

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра інформаційно-обчислювальних систем і управління

**Ліпов Богдан Богданович**

**Веб-орієнтована інформаційна система анонсів кінопродукції /  
Web-oriented information system of film production  
announcements**

Спеціальність 122 – Комп'ютерні науки  
Освітньо-професійна програма – Комп'ютерні науки

Дипломний проект

Виконав студент групи КН-42  
Ліпов Б.Б

---

Науковий керівник:  
к.т.н., доцент Турченко І.В

---

Дипломний проект допущено до  
захисту

« \_\_\_ » \_\_\_\_\_ 2023 р.

Завідувач кафедри  
\_\_\_\_\_ М.П. Комар

**Тернопіль – 2023**

Факультет комп'ютерних інформаційних технологій  
Кафедра інформаційно-обчислювальних систем і управління  
Освітній ступінь «бакалавр»  
Спеціальність 122 – Комп'ютерні науки  
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
\_\_\_\_\_ М.П. Комар  
« \_\_\_\_\_ » \_\_\_\_\_ 2022р.

З А В Д А Н Н Я  
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТА

Ліпов Богдан Богданович

(прізвище, ім'я, по батькові)

1. Тема проекту: веб-орієнтована інформаційна система анонсів кінопродукції/Web-oriented information system of film production announcements

керівник проекту к.т.н., доцент Турченко. І. В.

затверджені наказом по університету від 08 грудня 2022 р. № 491.

2. Строк подання студентом закінченого проекту 01 червня 2023 р.

3. Вихідні дані до проекту: технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- провести аналіз предметної області;
- проаналізувати існуючі аналоги;
- розробити структуру веб-орієнтованої інформаційної системи;
- розробити алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
- розробити інтерфейс інформаційної системи;
- реалізувати та протестувати веб-орієнтовану інформаційну систему.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
- схема алгоритму роботи веб-орієнтованої інформаційної системи ;
  - діаграма станів та переходів веб-орієнтованої інформаційної системи;

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., доцент Турченко І.В		

7. Дата видачі завдання 08 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області	30.12.2022	
2	Загальна структура та інформаційне забезпечення системи веб-орієнтованої інформаційної системи анонсів кінопродукції	24.03.2023	
3	Програмно-технологічне забезпечення веб-орієнтованої інформаційної системи анонсів кінопродукції	12.05.2023	
4	Повне завершення та оформлення дипломного проекту	01.06.2023	

Студент \_\_\_\_\_ Ліпов Б.Б  
( підпис )

Керівник проекту \_\_\_\_\_ Турченко І. В  
( підпис )

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту: 53 с., 23 рис., 1 табл., 3 додатки, 21 джерело.

Метою дипломного проекту є розробка веб-орієнтованої інформаційної системи анонсів кінопродукції.

Об'єкт дослідження – процеси анонсування кіно продукції.

Предмет дослідження – інформаційної системи анонсів кінопродукції.

Розроблено веб-орієнтовану інформаційну систему анонсів кінопродукції.

Реалізована веб-орієнтована інформаційна система анонсів кінопродукції надає користувачеві можливість ознайомитися з актуальною інформацією про майбутні фільми, надає опис, дату виходу, простер. Також надає можливість ознайомитися з останніми прем'єрами. Може використовуватися кінотеатрами та кіностудіями для анонсування їх продукції.

АНОНСИ КІНОПРОДУКЦІЇ, ВЕБ-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА, ФРЕЙМВОРК, БАЗА ДАНИХ .

## ABSTRACT

Explanatory note of the diploma project: 53 p., 23 figures, 1 table, 3 appendices, 21 sources.

The goal of the diploma project is the development of web-oriented information system of film production announcements.

The object of research is processes of announcing film productions.

The subject of the research is information system of film production announcements.

Web-oriented information system of film production announcements is developed.

Implemented web-oriented information system of film production announcements provides the user with the opportunity to read up-to-date information about future films, provides a description, release date, location. It also provides an opportunity to get acquainted with the latest premieres. It can be used by cinemas and film studios to announce their products.

FILM PRODUCTION ANNOUNCEMENTS, WEB-ORIENTED INFORMATION SYSTEM, FRAMEWORK, DATABASE.

# ТЕХНІЧНЕ ЗАВДАННЯ

## 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

1.1 Веб-орієнтована інформаційна система анонсів кінопродукції.

1.2 Область застосування – анонсування кінопродукції.

## 2. ОСНОВА ДЛЯ РОЗРОБЛЕННЯ

Основою для розроблення є завдання на дипломний проект, затверджене кафедрою інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

## 3. ПРИЗНАЧЕННЯ РОЗРОБЛЕНОГО КОМПЛЕКСУ

Метою дипломного проекту є розробка веб-орієнтованої інформаційної системи анонсів кінопродукції.

## 4. ДЖЕРЕЛА РОЗРОБЛЕННЯ

Джерелами даної розробки є матеріали навчальної і реферативної літератури, технічна документація, науково-дослідні статті, журнали, Інтернет.

## 5. ТЕХНІЧНІ ВИМОГИ

5.1 Основні функціональні вимоги до програмної системи:

- реєстрація та автентифікація користувачів;
- можливість керувати анонсами фільмів (додавання нових анонсів фільмів, редагування існуючих, відображення списку анонсів);
- перегляд докладної інформації про фільм;
- адміністративний доступ до інтерфейсу Django для керування базою даних.

## 5.2 Вимоги до апаратних засобів:

– мінімальні вимоги для запуску на фізичній машині оперативна пам'ять 2 гігабайти та процесор 2 GHz.

## 5.3 Вимоги до програмних засобів:

– для розробки програмне забезпечення - Python 3.6 і вище включно з фреймворк Django;

– для створення графічного інтерфейсу користувача – HTML, CSS, Bootstrap;

## 6. ПОРЯДОК КОНТРОЛЮ

6.1 Представлення дипломного проекту на попередній захист.

6.2 Представлення дипломного проекту на захист.

Завдання прийняв до виконання \_\_\_\_\_ Ліпов Б.Б.  
( підпис ) (прізвище та ініціали)

Керівник дипломного проекту \_\_\_\_\_ Турченко І.В.  
( підпис ) (прізвище та ініціали)

## ЗМІСТ

Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Коротка характеристика об'єкта управління.....	11
1.2 Опис предметної області .....	12
1.3 Огляд і аналіз існуючих аналогів .....	15
1.4 Постановка задачі проектування .....	18
2 Загальна структура та інформаційне забезпечення системи.....	21
2.1 Загальна структура проектованої системи .....	21
2.2 Алгоритмічне забезпечення системи .....	23
2.3 Інформаційне забезпечення.....	27
3 Програмно-технологічне забезпечення.....	33
3.1 Програмна реалізація .....	33
3.2 Інтерфейс користувача.....	38
3.3 Тестування програмного забезпечення.....	41
Висновки.....	47
Список використаних джерел.....	48
Додаток А Схема Алгоритму роботи веб-орієнтованої інформаційної системи	50
Додаток Б Діаграма сатнів та переходів веб-орієнтованої інформаційної системи.....	51
Додаток В Код програми обробки сторінок.....	52

					ДП.КН.9702718.053.ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розробив		Ліпов Б. Б.			Веб-орієнтована інформаційна система анонсів кінопродукції	Літ.	Арк.	Аркушів
Перевір.		Турченко І. В.				8	53	
Н. Контр.		Турченко І. В.			ЗУНУ, ФКІТ, КН-42			
Затвердив		Комар М. П.						



## ВСТУП

Кіноіндустрія є однією з найбільш популярних та прибуткових галузей розваг, тому є велике зростання кількості нових кіно робіт та серіалів. Це ставить перед глядачами завдання знаходити швидкий та зручний спосіб ознайомлення з найбільш очікуваними фільмами.

Особливо у зв'язку з початком пандемії COVID-19 у 2020 році, коли кінотеатри закривалися, а глядачі мали можливість дивитися фільми лише в онлайн-форматі.

У таких умовах створення веб-орієнтованої інформаційної системи анонсів кінопродукції, що забезпечить швидкий та зручний доступ до анонсів кінопродукції, є дуже актуальною задачею.

Метою дипломного проекту є створення веб-орієнтованої інформаційної системи анонсів кінопродукції.

Для реалізації мети проекту необхідно виконати наступні завдання:

– провести детальний аналіз потреб користувачів і визначити основні функціональні та нефункціональні вимоги до системи. Врахувати потреби аудиторії, яку ви хочете залучити, такі як кіноманів, шанувальників конкретних жанрів або акторів, та вимоги щодо доступності, швидкості завантаження тощо.

– розробити інтуїтивно зрозумілий та привабливий інтерфейс системи. Врахувати зручну навігацію, можливість швидкого пошуку та фільтрації фільмів, візуальне відображення інформації про фільми, можливість оцінювання, коментування та обміну відгуками.

– створити структуру бази даних, яка буде зберігати інформацію про фільми, акторів, режисерів, жанри, описи, зображення, відео, рейтинги та інші відомості. Забезпечити ефективне зберігання та доступ до даних.

– розробити функціональні модулі для додавання, редагування та видалення інформації про фільми, відображення списку фільмів, пошуку та фільтрації, коментування та оцінювання фільмів, створення списків перегляду, рекомендацій та інші функції, які ви визначили на основі аналізу потреб користувачів.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

– забезпечити можливість інтеграції зовнішніх сервісів, таких як бази даних фільмів, API для отримання даних про фільми, платформи для онлайн-прокату або покупки квитків, для забезпечення повноцінного функціоналу системи.

– виконати тестування системи, перевірити її функціональність, безпеку та продуктивність. Виправити помилки та відгладити проблеми, що виникають.

– розгорнути систему на веб-сервері, забезпечити її доступність користувачам. Забезпечити регулярне оновлення і підтримку системи, включаючи виправлення помилок, оновлення функціоналу та забезпечення безпеки даних.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Коротка характеристика об'єкта управління

Кінематограф є галуззю культури та економіки, яка об'єднує різні види професійної діяльності, пов'язані з виробництвом, поширенням, зберіганням та демонстрацією фільмів. Він також має значення в навчально-науковій сфері, як відомо з кінознавства.

Основною складовою кінематографу є кіноіндустрія, що виробляє кінофільми, спец ефекти та анімацію. У багатьох країнах кіноіндустрія є важливою галуззю економіки. Виробництво фільмів зазвичай здійснюється у спеціалізованих кіностудіях. Готові фільми демонструються у кінотеатрах, по телебаченню та поширюються на різних носіях для перегляду на домашній аудіо-відео техніці, наприклад, на відеокасетах та відеодисках.

Кінопродукція - це галузь індустрії, яка займається створенням фільмів або кіно продуктів. Вона включає в себе весь процес створення фільму, починаючи від розробки ідеї, написання сценарію, збору фінансування, вибору акторів і знімальної групи, проведення зйомок, монтажу, звукового оформлення, створення спеціальних ефектів, до дистрибуції та показу фільму на екрані кінотеатрів, телебаченні або в Інтернеті.

Кінопродукція може включати різні жанри фільмів, такі як драма, комедія, екшн, фантастика, жахи, анімація тощо. Вона може бути представлена як художніми фільмами, так і документальними стрічками. Кінопродукція є миттєвим медіа, яке сприймається через візуальні, аудіо та нормативні елементи, що дозволяє створювати враження, розповідати історії, передавати емоції та відтворювати реальність або уявні світи.

Кінопродукція є дуже популярним видом розваги для багатьох людей по всьому світу. Щороку виходять сотні нових фільмів, інформація про які потребується для великої кількості користувачів. Однак, існуючі інформаційні ресурси не завжди забезпечують потрібну якість та кількість інформації про фільми.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

## 1.2 Опис предметної області

Веб-орієнтована інформаційна система є програмою, яка виконує функції через веб-браузер, незалежно від пристрою та платформи. Він може бути простим наприклад, дошкою оголошень або контактною формою на Веб-орієнтована інформаційна система, або складним як текстовий процесор або мобільна гра з багатьма користувачами.

