхідно забезпечити на робочих місцях якісне освітлення, столи та крісла, що допомагають підтримувати правильну поставу, комфортні температуру та вологість повітря, належну вентиляцію, а також

зручне розташування комп'ютерної техніки для мінімізації напруги очей і м'язів. Крім того, слід передбачити заходи електробезпеки, зокрема використання справного обладнання та регулярну перевірку електромереж. Варто запроваджувати перерви для відпочинку, проводити регулярні прибирання і очищення вентиляційних систем від пилу, а також профілактичні медичні огляди.

В сукупності дані кроки допоможуть створити безпечне та комфортне робоче середовище, знизити ризик професійних захворювань, підвищити продуктивність праці та сприяти збереженню життя та здоров'я працівників ІТ-компанії.

Під час війни та після її закінчення увага до психічного здоров'я громадян повинна бути на особливо високому рівні, оскільки саме рівень психологічного благополуччя буде впливати на загальний стан здоров'я, економічне відновлення та добробут країни [24, с. 231]. Отже, роботодавці повинні створювати умови, сприятливі для збереження психічного здоров'я працівників, що включає й доступ до психологічної допомоги. Крім того, з огляду на воєнний стан, необхідно передбачити наявність облаштованих укриттів для тимчасового перебування працівників під час повітряної тривоги.

Пожежа становить загрозу для життя та здоров'я людини, а також охоплює й дію таких небезпечних та шкідливих факторів: токсичні продукти згорання, вогонь, підвищена температура середовища, дим, недостатність кисню, руйнування будівельних конструкцій, паніка [25, с. 49]. Для запобігання її виникненню в офісах ІТ-компаній необхідно забезпечити наявність справних засобів пожежогасіння, належне технічне обслуговування електромереж і електроприладів й евакуаційні шляхи, а також не допускати перевантаження розеток та регулярно проводити інструктажі з пожежної безпеки для працівників.

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проєктування та розробка заходів щодо їх попередження

Оцінка ризиків полягає у всеосяжному та систематичному виявленні небезпек та визначенні величини ризиків [26, с. 56], які виникають у процесі трудової діяльності.

Основна мета даної процедури – встановити, які потенційні загрози існують на робочому місці, оцінити ймовірність їх реалізації та наслідки для здоров'я працівників, а також визначити пріоритетність розробки та впровадження профілактичних заходів. Це сприяє створенню безпечного робочого середовища, адже оцінка ризиків дозволяє зосередити зусилля на найкритичніших аспектах охорони праці, зменшити рівень травматизму та професійних захворювань, а також підвищити загальну ефективність функціонування організації.

Завдання оцінки ризиків є наступними:

- виявлення потенційно небезпечних та шкідливих факторів, які можуть впливати на працівників;
- визначення ймовірності та серйозності наслідків визначених загроз на життя та здоров'я працівників;
- формулювання та впровадження заходів безпеки, спрямованих на усунення або зменшення впливу ризиків;
- відстежування ефективності впровадження заходів.

Розглядаючи ризики, що можуть виникати в умовах роботи в офісі ІТ-компанії, важливо враховувати, що переважна більшість виробничих процесів здійснюється з застосуванням електронної техніки, мережевих комунікацій та складних кабельних систем. Таке середовище зумовлює підвищену ймовірність виникнення електричних небезпек, зокрема й ураження електричним струмом, що несе загрозу життю та здоров'ю людини. Наслідками можуть бути електричні опіки, зупинка серця та порушення функцій нервової системи.

Ураження електричним струмом є можливими через низку факторів, включно з пошкодженими або неправильно прокладеними дротами, використанням несправної техніки, порушенням правил електромонтажу, а також нехтуванням правилами техніки безпеки під час експлуатації електроприладів (рис. 4.1).