Веб-сайт - це група взаємопов'язаних веб-сторінок, доступних глобально, і має єдине доменне ім'я. Веб-сайт може бути розроблений і підтримуваний окремою особою, бізнесом або організацією. Веб-сайти мають різні цілі, такі як сайти-візитки, блоги, новинні портали та інші [1].

Основні функції веб-сайту та веб-орієнтованої інформаційної системи включають:

- це ефективним способом представити свої продукти та послуги широкій аудиторії;
- розробка допомагає створити довіру та демонструє професійність і надійність вашого бренду;
- допомагає зміцнити бренд вашого бізнесу, створюючи унікальний дизайн та спілкування зі споживачами;
- може сприяти досягненню вашої бізнес-стратегії, просуваючи продукти, збільшуючи продажі або привертаючи нових клієнтів;
- може надати різноманітні інструменти для взаємодії зі споживачами, наприклад, контактні форми, онлайн-чати або FAQ-розділи.

Ці функції веб-сайту допоможуть власникам досягти своїх цілей та задовольнити потреби цільової аудиторії.

Веб програмування на Python - це процес створення та розробки веб-додатків, використовуючи мову програмування Python. Ця предметна область об'єднує різні технології, інструменти та концепції, які дозволяють розробникам створювати динамічні та інтерактивні веб-додатки.

										ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата							12

Python є однією з потужних та популярних мово програмування для веб-розробки. Вона має чистий та зрозумілий синтаксис, що дозволяє швидко та легко розробляти веб-додатки [13]. Для веб-програмування на Python існує багато фреймворків, таких як Django, Flask, Pyramid та інші, які надають готові інструменти та структуру для розробки веб-додатків.

У цій предметній області важливо мати розуміння HTTP (протоколу передачі гіпертексту), який використовується для комунікації між веб-браузерами та веб-серверами. Також необхідно мати знання про бази даних, такі як MySQL, SQLite або PostgreSQL, які використовуються для зберігання даних у веб-додатках.

Основні завдання веб-програмування на Python включають обробку запитів від користувачів, генерацію веб-сторінок з динамічним вмістом, взаємодію з базами даних, роботу з формами, автентифікацію та авторизацію користувачів, реалізацію API для зовнішніх сервісів, оптимізацію продуктивності та безпеку веб-додатків.

Веб програмування на Python є широко використовуваною та перспективною областю, яка дозволяє розробникам створювати потужні та інноваційні веб-додатки для різних цілей, включаючи електронну комерцію.

Електронна комерція використовується для продажу товарів та послуг через Інтернет. За допомогою веб-програмування на Python, HTML, CSS і JavaScript можна створити онлайн-магазини, електронні платформи для торгівлі товарами та послугами, системи обробки платежів, управління запасами, корзини покупок, авторизацію та автентифікацію користувачів, аналітику продажів та багато іншого.

За допомогою Python та його фреймворків, таких як Django або Flask, розробники можуть швидко створити потужні та безпечні веб-додатки для електронної комерції. HTML та CSS використовуються для створення зручного і привабливого інтерфейсу користувача, а JavaScript додає інтерактивність до сторінок, наприклад, можливість динамічного оновлення вмісту без перезавантаження сторінки або розрахунок загальної суми покупки у реальному часі.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Електронна комерція на основі веб-програмування на Python дозволяє підприємствам будь-якого розміру створювати та розширювати свої онлайн-присутності, забезпечувати зручний спосіб продажу продуктів та послуг, привертати та зберігати клієнтів, а також вести аналітику для покращення бізнес-процесів і прийняття стратегічних рішень.

Реалізація веб-сайту на Python має кілька переваг:

– універсальність так як Python є високорівневою, універсальною мовою програмування, яка підтримує різні типи веб-реалізацій, включаючи фреймворки Django і Flask. Можна вибрати той фреймворк, який найбільше під йде потребами проекту;

– Python має простий і лаконічний синтаксис, що сприяє швидкому освоєнню мови програмування. Він також має велику спільноту розробників, яка надає документацію, підручники та підтримку;

– Python має широкий вибір бібліотек і фреймворків, які полегшують розробку веб-сайтів. Наприклад, Django - один з найпопулярніших фреймворків Python, який надає готові рішення для роботи з базами даних, автентифікації та багатьом іншим аспектам веб-розробки;

– завдяки простоті синтаксису Python та наявності готових фреймворків, розробка веб-сайту на Python може бути швидкою і ефективною. Багато задач, які зазвичай вимагають багато коду в інших мовах програмування, можуть бути здійснені всього кількома рядками коду у Python;

– Python має велику кількість сторонніх бібліотек, які дозволяють розширювати можливості веб-сайту. Можна легко підключити бібліотеки для обробки зображень, роботи зі сторонніми сервісами, аналітики, машинного навчання та інших завдань;

– Python є платформо незалежною мовою, що означає, що веб-сайт може бути розгорнутий на різних операційних системах, таких як Windows, macOS або Linux. Python також підтримує велику кількість баз даних і протоколів комунікації.

В цілому, реалізація веб-сайту на Python дозволяє швидко розробити потужний та функціональний веб-додаток за допомогою зручних інструментів та

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

готових рішень. Python є популярним вибором серед розробників через свою простоту, гнучкість та велику спільноту.

### 1.3 Огляд і аналіз існуючих аналогів

Для аналізу аналогів було обрано такі веб-платформи: IMDb, Rotten Tomatoes, Letterboxd.

На рисунку 1.1 зображено головну сторінку IMDb [2], що є одною з найбільш відомих веб-платформ про кінопродукцію. Основною перевагою є наявність великої бази даних з інформацією про фільми, серіали, акторів та інших фахівців кіноіндустрії. Користувачі можуть знайти детальні відомості про фільми, переглянути трейлери, оцінки, рецензії та багато іншого.

Один з недоліків IMDb полягає у тому, що оцінки фільмів можуть бути суб'єктивними та не завжди відображати справжню якість фільму. Крім того, платформа має недостатньо інтерактивних функцій для спілкування та обміну думками з іншими користувачами.

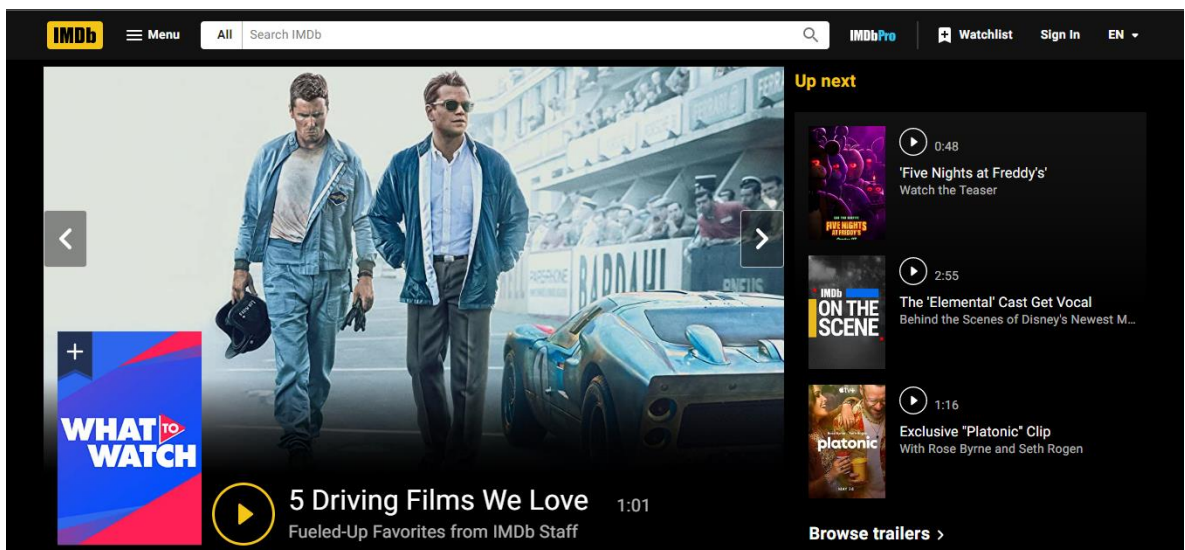


Рисунок 1.1 – Головна сторінка платформи IMDb

На рисунку 1.2 можна побачити головну сторінку веб-ресурсу Rotten Tomatoes [3].

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Rotten Tomatoes відомий своїми агрегованими оцінками критиків та глядачів. Веб-орієнтована система надає користувачам можливість оцінювати фільми, читати рецензії, дивитися трейлери та знайти інформацію про фільми. Агреговані оцінки дозволяють отримати загальний погляд на якість фільму з різних джерел.

Головним недоліком Rotten Tomatoes є можливість спотворення оцінок через феномен «оцінка аудиторії до виходу фільму», коли оцінки ставляться до фактичного перегляду фільму. Деякі критики також критикують методику обчислення агрегованих оцінок та їхню точність.

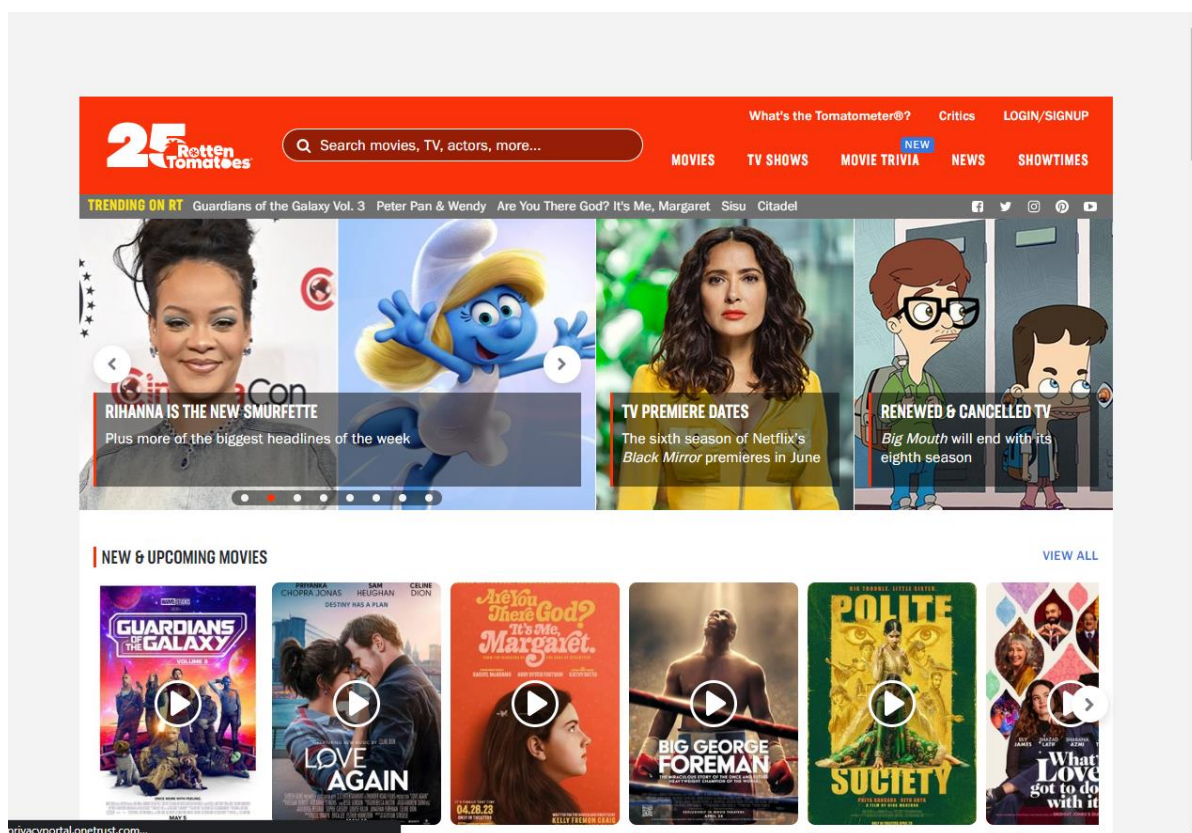


Рисунок 1.2 - Rotten Tomatoes головна сторінка

На рисунку 1.3 можна побачити головну сторінку Letterboxd [4].