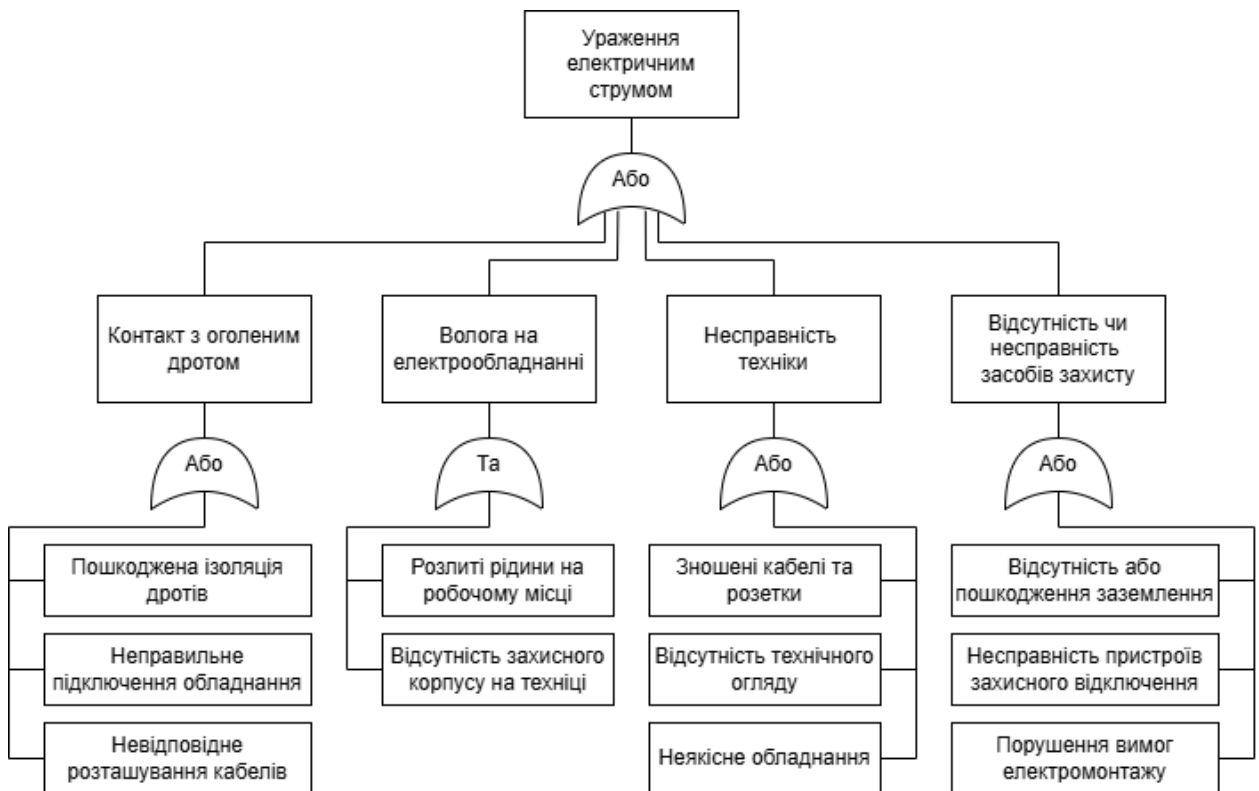


Рисунок 4.1 – Дерево відмов для аналізу небезпеки ураження електричним струмом в офісі ІТ-компанії

Оскільки ураження електричним струмом становить велику небезпеку для працівників, важливо вживати заходи безпеки та усувати наявні недоліки. Регулярна перевірка електричних систем та їх своєчасне обслуговування значно зменшують імовірність виникнення аварійних ситуацій, пов'язаних з короткими замиканнями, перегрівом обладнання чи порушенням ізоляції дротів. Для забезпечення обізнаності працівників щодо потенційних ризиків та правильних дій у разі виникнення небезпечної ситуації необхідно проводити інструктажі, а для уникнення випадкових пошкоджень кабелів на

робочих місцях варто враховувати їх розміщення та використовувати маркування. Додатковий рівень захисту гарантується завдяки використанню лише сертифікованого обладнання та наявності автоматичних вимикачів, заземлення і пристроїв захисного відключення. Перелічені заходи допоможуть знизити ризик ураження електричним струмом в офісі ІТ-компанії.

Температура приміщення відіграє важливу роль у створенні комфортних і безпечних умов праці. Якщо в офісі буде низька температура повітря, працівникам загрожуватимуть переохолодження та простудні захворювання, що, зі свого боку, негативно вплине на працездатність. Причини даної небезпеки наведено на рисунку 4.2.



Рисунок 4.2 – Дерево відмов для аналізу небезпеки від низької температури в офісі ІТ-компанії

Підвищена температура повітря в офісі також може становити небезпеку, оскільки знижує концентрацію уваги та спричиняє головний біль, втому, зневоднення, тепловий стрес. Причинами такої загрози можуть бути проблеми з вентиляцією чи кондиціонуванням та тепловиділення від техніки (рис. 4.3).



Рисунок 4.3 – Дерево відмов для аналізу небезпеки від високої температури в офісі ІТ-компанії

Комфортна температура повітря в офісі ІТ-компанії є важливою складовою належних умов праці, для її гарантування необхідно забезпечити надійну систему опалення та кондиціонування, підтримувати належний рівень вентиляції та регулярно контролювати параметри мікроклімату. Також варто проводити планове технічне обслуговування обладнання та своєчасно виявляти та усувати несправності.

Висновки до розділу

У розділі було розглянуто ключові аспекти гарантування безпеки та здорових умов праці на робочих місцях, зокрема в офісах ІТ-компаній. Метою було дослідження чинників, що впливають на охорону праці, виявлення потенційних небезпек, які можуть виникати під час трудової діяльності, а також формулювання конкретних заходів для зниження ризиків і покращення робочого середовища.

У роботі проаналізовано основні нормативно-правові документи, що регулюють питання охорони праці, зокрема Конституцію України, Закон України «Про охорону праці» та Кодекс законів про працю. Визначено відповідальність роботодавця за створення безпечних умов праці, а також

обов'язки працівників щодо дотримання правил безпеки та використання засобів індивідуального захисту.

Було проведено аналіз потенційних небезпек, характерних для офісного середовища ІТ-компаній, що включають фізичні, хімічні, біологічні, психофізіологічні фактори. Для зниження їх впливу запропоновано низку практичних рішень: організація ергономічних робочих місць, забезпечення відповідного рівня освітлення і мікроклімату, регулярне технічне обслуговування електроприладів і вентиляції, проведення інструктажів з пожежної безпеки, запровадження перерв для відпочинку, а також підтримка психічного здоров'я працівників, що особливо актуально в умовах воєнного стану. Розроблені заходи спрямовані на створення комфортного та безпечного середовища, що знижує ризики травматизму і професійних захворювань, а також підвищує продуктивність праці.

ВИСНОВКИ

У ході кваліфікаційної роботи було розглянуто ключові аспекти створення вебдодатку для рецензування кінострічок. Зокрема, було проаналізовано предметне середовище й наявні аналоги, що дозволило сформулювати задачі для подальшого дослідження.

Аналіз предметної області включно з визначенням ролей та об'єктів системи, а також вхідних і вихідних даних, дозволив визначити архітектуру системи та сформулювати вимоги до її функціональності. В роботі було розглянуто проектування бази даних за ключовими сутностями та побудовано об'єктно-орієнтовану модель. Алгоритмічне забезпечення з побудовою схем взаємодії користувача з системою стало основою для подальшої реалізації логіки вебдодатку.

Застосування таких засобів, як мова програмування Python, фреймворк Django, база даних PostgreSQL, розміщена на хмарному сервісі NeonDB та середовище розробки PyCharm, у розробці вебдодатку було зумовлено широкими можливостями для розробки та підтримки вебсервісів. Крім того, було розглянуто вимоги до технічного та програмного забезпечення як з боку користувача, так і з боку розробника. Програмна реалізація включала опис середовища виконання, реалізацію безпеки доступу, бази даних та взаємодії між компонентами проєкту, які включають моделі, представлення, шаблони, а також деплой сайту та підключення AWS для збереження медіа.

Окремо було розглянуто роль законодавства у питаннях охорони праці у процесі трудової діяльності працівників ІТ-сфери, а також визначено основні ризики, як виникають у професійному середовищі.