Letterboxd є соціальною платформою для кіно любителів, де користувачі можуть вести список переглянутих фільмів, відзначати їх рейтингами, писати рецензії та обмінюватися враженнями з іншими користувачами. Веб-ресурс також надає зручний інтерфейс для пошуку фільмів та детальну інформації про них.

										ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата							16



Недоліком є те, що Letterboxd є соціальною платформою, і це в свою чергу може обмежувати доступ до широкої бази даних та розширеного функціоналу порівняно з іншими веб-сайтами. Деякі користувачі відмічають необхідність платного абонементу для повного функціоналу.

Отож під час огляду і аналізу існуючих аналогів було виявлено декілька систем, що вже існують на ринку та надають схожий функціонал. Було проаналізовано їх основні переваги та недоліки.

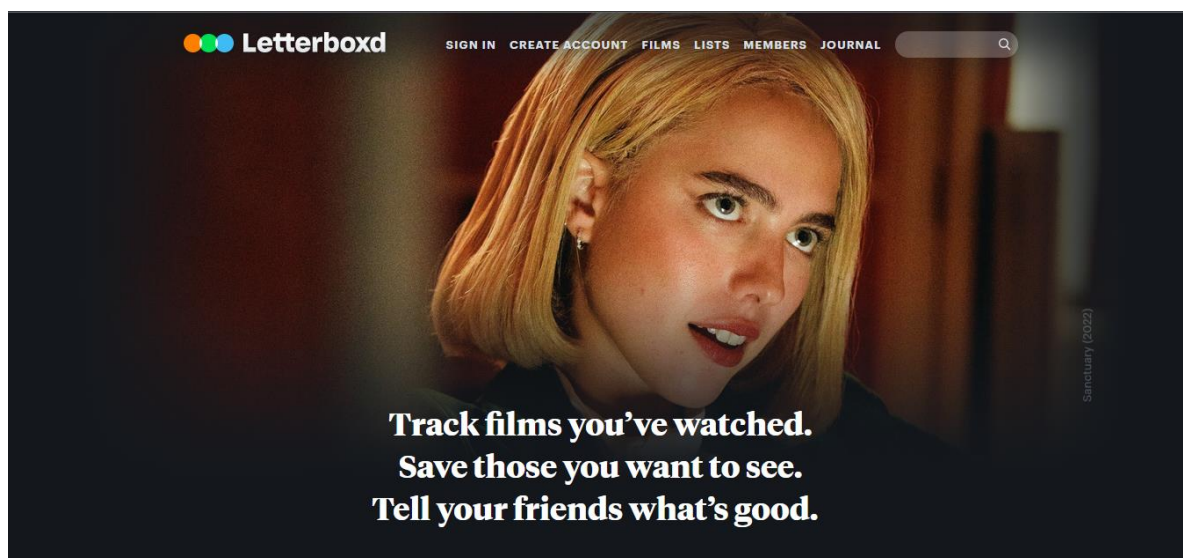


Рисунок 1.3 – Головна сторінка Letterboxd

Під час аналізу існуючих веб-платформ та систем для анонсів кінопродукції було виявлено, що багато з них мають недоліки, що ускладнюють користування та навіть відштовхують від використання.

Наприклад, деякі ресурси мають заплутаний та незручний інтерфейс, що ускладнює пошук потрібної інформації. Інші - не пропонують детальних описів фільмів, що може ставити глядача в складну ситуацію при виборі фільму для перегляду.

Також виявлено, що багато веб-ресурсів мають обмеження на кількість запитів або вимагають реєстрації користувачів, що може бути незручним для більшості користувачів.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Ця інформація буде корисною при подальшій розробці та вдосконаленні веб-інформаційної системи кінопродукції, оскільки дозволить уникнути деяких недоліків існуючих аналогів і надати користувачам більш зручний та привабливий продукт.

Враховуючи результати огляду і аналізу існуючих аналогів, можна зробити висновок, що існує певна ніша для нової системи, яка забезпечить зручний та простий у використанні інтерфейс користувача, гнучкі можливості налаштування та широкий функціонал.

Отже, з урахуванням виявлених недоліків, було вирішено розробити нову веб-орієнтовану інформаційну систему для анонсів кінопродукції, яка буде простою у використанні та матиме зручний та лаконічний інтерфейс.

Недоліки існуючих рішень свідчать про потребу в створенні нової веб-орієнтованої інформаційної системи анонсів кінопродукції. Основною метою такої системи є забезпечення зручного та швидкого доступу користувачів до актуальної та повної інформації про кінофільми, що виходять у прокат, їх опису, трейлерів та інформації про сеанси у кінотеатрах.

Задачі, що вирішуватиме нова інформаційна система, полягають у створенні зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, а також у наданні повної та актуальної інформації про фільми, яка повинна бути легко доступною та зрозумілою.

Отже, розробка нової веб-орієнтованої інформаційної системи анонсів кінопродукції є актуальним завданням, що може вирішити проблеми, які існують у сучасних рішеннях.

#### 1.4 Постановка задачі проектування

На основі проаналізованих веб-ресурсів було поійнято рішення про створення веб-орієнтованої інформаційної системи анонсів кінопродукції, що забезпечуватиме користувачів зручною навігацією, повними та детальними описами фільмів, а також безкоштовним доступом до інформації.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Одним з кроків у створенні системи буде розробка бази даних, що міститиме інформацію про фільми, включаючи назву, жанр, рік випуску, режисера та акторський склад. Далі буде розроблений веб-інтерфейс системи, який забезпечить користувачам зручну навігацію та пошук фільмів за різними критеріями.

Окрему увагу буде приділено розробці алгоритмів, які забезпечать автоматичне оновлення бази даних інформацією про рейтинги фільмів, що виходять у прокат або уже вийшли. Крім того, буде розроблений функціонал для відгуків користувачів про фільми, що дозволить збирати та аналізувати зворотний зв'язок від глядачів.

Після завершення розробки системи планується проведення тестування, щоб переконатися у правильності роботи всіх функцій та унеможливити виникнення недоліків. Також планується забезпечити систему підтримкою та регулярним оновленням.

У підсумку, створення веб-орієнтованої інформаційної системи анонсів кінопродукції є важливим та актуальним завданням, вирішення якого задовільнить вимоги та потреби користувачів такого веб-ресурсу.

Інтерфейс веб-інформаційної системи розроблятиметься на мові програмування HTML та CSS з використанням фреймворку Django і Bootstrap, що забезпечить зручну навігацію та можливість швидкого пошуку необхідної інформації.

Для реалізації веб-орієнтованої інформаційної системи доцільно використати мову програмування Python та фреймворк Django, що забезпечить швидку та ефективну розробку веб-додатків. Також буде розроблена база даних для зберігання інформації про фільми та користувачів.

Використання Python та Django дозволить легко розробити зручний та інтуїтивно зрозумілий інтерфейс, а використання бази даних SQLite забезпечить швидкий доступ до необхідної інформації.

Основною метою розробки веб-ресурсу є створення зручної та легко доступної інформаційної системи для глядачів кінофільмів. Для досягнення цієї мети було вирішено реалізувати наступні функціональні можливості:

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

- можливість переглядати інформацію про кінофільми, таку як заголовок, опис, трейлер, жанр та оцінки користувачів;
- можливість додавати нові фільми до бази даних та редагувати існуючі записи;
- можливість оцінювати фільми та додавати коментарі до них;
- можливість шукати фільми за різними параметрами, такими як жанр, рік випуску, рейтинг та інші.

Для забезпечення високої продуктивності та швидкості роботи веб-додатку було доцільно використовувати базу даних що буде реалізована в SQLite. Також варто розглянуто можливість використання хешування та оптимізації запитів до бази даних.

З метою забезпечення безпеки та захисту від несанкціонованого доступу до інформації було доцільно використовувати систему авторизації та автентифікації користувачів з використанням технології JWT (JSON Web Tokens).

Отже, розроблятиметься веб-система, яка забезпечуватиме зручний та легкий доступ до інформації про кінофільми, а також надаватиме можливість додавати нові записи, оцінювати та коментувати фільми.

Метою дипломного проекту є створення веб-орієнтованої інформаційної системи анонсів кінопродукції.

Для досягнення мети необхідно виконати наступні завдання:

- провести аналіз предметної області;
- проаналізувати існуючі аналоги;
- розробити структуру веб-орієнтованої інформаційної системи;
- розробити алгоритмічне та інформаційне забезпечення веб-орієнтованої інформаційної системи;
- розробити інтерфейс інформаційної системи;
- реалізувати та протестувати веб-орієнтовану інформаційну систему.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

## 2 ЗАГАЛЬНА СТРУКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

### 2.1 Загальна структура проекрованої системи

Загальна структура веб-орієнтованої інформаційної системи анонсів кінопродукції включатиме наступні компоненти:

- стартова сторінка веб-орієнтованої інформаційної системи, зображення або банер з актуальні анонсами найновіших фільмів, посилання на найпопулярніші фільми та інші елементи, що привертають увагу користувачів;
- сторінка що надає повний список фільмів, розподілений за категоріями, жанрами чи іншими характеристиками.
- кожен фільм має свою окрему сторінку з детальною інформацією про нього, включаючи постер, опис сюжету, трейлери, рейтинги, коментарі користувачів тощо;
- сторінка яка дозволяє користувачам шукати фільми за назвою, акторами, режисерами, жанрами або іншими параметрами. Також можуть бути наявні фільтри для сортування результатів за рейтингом, даті реалізації, тривалості тощо;
- сторінка реєстрації нових користувачів і вхід в систему для зареєстрованих користувачів;
- сторінка для адміністраторів, де можна додавати та оновлювати інформацію про фільми, керувати користувачами, модерувати коментарі та підтримувати інші адміністративні функції;

Веб-орієнтована інформаційна система анонсів кінопродукції використовує три основні компоненти: веб-браузер, веб-сервер і базу даних, тому буде використовуватися клієнт-серверна архітектура.

Клієнт-серверну архітектуру можна означити, як концепцію інформаційної мережі, в якій основна частина її ресурсів знаходиться на серверах. Така архітектура визначає наступні три компоненти: набір серверів, які надають інформацію чи певні послуги програмам, які до них звертаються; набір клієнтів, які

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

використовують цю інформацію чи послуги, надані серверами; мережа, яка забезпечує взаємодію між клієнтами та серверами.

Клієнт-серверна архітектура є розподіленою архітектурою, в якій програма або система поділена на дві основні компоненти: клієнтів і серверів. Клієнти і сервери взаємодіють між собою, виконуючи різні ролі і надаючи різні послуги (рисунок 2.1).

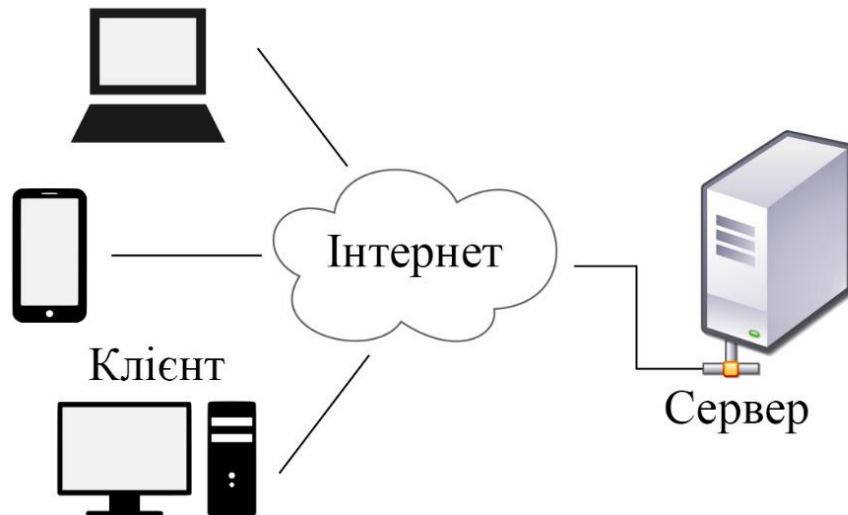


Рисунок 2.1 – Клієнт серверна архітектура

Основні концепції клієнт-серверної архітектури:

– клієнт - це програма або пристрій, який звертається до сервера для отримання певних послуг або ресурсів. Клієнти можуть бути різного типу, включаючи веб-браузери, мобільні додатки, настільні програми тощо;

– сервер - це програма або пристрій, який надає послуги або ресурси клієнтам. Сервери можуть бути спеціалізованими для надання конкретних послуг, таких як веб-сервери, бази даних, файлові сервери тощо;

– клієнти і сервери взаємодіють між собою за допомогою мережі, передаючи дані із запитам та відповідями. Це може бути здійснене через різні протоколи, такі як HTTP, TCP/IP, SOAP, REST тощо;

– клієнт-серверна архітектура дозволяє розподілення функцій між клієнтами і серверами. Сервери зазвичай виконують обробку даних, зберігання і доступ до

										ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата							22

ресурсів, тоді як клієнти забезпечують інтерфейс для користувача і взаємодіють з серверами для отримання необхідних послуг;

– клієнт-серверна архітектура дозволяє гнучко масштабувати систему, додавати нові клієнти або сервери за потреби. Це дозволяє розподілити навантаження і покращити продуктивність системи;

– клієнт-серверна архітектура дозволяє реалізувати механізми безпеки, такі як автентифікація і авторизація, для забезпечення захисту даних та ресурсів.

Клієнт-серверна архітектура широко використовується в розробці різноманітних додатків та систем, включаючи веб-додатки, мобільні додатки, ігрові сервери, хмарні сервіси та багато іншого. Вона дозволяє розподілити функції та забезпечити ефективну співпрацю між клієнтами і серверами.

## 2.2 Алгоритмічне забезпечення системи

Алгоритмічне забезпечення веб-орієнтованої інформаційної системи анонсів кінопродукції включає в себе ряд алгоритмів та функцій, що допомагають забезпечити ефективну роботу системи та зручну користувацьку взаємодію.

Основні алгоритмічні елементи можуть включати:

– розробка алгоритмів для ефективного пошуку та фільтрації фільмів залежно від різних критеріїв, таких як жанр, рік випуску, актори, режисери тощо. Це допоможе користувачам знайти фільми, що їх цікавлять, швидко і зручно;

– використання алгоритмів рекомендацій для пропозицій користувачам фільмів, які можуть їх зацікавити, на основі їхніх попередніх вподобань, переглядів або оцінок;

– розробка алгоритмів для обробки та відображення інформації про фільми, включаючи описи, зображення, трейлери, акторів, режисерів та інші відомості. Це може включати використання алгоритмів для форматування та структурування даних для зручного відображення на веб-сторінках;

– розробка алгоритмів для керування користувачами, включаючи реєстрацію та авторизацію. Також можна включити алгоритми стеження за

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

активністю користувачів, обробку коментарів та оцінок, ведення списків перегляду та інші функції, які забезпечують зручний користувацький досвід;

– розробка алгоритмів для оптимізації продуктивності системи, включаючи кешування даних, розподілення навантаження, оптимізацію запитів до бази даних та інші методи підвищення швидкодії та ефективності системи;

– розробка алгоритмів для інтеграції зовнішніх сервісів, таких як бази даних фільмів, API для отримання даних про фільми, платформи для онлайн-прокату або покупки квитків. Це дозволить системі отримувати актуальну інформацію та розширити свій функціонал.

Ці алгоритми та функції спільно працюють для забезпечення зручного, швидкого та функціонального веб-додатку для анонсів кінопродукції.

На рисунку 2.2 представлено схему алгоритму реєстрації користувача.

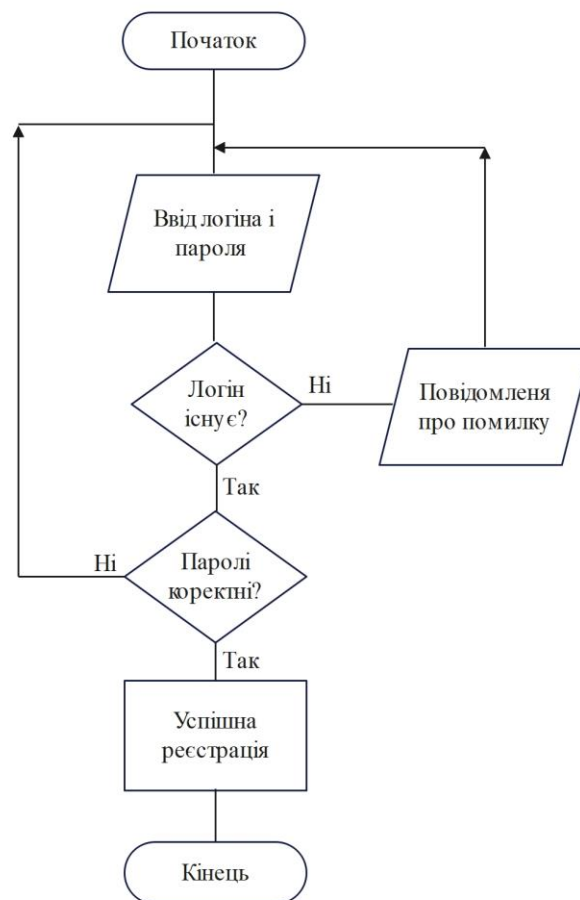


Рисунок 2.2 – Схема алгоритму реєстрації користувача



На рисунку 2.3 представлено схему алгоритму авторизації користувача.

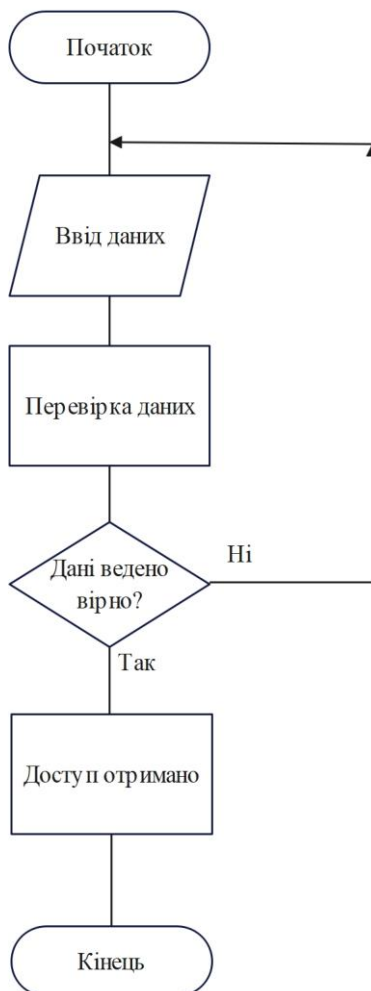


Рисунок 2.3 – Схема алгоритму авторизації користувача

Схема алгоритму роботи веб-орієнтованої інформаційної системи представлена в додатку А. Розроблено діаграму станів та переходів, що представлена в додатку Б. Вона показує яка поведінка буде у програмної системи, при отриманні керуючих дій. Під керуючими діями розуміється керуюча інформація, яку отримує система ззовні.

Для детальнішого представлення роботи проекрованої системи на рисунку 2.4 представлено діаграму діяльності. Вона зображається графом, вершинами якого є дії або види діяльності, а дугами – переходи від одного стану до іншого.

Головним призначенням використання діаграми діяльності є візуалізація особливостей реалізації операцій класів, коли потрібно зобразити алгоритм їх виконання.

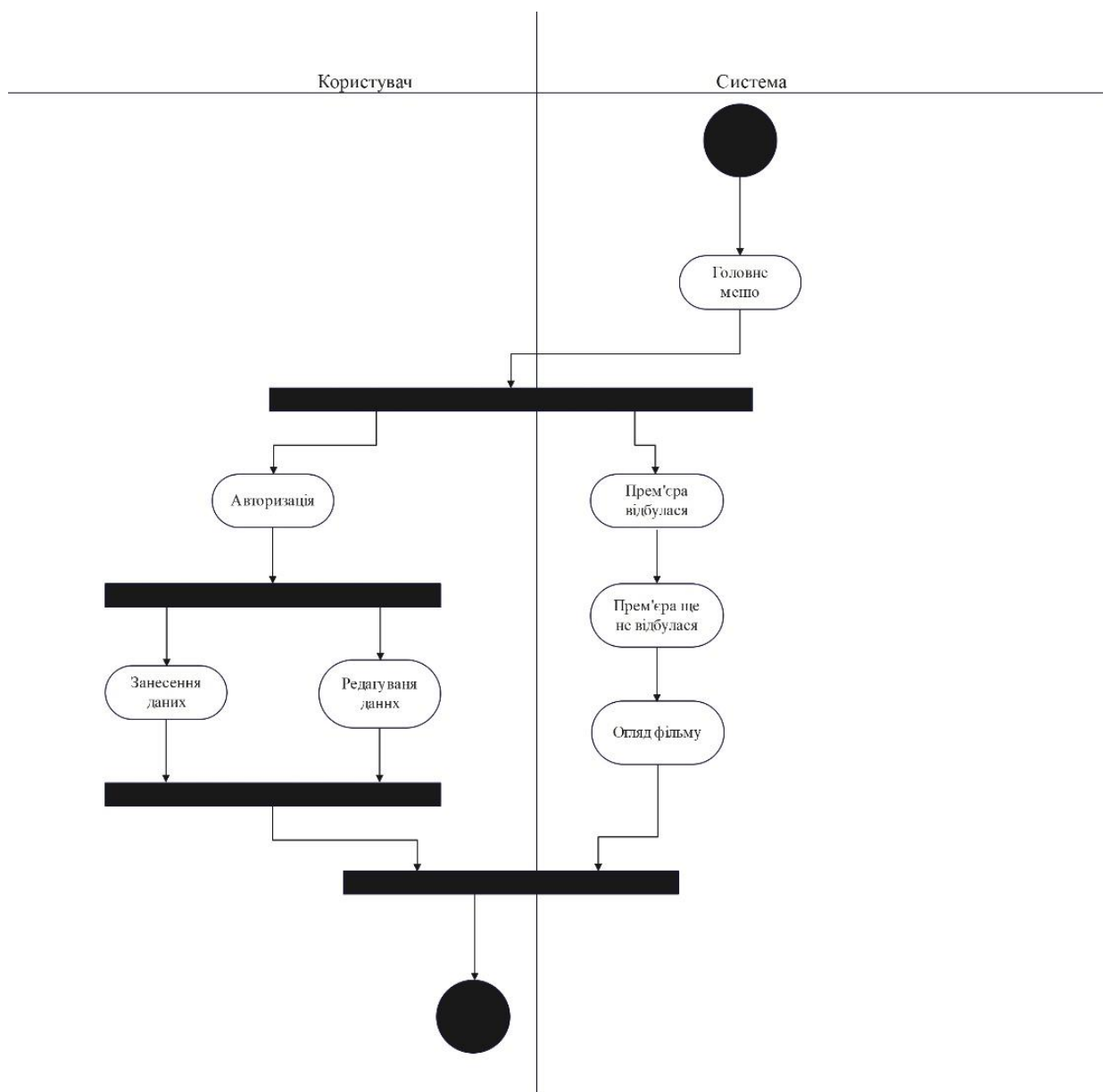


Рисунок 2.4 – Діаграма діяльності

Користувач спочатку авторизується в системі. Після авторизації він може занести дані про прем'єру фільму і також відредагувати дані, які вже були занесені.

Також бачимо, що якщо прем'єра фільму ще не відбулася, система відображає цей статус. Якщо прем'єра відбулася, користувач може перейти до огляду фільму. Весь цей процес відбувається через взаємодію користувача з системою.

## 2.3 Інформаційне забезпечення

Інформаційне забезпечення веб-орієнтованої інформаційної системи анонсів кінопродукції включає в себе процес збору, обробки, зберігання в базі даних інформацію про кінопродукцію і користувачів системи.