У результаті роботи було досягнуто поставлену мету – розробку функціонального вебдодатку для рецензування та оцінювання кінострічок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Черниш Д.О., Новожилова М.В. ВИКОРИСТАННЯ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ВИЯВЛЕННЯ ФІНАНСОВОГО ШАХРАЙСТВА В ІНДУСТРІЇ ОНЛАЙН-ГЕМБЛІНГУ. Інформаційні технології: теорія і практика: Тези VIII (II) Міжнародної Інтернет-конференції здобувачів вищої освіти і молодих учених (Запоріжжя-Харків-Дніпро, 2-4 квітня 2025 р.). – Запоріжжя : НУ «Запорізька політехніка», 2025. С. 168–169. [Електронний ресурс] // Режим доступу: https://knameedu.sharepoint.com/sites/msteams_73a185/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmsteams%5F73a185%2FShared%20Documents%2FGeneral%2F%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%20%D0%B4%D1%96%D1%8F%D0%BB%D1%8C%D0%BD%D1%96%D1%81%D1%82%D1%8C%2F2025%5FInternet%5Fconference%2F%D0%97%D0%B1%D1%96%D1%80%D0%BA%D0%B0%20%D0%86%D0%A2%D0%A2%D0%9F%5F2025%2Epdf&parent=%2Fsites%2Fmsteams%5F73a185%2FShared%20Documents%2FGeneral%2F%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%20%D0%B4%D1%96%D1%8F%D0%BB%D1%8C%D0%BD%D1%96%D1%81%D1%82%D1%8C%2F2025%5FInternet%5Fconference&p=true&ga=1
2. Saksham Sharma E. a. Review Paper on Analysis of Human Behaviour on Movie Reviews. International Journal on Recent and Innovation Trends in Computing and Communication. 2023. Vol. 11, no. 9. P. 3533–3538. [Електронний ресурс] // Режим доступу: <https://doi.org/10.17762/ijritcc.v11i9.9572>
3. IMDb Statistics and User Count for 2024 [Електронний ресурс] // Режим доступу: <https://expandedramblings.com/index.php/imdb-facts-statistics>
4. Головна сторінка IMDb [Електронний ресурс] // Режим доступу: <https://www.imdb.com/>

5. Головна сторінка Letterboxd [Електронний ресурс] // Режим доступу: <https://letterboxd.com/>
6. Головна сторінка Rotten Tomatoes [Електронний ресурс] // Режим доступу: <https://www.rottentomatoes.com/>
7. Python Programming Language [Електронний ресурс] // Режим доступу: <https://www.python.org/>
8. Django Web Framework [Електронний ресурс] // Режим доступу: <https://www.djangoproject.com/>
9. TIOBE Software BV. “TIOBE Programming Community Index – June 2025.” TIOBE Index [Електронний ресурс] // Режим доступу: <https://www.tiobe.com/tiobe-index/>
10. Радоуцький К. Є., Гордієнко М. А. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ЗА ДОПОМОГОЮ PYTHON-ФРЕЙМВОРКУ. Матеріали V міжнародної науково-практичної конференції «Інформаційні моделюючі технології, системи та комплекси». 18-19 квітня 2024 р., Черкаси, Україна. – Черкаси : Черкаський національний університет імені Богдана Хмельницького, 2024. – С. 124–126 [Електронний ресурс] // Режим доступу: https://fotius.cdu.edu.ua/wp-content/uploads/2024/05/Book_IMTCK_2024.pdf#page=124
11. PostgreSQL Global Development Group. PostgreSQL: The World's Most Advanced Open Source Relational Database [Електронний ресурс] // Режим доступу: <https://www.postgresql.org/>
12. PyCharm: the Python IDE for Professional Developers [Електронний ресурс] // Режим доступу: <https://www.jetbrains.com/pycharm/>
13. django-crispy-forms. Django Crispy Forms [Електронний ресурс] // Режим доступу: <https://django-crispy-forms.readthedocs.io/>
14. django-debug-toolbar. Django Debug Toolbar [Електронний ресурс] // Режим доступу: <https://django-debug-toolbar.readthedocs.io/>
15. django-storages. Django Storages [Електронний ресурс] // Режим доступу: <https://django-storages.readthedocs.io/>