База даних для інформаційної системи буде створена засобами SQLite3, який є середовищем для створення вбудованого типу бази даних, і не вимагає окремого серверу баз даних. База даних буде зберігатися в одному файлі, що полегшує роботу з нею.

Особливості бази даних SQLite3:

- підтримує такі типи даних, як цілі числа (INTEGER), числа з рухомою комою (REAL), рядки (TEXT), бінарні дані (BLOB) та булеві значення (BOOLEAN). Тип даних NULL також є можливим;
- не потребує окремого серверу баз даних, а база даних зберігається в одному файлі. Це полегшує розгортання та управління базою даних;
- підтримує стандартну мову запитів SQL для створення, зчитування, оновлення та видалення даних (CRUD-операції);
- підтримує механізм транзакцій для забезпечення цілісності даних.
- дозволяє створювати індекси на стовпцях таблиць, що поліпшує швидкодію запитів;
- дозволяє створювати багато таблиць та встановлювати зв'язки між ними, такі як один до багатьох, багато до багатьох;
- має вбудовану підтримку резервного копіювання бази даних та відновлення збереженого стану.

SQLite3 є популярним вибором для додатків з невеликим обсягом даних або мобільних додатків. Вона має простий синтаксис та добре підходить для невеликих проектів з обмеженими потребами в шкалуванні та одночасному доступі до бази даних.

Основні аспекти інформаційного забезпечення:

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

- створення бази даних, розробку схеми даних, встановлення зв'язків між сутностями та інші процеси обробки даних;
  - забезпечення доступу до баз даних, АРІ та інших джерел, що містять інформацію про фільми, телесеріали, акторів, режисерів та інші відомості кінопродукції. Це можуть бути внутрішні бази даних, зовнішні постачальники даних або інші джерела, які забезпечують актуальну інформацію;
  - обробка інформації про кінопродукцію, її структурування та класифікація для забезпечення зручного пошуку, фільтрації та відображення за запитом користувачів;
  - розробка механізмів зберігання даних, які забезпечують надійність, швидкість та безпеку інформації. Це включає використання баз даних, файлових систем або інших технологій для зберігання інформації про кінопродукцію;
  - розробка інтерфейсу, який дозволяють користувачам здійснювати пошук, перегляд, фільтрацію та отримання детальної інформації про кінопродукцію. Це може включати розробку сторінок з анонсами, детальними описами фільмів, списками акторів та режисерів, трейлерами, оцінками користувачів та іншої інформації.
  - розробка інструментів та інтерфейсу для адміністраторів системи, які дозволяють управляти даними, додавати та оновлювати інформацію про кінопродукцію, керувати користувачами та їхніми правами доступу;
  - розробка заходів безпеки для захисту інформації про кінопродукцію, включаючи захист від несанкціонованого доступу, злому або втрати даних;
  - забезпечення можливості інтеграції зовнішніх сервісів, таких як платформи для онлайн-прокату або покупки квитків, соціальні медіа тощо. Це дозволяє користувачам здійснювати додаткові дії або отримувати додаткову інформацію через інші сервіси.
- Всі ці аспекти інформаційного забезпечення спільно працюють для створення функціональної, зручної та надійної Веб-орієнтованої інформаційної системи анонсів кінопродукції.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

У Django база даних для користувачів генерується автоматично при використанні модуля `django.contrib.auth`, який надає вбудовану систему автентифікації та авторизації. Для створення бази даних для користувачів у Django, необхідно виконати наступні кроки:

- створити новий проект Django або використати вже існуючий [18];
- у файлі `settings.py` проекту встановити параметри підключення до бази даних, такі як тип бази даних, адреса, ім'я користувача, пароль та інші налаштування, якщо потрібно [18];
- у файлі `settings.py` потрібно включити модуль `django.contrib.auth` у списку `INSTALLED_APPS`. Це дозволить використовувати функціональність, пов'язану з автентифікацією та авторизацією [18];
- виконати міграції командою `python manage.py makemigrations`, щоб згенерувати міграційні файли на основі визначених моделей користувачів. Потім виконайте команду `python manage.py migrate`, щоб застосувати міграції та створити таблиці бази даних для користувачів, а також інші таблиці, які можуть знадобитися [18];
- запустити команду `python manage.py createsuperuser` для створення супер користувача, який матиме повний доступ до адміністративного інтерфейсу та можливість керування користувачами [18].

Після виконання цих кроків база даних для користувачів буде готова. Django автоматично створює необхідні таблиці для збереження даних про користувачів, групи, дозволи та іншу інформацію. Також можна використовувати вбудовані класи та методи Django для роботи з користувачами, такі як `User` для представлення користувача, `AuthenticationForm` для автентифікації та інші.

Аналогічним способом створюються інші таблиці, для цього потрібно в файлі `models.py` описати які потрібні таблиці. Розглянемо таблицю «Фільми» в базі даних системи. На рисунку 2.5 бачимо опис таблиці «Фільми» за допомогою коду Python.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

```

class Films(models.Model):
    img = models.FileField('зображення', upload_to='mysite/main/static/main/img')
    title = models.CharField('Назва фільму', max_length=50)

    title_en = models.CharField('Назва фільму en', max_length=50)
    anons = models.CharField('Опис', max_length=250)
    ful_text = models.TextField('Повний опис')
    slug = models.SlugField(default='default-slug', unique=True, max_length=50)
    imdb_rating = models.DecimalField('Рейтинг IMDb', max_digits=3, decimal_places=1, null=True, blank=True)
    imdb_id = models.CharField('IMDB id', max_length=20, null=True, blank=True)
    premiere_date = models.DateField('Дата прем'єри')
    date = models.DateTimeField('Дата публікації')

```

Рисунок 2.5 – Опис таблиці «Фільми»

У цьому коді визначається модель Films (сутність), яка є підкласом models.Model в Django. Кожне поле моделі відображає стовпець в базі даних, яку Django створить під час міграцій.

Основні поля (атрибути), які використовуються в цій моделі:

- img – поле FileField, яке представляє зображення фільму, використовується для завантаження файлу зображення у вказану директорію;
- title – поле CharField, яке містить назву фільму, вказується максимальна довжина поля (50 символів);
- title\_en – поле CharField, яке містить назву фільму англійською мовою;
- anons – поле CharField, яке містить опис фільму, вказується максимальна довжина поля (250 символів);
- ful\_text – поле TextField, яке містить повний опис фільму без обмежень довжини;
- slug – поле SlugField, яке використовується для створення унікальної URL-адреси фільму на основі його назви, значення поля автоматично створюється на основі назви фільму, що міститься в title\_en;
- imdb\_rating – поле DecimalField, яке містить рейтинг фільму в IMDb (Internet Movie Database), вказується кількість цифр до десяткової коми (три) та кількість цифр після десяткової коми (одну);
- imdb\_id – поле CharField, яке містить ідентифікатор фільму в IMDb;
- premiere\_date – поле DateField, яке містить дату прем'єри фільму;
- date – поле DateTimeField, яке містить дату та час публікації фільму.

										ДП.КН.9702718.053.ПЗ	Арк.
											30
Змн.	Арк.	№ докум.	Підпис	Дата							

Модель має визначення, задане класом Meta, в якому можна настроїти її властивості, такі як назви для адміністративного інтерфейсу Django. У даному випадку використовується назви "Фільм" та "Фільми" для відображення моделі в адміністративному інтерфейсі Django.

Функція `__str__` повертає рядкове представлення об'єкту моделі (сутності). У випадку виклику функції `str()` моделі «Films» буде повернена назва фільму.

Крім моделі «Фільми», що вже розглянуто вище, є ще три сутності (моделі) Series, Season і Episode, отож глянемо на їх структуру таблиць у базі даних Django.

Модель Series представляє серіал і містить поля, які відображаються у таблиці. Вона має поля `img` (зображення), `title` (назва серії) і `description` (опис серії). Функція `__str__` визначена для зручного відображення об'єктів цієї моделі у рядковому форматі. Клас Meta використовується для налаштування відображення моделі в адміністративному інтерфейсі Django, рисунок 2.6.

```
class Series(models.Model):
    img = models.FileField('зображення', upload_to='mysite/main/static/main/img')
    title = models.CharField('Назва серії', max_length=255)
    description = models.TextField('Опис серії')

    def __str__(self):
        return self.title

    class Meta:
        verbose_name = 'Серіал'
        verbose_name_plural = 'Серіали'
```

Рисунок 2.6 – Опис таблиці «Серіали»

Модель Season представляє сезон серіалу і містить поля, що описують номер сезону (`season_number`), кількість серій (`episode_count`), опис сезону (`description`) та зв'язок з моделлю Series за допомогою зовнішнього ключа (`ForeignKey`). Зовнішній ключ `series` вказує на зв'язок з моделлю Series, і параметр `related_name` задає зворотний зв'язок для доступу до сезонів з об'єкту серіалу.

Функція `__str__` повертає рядок, який відображає номер сезону та назву серіалу (рисунок 2.7).

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

```

class Season(models.Model):
    series = models.ForeignKey(Series, on_delete=models.CASCADE, related_name='seasons')
    season_number = models.PositiveIntegerField('Номер сезону')
    episode_count = models.PositiveIntegerField('Кількість серій')
    description = models.TextField('Опис сезону', blank=True)

    def __str__(self):
        return f"Season {self.season_number} of {self.series.title}"

```

Рисунок 2.7 – Опис таблиці «Сезони»

Модель Episode представляє серію серіалу і містить поля для номера серії (episode\_number), назви серії (title) і зв'язку з моделлю Season за допомогою зовнішнього ключа. Зовнішній ключ season вказує на зв'язок з моделлю Season, і параметр related\_name задає зворотний зв'язок для доступу до серій з об'єкту сезону. Функція \_\_str\_\_ повертає рядок, який відображає назву серіалу, номер сезону та номер серії (рисунок 2.8).

```

class Episode(models.Model):
    season = models.ForeignKey(Season, on_delete=models.CASCADE, related_name='episodes')
    episode_number = models.PositiveIntegerField('Номер серії')
    title = models.CharField('Назва серії', max_length=255)

    def __str__(self):
        return f"{self.season.series.title} - Season {self.season.season_number}, Episode {self.episode_number}"

```

Рисунок 2.8 – Опис таблиці «Серії»

Ці моделі описують структуру таблиць у базі даних і визначають зв'язки між ними. Після виконання міграцій Django створить відповідні таблиці у базі даних на основі цих моделей.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32



## 3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1 Програмна реалізація

Перед початком розробки веб-орієнтованої інформаційної системи на Python, потрібно виконати декілька підготовчих кроків, а саме установити редактори коду:

– Visual Studio Code (VS Code) – це легкий і потужний редактор коду, який підтримує багато мов програмування та надає розширення для поліпшення розробки веб-додатків;

– PyCharm – це інтегроване середовище розробки (IDE) для мови програмування Python, яке надає розширені можливості для розробки веб-додатків.

Необхідно завантажити та налаштувати мови програмування:

– Python – це високорівнева мова програмування, яка має простий синтаксис та велику кількість бібліотек для розробки веб-додатків [13];

– JavaScript (JS) – мова програмування, яка використовується для створення інтерактивних елементів на веб-сторінках і взаємодії з користувачем [12];

– HTML (HyperText Markup Language) – мова розмітки, яка використовується для створення структури веб-сторінок [10];

– CSS (Cascading Style Sheets) – мова опису стилів, яка використовується для задання зовнішнього вигляду веб-сторінок [10].

Необхідно завантажити потрібні фреймворки та стилі:

– Django – високорівневий веб-фреймворк на основі Python, який дозволяє швидко розробляти потужні веб-додатки з використанням моделей бази даних, URL-маршрутизації, автентифікації та інших функцій [18];

– Bootstrap – популярний фреймворк для розробки веб-додатків з використанням HTML, CSS та JavaScript. Він надає набір стилів, компонентів та інструментів для швидкого створення сучасного та адаптивного інтерфейсу користувача [19].

Наступним кроком є встановлення необхідного оточення розробки Python. Це включає встановлення Python, налаштування віртуального середовища і установку необхідних пакетів і залежностей.

										ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата							33

Після кроків, описаних вище, було розроблено функціональні компоненти веб-сайту, такі як сторінки, форми, автентифікація користувачів, робота з базою даних тощо, розподілено функціональність на окремі модулі або файли для зручності управління і підтримки коду.

Розглянемо реалізацію деяких функції з прикладами коду. Функція переключання між вкладками, яка зустрічається майже на всіх сторінках.

Зразок коду перемикачів між вкладками подано нижче:

```
function showTable(tableId) {
  var table1 = document.getElementById("table1");
  var table2 = document.getElementById("table2");

  if (tableId === "table1") {
    table1.style.display = "table";
    table2.style.display = "none";
  } else if (tableId === "table2") {
    table1.style.display = "none";
    table2.style.display = "table";
  }
}
```

Також в даній таблиці присутня фільтрація фільмів, тобто якщо прем'єра фільму відбулася, то в таблиці «Скоро» фільм не буде відображатися, а буде відображатися у таблиці «Недавно». У випадку, якщо прем'єри ще не було, то фільм буде відображатися в таблиці «Скоро», також таблиці обмежені в записях, тобто не буде виводитися більше п'яти записів з бази даних. Виведення списку в HTML розмітці:

```
{% for el in Passed_films|slice:":5" %}
  <tr>
    <th scope="row">1</th>
    <td></td>
    <td><a href="">{{el.title}}</a><p>{{el.title_en}}</p></td>
    <td>{{el.anons}}</td>
    <td>{{el.premiere_date|date:"d.m.Y"}}</td>
  </tr>
{% endfor %}
```

Виведення списку та фільтрування Python:

```
def index(request):
  dt = datetime.now()
  data = dt.strftime("%d.%m.%Y")
```

									ДП.КН.9702718.053.ПЗ	Арк.
										34
Змн.	Арк.	№ докум.	Підпис	Дата						

```

expected_films = Articles.objects.filter(premiere_date__gte=dt)
Passed_films = Articles.objects.filter(premiere_date__lt=dt)
return render(request, 'main/index.html', {'expected_films': expected_films, 'Passed_films': Passed_films, 'data':
data})

```

Переходимо до реєстрації та автентифікації, тут є дві кнопки, працюють ці кнопки так само, як вище описано перемикання між вкладками «Скоро» та «Недавно». Нижче наведено код, який відповідає за реєстрацію і вхід користувачів.

Реєстрація і автентифікація користувача на Python реалізована кодом:

```

def login_reg(request):
    if request.method == 'POST':
        # Отримання даних форми входу
        login_form = AuthenticationForm(request, data=request.POST)
        # Отримання даних форми реєстрації
        register_form = UserCreationForm(request.POST)

        if login_form.is_valid():
            # Вхід в систему
            user = login_form.get_user()
            login(request, user)
            return redirect('about')

        if register_form.is_valid():
            # Реєстрація нового користувача
            register_form.save()
            username = register_form.cleaned_data.get('username')
            password = register_form.cleaned_data.get('password1')
            user = authenticate(username=username, password=password)
            login(request, user)
            return redirect('home')
        else:
            login_form = AuthenticationForm()
            register_form = UserCreationForm()

    return render(request, 'main/login_reg.html', {'login_form': login_form, 'register_form': register_form})

```

Підчас реєстрації необхідно ввести необхідні дані: логін та двічі ввести пароль.

Код реєстрації на HTML розмітці:

```

<form method="post" action="{% url 'login_reg' %}">
    {% csrf_token %}
    <div class="form-floating mb-3">
        <label for="{{ register_form.username.id_for_label }}" class="form-label">Ім'я користувача</label>
        {{ register_form.username }}
    </div>
    <div class="form-floating mb-3">
        <label for="{{ register_form.password1.id_for_label }}" class="form-label">Пароль</label>
        {{ register_form.password1 }}
    </div>

```

										ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата							35

```

<div class="form-floating mb-3">
  <label for="{{ register_form.password2.id_for_label }}" class="form-label">Підтвердження пароля</label>
  {{ register_form.password2 }}
</div>
<div class="form-floating mb-3">
  <button class="btn btn-dark" type="submit">Реєстрація</button>
</div>
</form>

```

Також код HTML розмітки для автєфікації користувача:

```

<form method="post" action="{% url 'login_reg' %}">
  {% csrf_token %}
  <div class="form-floating mb-3">
    <label for="{{ login_form.username.id_for_label }}" class="form-label">Ім'я користувача</label>
    {{ login_form.username }}
  </div>
  <div class="form-floating mb-3">
    <label for="{{ login_form.password.id_for_label }}" class="form-label">Пароль</label>
    {{ login_form.password }}
  </div>
  <div class="form-floating mb-3">
    <button class="btn btn-dark" type="submit">Увійти</button>
  </div>
</form>

```

Розглянемо функції, які є на панелі користувача. Є форма добавлення нових записів, також, крім форми добавлення постів, є кнопки для перемикання між формами, яку вже розглядали.

Код форми добавлення записів на HTML:

```

<form action="" method="post" enctype="multipart/form-data">
  {% csrf_token %}
  <div class="form-floating mb-3">
    <input type="file" class="form-control" name="img" id="img" value="Добавте картинку">
  </div>
  <div class="form-floating mb-3">
    <input type="text" class="form-control" name="title" id="title" value="Назва">
  </div>
  <div class="form-floating mb-3">
    <input type="text" class="form-control" name="title_en" id="title_en" value="Англійська Назва">
  </div>
  <div class="form-floating mb-3">
    <input type="text" class="form-control" name="anons" id="anons" value="Опис">
  </div>
  <div class="form-floating mb-3">
    <input type="text" class="form-control" name="ful_text" id="ful_text" value="Повний опис">
  </div>
  <div class="form-floating mb-3">
    <input type="date" class="form-control" name="premiere_date" id="premiere_date">
  </div>
  <div class="form-floating mb-3">
    <input class="btn btn-dark" type="submit" value="Публікувати">
  </div>
</form>

```

					ДП.КН.9702718.053.ПЗ		Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			36

```
</div>  
</form>
```

Також код для обробки цих даних на Python:

```
def about(request):  
    if request.method == 'POST':  
        image = request.FILES['img']  
        title = request.POST['title']  
        english_title = request.POST['title_en']  
        short_description = request.POST['anons']  
        full_description = request.POST['ful_text']  
        release_date = request.POST['premiere_date']  
        movie = Artiles(img=image, title=title, title_en=english_title, anons=short_description,  
                        ful_text=full_description, premiere_date=release_date)  
        movie.save()  
        update_imdb_id()  
        update_imdb_ratings()  
        return redirect('about')  
    return render(request, 'main/about.html')
```

Отож, вже виконано реалізацію функціоналу веб-орієнтованої інформаційної системи кінопродукції. Була ретельно розглянута архітектура системи, і для розробки вибрано фреймворк Django.

Під час реалізації були створені моделі для бази даних, зокрема моделі фільмів, користувачів та інших сутностей. За допомогою міграцій, були створені необхідні таблиці бази даних з правильною структурою(див підрозділ 2.3).

Також розроблена можливість для додавання, редагування та видалення фільмів та інших об'єктів. Для цього були використані Django-форми та відповідні шаблони, що забезпечують зручний інтерфейс для користувачів.

Реалізовано опції автентифікації та авторизації користувачів за допомогою модуля `django.contrib.auth`. Користувачі мають змогу зареєструватися і увійшовши увійти в систему і мати доступ до певних функцій в залежності від їх ролі.

У додатку В представлено основний код програми обробки сторінок.

Наступними кроками є написання тестів для перевірки працездатності і коректності веб-сайту (підрозділ 3.2). Доцільно використовувати модулі тестування, які надаються фреймворком або відповідними бібліотеками.

Також важливо перевірити продуктивність веб-орієнтовної інформаційної системи і виконати оптимізаційні заходи, якщо необхідно. Забезпечити безпеку

									ДП.КН.9702718.053.ПЗ	Арк.
										37
Змн.	Арк.	№ докум.	Підпис	Дата						

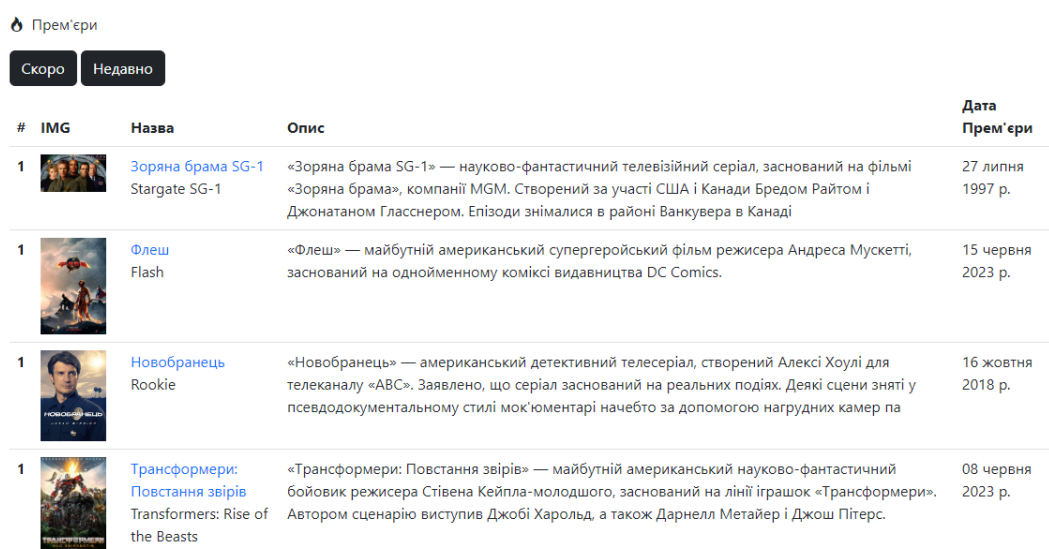
веб-орієнтованої системи потрібно, використовуючи належні практики, такі як захист від атак, обробка валідації даних тощо.

Після завершення розробки веб-орієнтованої системи буде здійснено її розгортання. Це також включає налаштування сервера, конфігурацію середовища, налаштування бази даних.

### 3.2 Інтерфейс користувача

Для розробки інтерфейсу веб-орієнтованої системи анонсів кінопродукції використовувалися редактор коду Visual Studio Code, HTML для створення структури сторінок, CSS для стилізації елементів та встановлення зовнішнього вигляду, а також використовувався фреймворк Bootstrap для швидкого створення адаптивного та сучасного дизайну. Ці інструменти допомогли ефективно розробляти інтерфейс.

Панель «Прем'єра» складається з двох таблиць «Скоро» та «Недавно», в таблиці «Скоро» міститься інформація про фільми і серіали, які ще не відбулися (рисунок 3.1).



Прем'єри

Скоро    Недавно





#	IMG	Назва	Опис	Дата Прем'єри
1		<a href="#">Зоряна брама SG-1</a> Stargate SG-1	«Зоряна брама SG-1» — науково-фантастичний телевізійний серіал, заснований на фільмі «Зоряна брама», компанії MGM. Створений за участі США і Канади Бредом Райтом і Джонатаном Гласснером. Епізоди знімалися в районі Ванкувера в Канаді	27 липня 1997 р.
1		<a href="#">Флеш</a> Flash	«Флеш» — майбутній американський супергеройський фільм режисера Андреаса Мускетті, заснований на однойменному коміксі видавництва DC Comics.	15 червня 2023 р.
1		<a href="#">Новобранець</a> Rookie	«Новобранець» — американський детективний телесеріал, створений Алексі Хоулі для телеканалу «ABC». Заявлено, що серіал заснований на реальних подіях. Деякі сцени зняті у псевдодокументальному стилі мок'юментарі начебто за допомогою нагрудних камер па	16 жовтня 2018 р.
1		<a href="#">Трансформери: Повстання звірів</a> Transformers: Rise of the Beasts	«Трансформери: Повстання звірів» — майбутній американський науково-фантастичний бойовик режисера Стівена Кейпла-молодшого, заснований на лінії іграшок «Трансформери». Автором сценарію виступив Джеб Харольд, а також Дарнелл Метайер і Джош Пітерс.	08 червня 2023 р.