16. boto3. Boto3: AWS SDK for Python [Електронний ресурс] // Режим доступу: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>
17. dj-database-url. dj-database-url: Use Database URLs in Django [Електронний ресурс] // Режим доступу: <https://pypi.org/project/dj-database-url/>
18. Render. Render: The Modern Cloud for Application Hosting [Електронний ресурс] // Режим доступу: <https://render.com>
19. Neon. Neon: Serverless Postgres [Електронний ресурс] // Режим доступу: <https://neon.tech>
20. Конституція України : від 28.06.1996 № 254к/96-ВР : станом на 1 січ. 2020 р. [Електронний ресурс] // Режим доступу: <https://zakon.rada.gov.ua/laws/show/254%D0%BA/96-%D0%B2%D1%80#Text>
21. Про охорону праці : Закон України від 14.10.1992 № 2694-ХІІ : станом на 4 квіт. 2025 р. [Електронний ресурс] // Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>
22. Кодекс законів про працю України : Кодекс України від 10.12.1971 № 322-VIII : станом на 2 трав. 2025 р. [Електронний ресурс] // Режим доступу: <https://zakon.rada.gov.ua/laws/show/322-08#Text>
23. Швагер Н. Ю., Янов Є. К. Сучасні підходи до оптимізації системи управління охороною праці підприємств. Матеріали Міжнародної науково-технічної конференції «Розвиток промисловості та суспільства». Кривий Ріг. 2021. Криворізький національний університет. С. 6 [Електронний ресурс] // Режим доступу: [https://www.knu.edu.ua/storage/files/2/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0%D0%9A%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D1%96%D1%97/%D0%9A%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D1%96%D1%97%202021/%D0%A0%D0%BE%D0%B7%D0%B2%D0%B8%D1%82%D0%BE%D0%BA/%D0%A2%D0%B5%D0%B7%D0%B8_%202021_%20%D0%9E%D0%9A2.pdf#page=](https://www.knu.edu.ua/storage/files/2/%D0%9D%D0%B0%D1%83%D0%BA%D0%B0%D0%9A%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D1%96%D1%97/%D0%9A%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D1%96%D1%97%202021/%D0%A0%D0%BE%D0%B7%D0%B2%D0%B8%D1%82%D0%BE%D0%BA/%D0%A2%D0%B5%D0%B7%D0%B8_%202021_%20%D0%9E%D0%9A2.pdf#page=8)

24. Горбенко В.С., Шароватова О.П. Психосоціальна підтримка на робочому місці в контексті розвитку соціальних аспектів охорони праці в сучасній Україні. Матеріали III Міжнародної науково-практичної інтернет-конференції студентів та молодих науковців «Актуальні питання охорони праці у контексті сталого розвитку та європейської інтеграції України», Харків, 09–11 листоп. 2022 р. : тези доп. / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2022. С. 231–233. [Електронний ресурс] // Режим доступу: https://science.kname.edu.ua/images/dok/konferentsii/2022/Tezy_2022/_9-11_11_22.pdf#page=231

25. Шевченко В.В. Теоретичні засади організації пожежної безпеки в контексті забезпечення системи охорони праці. Збірник наукових праць Кам'янець-Подільського національного університету імені Івана Огієнка. Серія педагогічна №27. 2021. С. 47–52. [Електронний ресурс] // Режим доступу: <http://ped-series.kpnu.edu.ua/article/view/251657>

26. Курепін В.М. Оцінка ризиків: критерії оцінки та їх вплив на профілактику виробничих ризиків. Збірник наукових праць за науковими матеріалами III Всеукраїнської науково–практичної конференції викладачів та фахівців–практиків «Охорона праці: освіта і практика» та XIII Всеукраїнської науково-практичної конференції курсантів, студентів, аспірантів та ад'юнктів «Проблеми та перспективи розвитку охорони праці». Львів. 2023. С. 56–58. [Електронний ресурс] // Режим доступу: https://dspace.mnau.edu.ua/jspui/bitstream/123456789/14161/1/OP-2023_29.05-56-58.pdf