Рисунок 3.1 – Перемикання між вкладками

Таблиця «Недавно» показує серіали та фільми, прем'єра яких вже відбулася (рисунок 3.2).


#	First	Last	Handle
1	 Зоряна брама SG-1	Stargate SG-1	«Зоряна брама SG-1» — науково-фантастичний телевізійний серіал, заснований на фільмі «Зоряна брама», компанії MGM. Створений за участі США і Канади Бредом Райтом і Джонатаном Гласснером. Епізоди знімалися в районі Ванкувера в Канаді

Рисунок 3.2 – Таблиця «Недавно»

Розглянемо тепер інтерфейс входу і реєстрації на сайті, на рисунку 3.3 зображено форму входу на сайт, що містить дві кнопки, які перемикають користувача між формами реєстрації та входу.

Реєстрація Вхід

Логін

Пароль

Увійти

Рисунок 3.3 – Форма входу

На рисунку 3.4 зображено форму реєстрації, потрібно вказати більше даних для реєстрації.

Реєстрація Вхід

Email адрес

Логін

Пароль

Повтор паролю

Реєстрація

Рисунок 3.4 – Форма реєстрації

Панель користувача містить форму додавання записів в базу даних, а також форму редагування постів (рисунок 3.5).

[Добавити пост](#)

[Редагування постів](#)

[Плей лист](#)

[Налаштування](#)

Вибір файлу Файл не вибрано

Назва

Англійська назва

Опис

Повний опис

Дата примери

дд.мм.рррр --:--

Публікувати

Рисунок 3.5 – Панель користувача

Таким чином, система має зручний та зрозумілий інтерфейс користувача.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40



### 3.3 Тестування програмного забезпечення

Тестування веб-орієнтованої інформаційної системи анонсів кінопродукції, включає проведення різних видів тестів для перевірки його функціональності, коректності роботи та відповідності вимогам. Нижче розглянемо кілька типів тестів, які були проведені:

- виконання тестових сценаріїв для перевірки правильності роботи основних функцій веб-ресурсу, таких як додавання анонсу, перегляд анонсів, пошук за критеріями, сортування тощо;
- перевірка зручності взаємодії користувача з веб-ресурсу, перевірка коректності відображення інформації, перевірка відповідності дизайну та вимогам щодо вигляду та поведінки;
- перевірка точності пошуку та правильності застосування фільтрів для знаходження анонсів за певними критеріями, такими як жанр, рік випуску, рейтинг тощо;
- перевірка реакції системи на некоректні дані або ситуації, такі як некоректний формат даних, помилкові запити, недоступні ресурси тощо. Перевірка, забезпечення коректної обробки та відображення помилок для користувача;
- перевірка швидкості завантаження сторінок, відповідного реагування на запити користувачів та можливості масштабування сайту при великій кількості користувачів або великому обсязі даних;
- перевірка відображення та роботи веб-орієнтованої інформаційної системи на різних браузерях, операційних системах та пристроях, щоб забезпечити її сумісність різними платформами.

Тестування відображення веб-системи на різних браузерях приведено в таблиці 3.1.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Таблиця 3.1 – Тестування на різних браузерях

Опис	Кроки	Результат
Відкриття сайту в Google Chrome	1.Відкриття браузера Google Chrome 2.Ведення URL адресу 3.Натиснути Enter	Сайт успішно відкривається і коректно відображається на Google Chrome
Відкриття сайту в Mozilla Firefox	1.Відкриття браузера Mozilla Firefox 2.Ведення URL адресу 3.Натиснути Enter	Сайт успішно відкривається і коректно відображається на Mozilla Firefox
Відкриття сайту в Microsoft Edge	1.Відкриття браузера Microsoft Edge 2.Ведення URL адресу 3.Натиснути Enter	Сайт успішно відкривається і коректно відображається на Microsoft Edge
Відкриття сайту в Opera	1.Відкриття браузера Opera 2.Ведення URL адресу 3.Натиснути Enter	Сайт успішно відкривається і коректно відображається на Opera
Відкриття сайту в Safari	1.Відкриття браузера Safari 2.Ведення URL адресу 3.Натиснути Enter	Сайт успішно відкривається і коректно відображається на Safari

При тестуванні веб-ресурсу анонсів кінопродукції також важливо враховувати потенційні проблеми безпеки, такі як захист від SQL-ін'єкцій, XSS

						ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			42

атак, автентифікації та авторизації користувачів. Застосування різних видів тестування допомагає виявити та виправити можливі проблеми, забезпечити якісну та надійну роботу веб-орієнтовна інформаційна система анонсів кінопродукції для користувачів.

Розглянемо як у Django працює шифрування паролів. У Django шифрування паролів здійснюється за допомогою алгоритму хешування та збереження хешованого значення в базі даних. Основний механізм шифрування паролів в Django називається "хешування з сіллю" (salted hashing). Нижче наведено опис процесу шифрування паролів в Django:

- створення хешу пароля це - коли користувач реєструється або змінює свій пароль, Django генерує випадкову "сіль" (salt) - це випадкова послідовність символів, яка додається до паролю [18];

- пароль разом зі сіллю хешується за допомогою сильного хеш-алгоритму, такого як bcrypt, PBKDF2 або Argon2. Хеш - це незворотне перетворення паролю та солі у вигляді випадкової послідовності символів фіксованої довжини [18];

- отриманий хеш пароля разом зі "сіллю" зберігається в полі бази даних, призначеному для зберігання хешованих паролів [18];

- при наступній автентифікації користувача введений пароль хешується з використанням тієї ж "солі", яка була використана під час створення хешу. Отриманий хеш порівнюється зі збереженим хешем в базі даних якщо вони співпадають, то пароль вважається вірним [18].

Цей підхід забезпечує безпеку паролів, оскільки навіть при доступі до бази даних зломиснику важко відновити початковий пароль з хешу. Крім того, кожен пароль має свою унікальну "сіль", що унеможливорює використання попередньо згенерованих розшифрованих хешів для вторгнення.

Django забезпечує цей механізм шифрування паролів автоматично через свій вбудований механізм автентифікації та управління користувачами. В системі було використано механізм що застосовує клас User та функції, такі як set\_password() для створення та перевірки паролів.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

З метою тестування реєстрації нового користувача було введено некоректний пароль а саме, був меншим, ніж вісім символів, система видала повідомлення про помилку, після введення коректних даних користувача було зареєстровано і добавлено в базу даних (рисунки 3.6, 3,7)

Ім'я користувача

Пароль

Підтвердження пароля

Рисунок 3.6 - Реєстрація

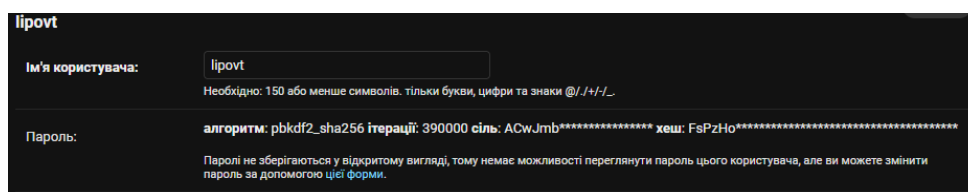


Рисунок 3.7 – Користувач в базі даних

Також було протестована адаптивність веб-орієнтованої інформаційної системи на різних пристроях. Завдяки тому, що використовується стилі Bootstrap, користувач може зайти на веб-платформу з різних пристроїв (рисунки 3.8, 3.9)

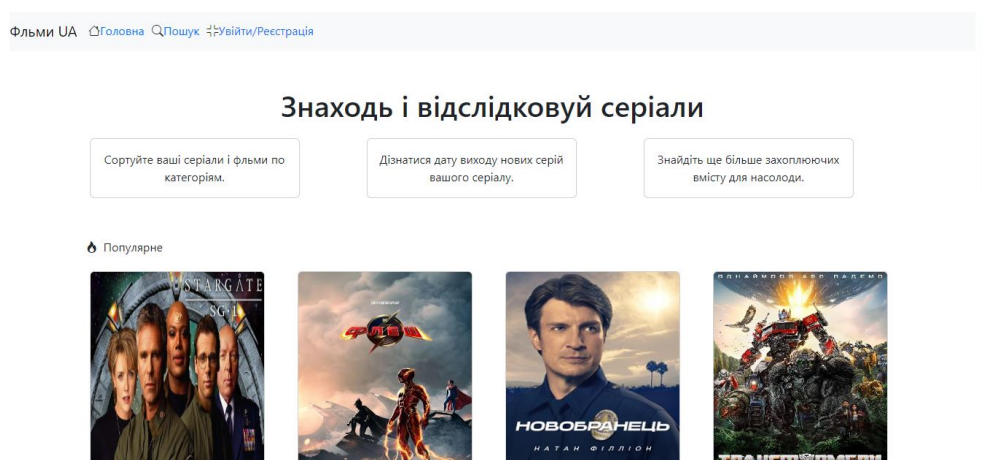


Рисунок 3.8 – Відображення головної сторінки веб-орієнтованої системи на екрані комп'ютера

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

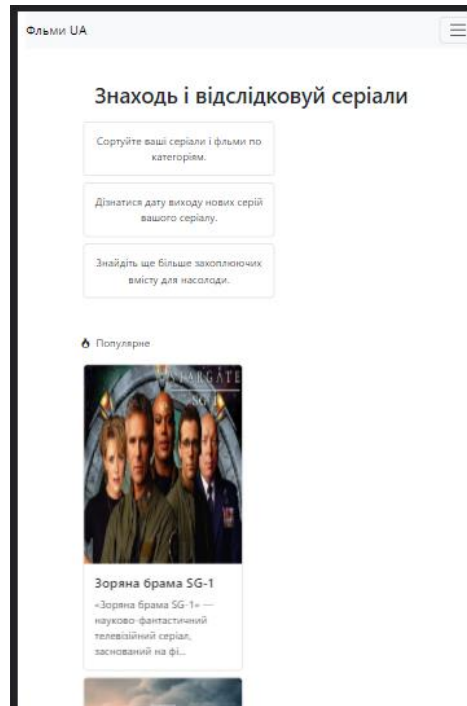


Рисунок 3.9 – Відображення головної сторінки веб-орієнтованої системи на екрані мобільного пристрою

Також було протестовано коректність занесення даних в базу даних. Для цього розроблено форму і заповнено відповідними даними (рисунок 3.10).

Вибрати файл Iron\_Man\_3.jpg

Залізна Людина 3

Iron Man 3

«Залізна людина 3» — художній фільм режисера Шейна Блека, заснований на одной

«Залізна людина 3» (англ. Iron Man 3) — художній фільм режисера Шейна Блека, зас

03.05.2013 18:58 🗑️

Публікувати

Рисунок 3.10 – Форма введення даних про фільм

					ДП.КН.9702718.053.ПЗ	Арк. 45
Змн.	Арк.	№ докум.	Підпис	Дата		

Тепер перевіряємо чи дані з'явилися в базі даних та на головній сторінці, рисунки 3.11 та 3.12 демонструють, що дані були введені і відображаються коректно.

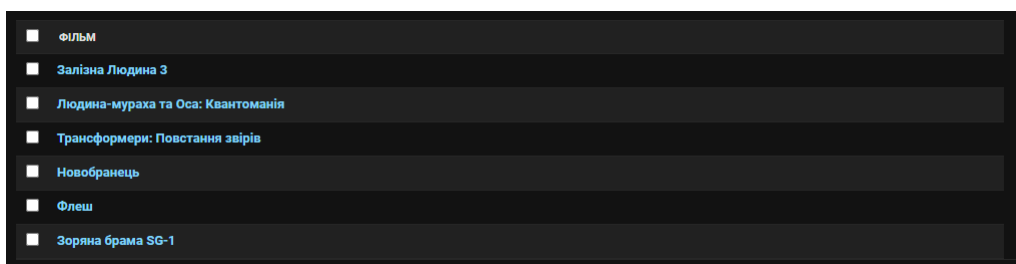


Рисунок 3.11 – Таблиця “Фільми”



Рисунок 3.12 – Відображення на головній сторінці

Під час тестування було виявлено дві помилки, пов'язані зі занесенням даних в базу даних. Перша помилка пов'язана з правильним збереженням зображень. Друга помилка виявився у форматі запису дати, коли система некоректно обробляла введену інформацію.

Виявлені помилки було виправлено. А саме були зроблені необхідні зміни в програмному кодї, щоб забезпечити правильне занесення зображень в базу даних та відповідну обробку формату дати.

Після внесених змін помилки були виправлені, і система працює коректно. Основні функції веб-орієнтованої інформаційної системи було протестовано і виявлено що вони відповідають вимогам.

## ВИСНОВКИ

Під час виконання дипломного проекту було:

1) проведено аналіз предметної області, що дало змогу отримати необхідні знання про кінопродукцію та існуючі системи анонсів фільмів;

2) проаналізовано існуючі аналоги і досліджено їх переваги та недоліки, Цей аналіз дав змогу сформуванню основні вимоги до веб-орієнтованої інформаційної системи анонсів кінопродукції;

3) описано структуру веб-орієнтованої інформаційної системи кінопродукції, яка включає модулі, компоненти та взаємодію між ними;

4) розроблено алгоритмічне забезпечення веб-орієнтованої інформаційної системи, діаграму діяльності, діаграму станів та переходів;

5) спроектовано та реалізовано базу даних в середовищі SQLite, в якій зберігається інформація про користувачів та фільми;

6) реалізовано основний функціонал веб-орієнтованої інформаційної системи на мові програмування Python за допомогою фреймворку Django;

7) розроблено інтерфейс веб-орієнтованої системи за допомогою технологій Bootstrap, HTML та CSS;

8) проведено тестування та налагодження веб-орієнтованої інформаційної системи для перевірки правильності роботи функціоналу. Виявлені помилки були виправлені.

Оскільки усі функції відповідають поставленим вимогам, система готова до використання. Розроблена веб-орієнтована інформаційна система кінопродукції може використовуватися кінотеатрами та кіностудіями для анонсування їх продукції.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-сайт. URL: <https://uk.wikipedia.org/wiki/Вебсайт>
2. Internet Movie Database IMDb: Ratings, Reviews, and Where to Watch the Best Movies & TV Shows. URL: <https://www.imdb.com/>
3. Rotten Tomatoes: Movies TV Shows, Movie Trailers, Reviews - Rotten Tomatoes. URL: <https://www.rottentomatoes.com/>
4. Letterboxd Social film discovery. URL: <https://letterboxd.com/>
5. Бородкіна І.Л., Бородкін Г.О. Інженерія програмного забезпечення: Посібник для студентів вищих навчальних закладів. Київ: Центр учбової літератури, 2020. 204 с.
6. Н. Мельник, Є. Левус. Вступ до інженерії програмного забезпечення. Львівська Політехніка, 2018. 248 с.
7. Sommerville I. Software Engineering Addison-Wesley, 2018. 773 p.
8. Олег Б. Web-програмування на базі ASP.NET. Розробка веб-додатків. Видавництво "Книжковий Клуб Клуб Сімейного Дозвілля", 2020. 400 с.
9. Jon Duckett. HTML and CSS: Design and Build Websites. Wiley, 2011. 512 p.
10. Brad Traversy. Modern HTML & CSS From The Beginning (Including Sass). UdeMy, 2020. 564 p.
11. Marijn Haverbeke. Eloquent JavaScript. No Starch Press, 2018. 472 p.
12. Манжос С. JavaScript: Навчальний посібник. Книжковий Клуб Клуб Сімейного Дозвілля, 2019. 352 с.
13. Олексій В. Програмування мовою Python. Навчальна книга – Богдан, 2019. 504 с.
14. Ерік М. Пришвидшений курс Python. Практичний, проєктно-орієнтований вступ до програмування. Видавництво Старого Лева, 2021. 600 с.
15. Robbins P. Python Programming for Beginners. Independently published, 2023. 114 p.
16. Barry P. Head First Python: A Brain-Friendly Guide Paperback. O'Reilly Media, 2016. 584 p.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

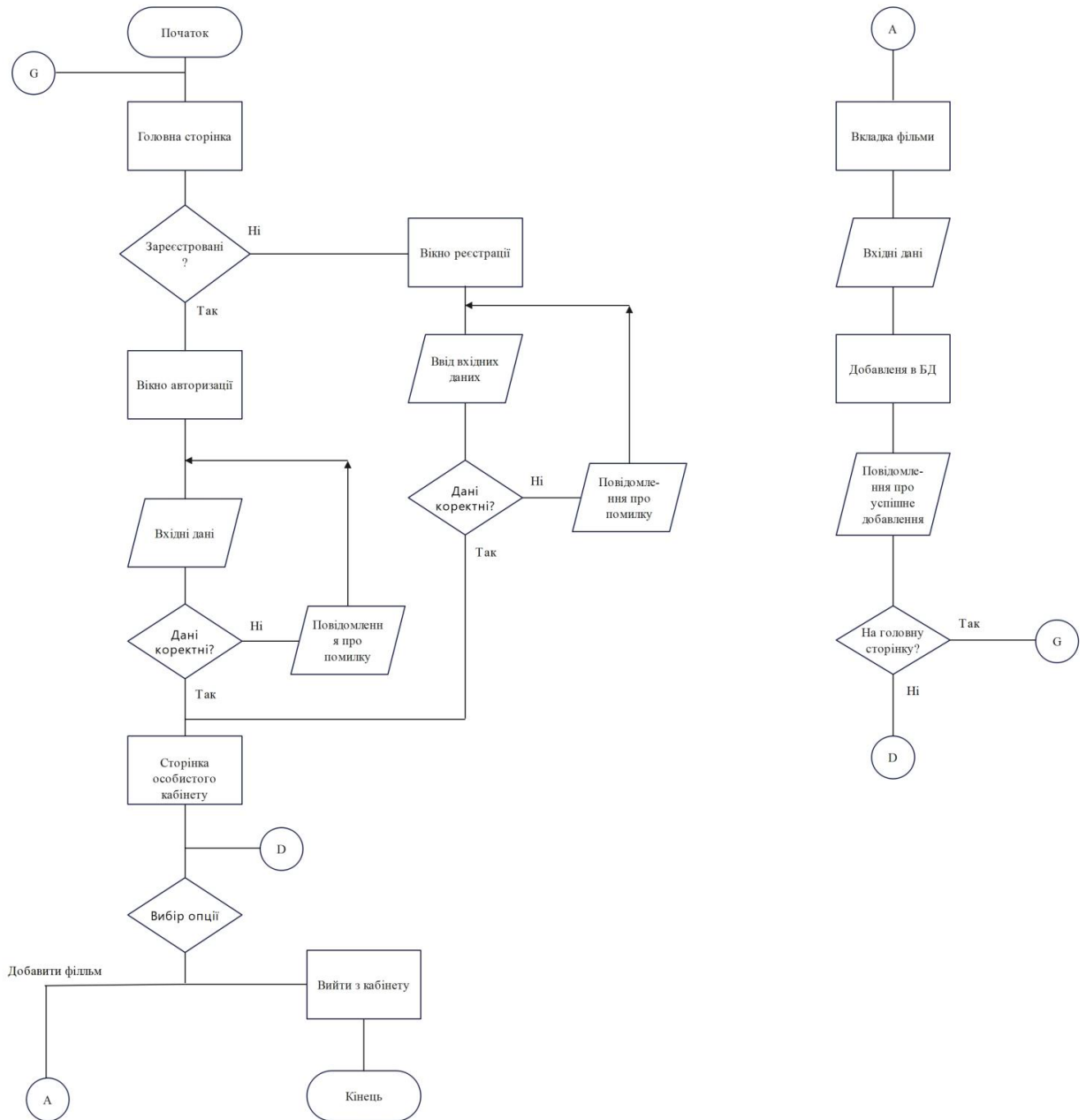


- 17.Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>
- 18.Django документація. URL: <https://docs.djangoproject.com/>
- 19.Bootstrap документація. URL: <https://getbootstrap.com/docs/>
- 20.W3Schools. URL: <https://www.w3schools.com/>
- 21.Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С.  
Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим(бакалаврським) рівнем вищої освіти.— Тернопіль: ЗУНУ, 2021. 56 с.

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

# ДОДАТОК А

## СХЕМА АЛГОРИТМУ РОБОТИ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ



ДОДАТОК Б  
ДІАГРАМА СТАНІВ ТА ПЕРЕХОДІВ ВЕБ-ОРІЄНТОВАНОЇ  
ІНФОРМАЦІЙНОЇ СИСТЕМИ

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

## ДОДАТОК В

### КОД ПРОГРАМИ ОБРОБКИ СТОРІНОК

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.forms import AuthenticationForm, UserCreationForm
from django.contrib.auth import login, authenticate
from .models import Artiles
from datetime import datetime
import requests

def index(request):
    dt = datetime.now()
    data = dt.strftime("%d.%m.%Y")
    film = Artiles.objects.all()
    expected_films = Artiles.objects.filter(premiere_date__gte=dt)
    mnuw_films = Artiles.objects.filter(premiere_date__lt=dt)
    return render(request, 'main/index.html', {'film': film, 'mnuw_films': mnuw_films, 'expected_films': expected_films,
'data': data})

def login_reg(request):
    if request.method == 'POST':
        # Отримання даних форми входу
        login_form = AuthenticationForm(request, data=request.POST)
        # Отримання даних форми реєстрації
        register_form = UserCreationForm(request.POST)

        if login_form.is_valid():
            # Вхід в систему
            user = login_form.get_user()
            login(request, user)
            return redirect('about')

        if register_form.is_valid():
            # Реєстрація нового користувача
            register_form.save()
            username = register_form.cleaned_data.get('username')
            password = register_form.cleaned_data.get('password1')
            user = authenticate(username=username, password=password)
            login(request, user)
            return redirect('home')
        else:
            login_form = AuthenticationForm()
            register_form = UserCreationForm()

    return render(request, 'main/login_reg.html', {'login_form': login_form, 'register_form': register_form})

def post(request, post_slug):
    post = get_object_or_404(Artiles, slug=post_slug)
    return render(request, 'main/post.html', {'post': post})

def update_imdb_ratings():
    films = Artiles.objects.all()
    for film in films:
        imdb_id = film.imdb_id
        response = requests.get(f"https://api.imdb.com/ratings/{imdb_id}")
        if response.status_code == 200:
            imdb_rating = response.json()["rating"]
            film.imdb_rating = imdb_rating
            film.save()
```

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

```

def update_imdb_id():
    tmdb_api_key = '252fbd92540340f4917b023b74bd331'

    films = Articles.objects.filter(imdb_id__isnull=True)
    for film in films:
        title = film.title

        search_url = f"https://api.themoviedb.org/3/search/movie?api_key={tmdb_api_key}&query={title}&language=uk"
        response = requests.get(search_url)
        if response.status_code == 200:
            data = response.json()
            results = data['results']
            if results:
                movie = results[0]
                imdb_id = movie['imdb_id']
                film.imdb_id = imdb_id
                film.save()

def about(request):
    if request.method == 'POST':
        image = request.FILES['img']
        title = request.POST['title']
        english_title = request.POST['title_en']
        short_description = request.POST['anons']
        full_description = request.POST['ful_text']
        release_date = request.POST['premiere_date']

        movie = Articles(img=image, title=title, title_en=english_title, anons=short_description,
            ful_text=full_description, premiere_date=release_date)
        movie.save()
        update_imdb_id()
        update_imdb_ratings()

    return redirect('about')
return render(request, 'main/about.html')

```

					ДП.КН.9702718.053.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53