

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Відокремлений структурний підрозділ  
«Тернопільський фаховий коледж  
Тернопільського національного технічного університету імені Івана Пулюя»  
Відділення телекомунікацій та електронних систем  
Циклова комісія комп'ютерних наук**

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до кваліфікаційної роботи**

**фахового молодшого бакалавра**

**на тему: Розробка вебзастосунку для автоматизації роботи кінотеатру  
«Grand Vision»**

Виконав: студент IV курсу, групи КН-421  
спеціальності: 122 Комп'ютерні науки

Олег В'ЮННИК

---

Керівник \_\_\_\_\_ Галина МАРЦІЯШ

Рецензент \_\_\_\_\_  
(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ  
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ  
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА  
ПУЛЮЯ»

Відділення телекомунікацій та електронних систем  
Циклова комісія комп'ютерних наук  
Освітньо-професійний ступінь «фаховий молодший бакалавр»  
Спеціальність 122 «Комп'ютерні науки»  
Галузь знань 12 «Інформаційні технології»

**ЗАТВЕРДЖУЮ**

Голова циклової комісії  
комп'ютерних наук  
\_\_\_\_\_ Галина МАРЦІЯШ  
« 02 » березня 2026 року

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

В'юнникові Олегу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебзастосунку для автоматизації роботи кінотеатру «Grand Vision»

керівник роботи Марціяш Галина Ярославівна,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу №4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: \_\_\_\_\_ р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: Java, JavaScript, JSX, SQL, HTML, CSS; фреймворки: Spring Boot, Hibernate, React, Vite, Tailwind CSS; база даних PostgreSQL; стандарти IEEE 29148-2018, IEEE 29119, ДСТУ 8302:2015.

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

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу web-сайту

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

## 2 Розробка технічного та робочого проекту

### 2.1 Розробка структури сайту і web-сторінок

### 2.2 Створення та верстка сторінок сайту

### 2.3 Розробка структури бази даних сайту

### 2.4 Програмування сайту

#### 2.4.1 Написання клієнтської частини

#### 2.4.2 Написання admin- частини

### 2.5 Тестування web- сайту

## 3 Спеціальний розділ

### 3.1 Інструкція з розміщення сайту в Інтернеті

### 3.2 Інструкція з обслуговування та наповнення сайту

### 3.3 Інструкція з популяризації та підтримки сайту

## 4 Економічний розділ

### 4.1 Визначення стадій технологічного процесу та загальної тривалості

#### проведення НДР

### 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

### 4.3 Розрахунок витрат на електроенергію

### 4.4 Розрахунок суми амортизаційних відрахувань для розробки вебзастосунку для автоматизації кінотеатру «Grand Vision»

### 4.5 Обчислення накладних витрат

### 4.6 Складання кошторису витрат та визначення собівартості розробки вебзастосунку для автоматизації кінотеатру «Grand Vision»

### 4.7 Розрахунок ціни розробки вебзастосунку для автоматизації кінотеатру «Grand Vision»

### 4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

## 5 Охорона праці, техніка безпеки та екологічні вимоги

### 5.1 Охорона праці в Україні. Законодавчі акти з охорони праці

### 5.2 Основні визначення електробезпеки

## 6 Висновки (навести результати роботи по кожному розділу зокрема і загальний висновок по кваліфікаційній роботі)

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – вебсайту.

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

1. Структурна схема web-сайту
2. Схема використання прецедентів

3. ER діаграма баз даних
4. Таблиця техніко-економічних показників

#### 6. Консультанти розділів роботи

Розділ	Ім'я та прізвище, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Геннадій ГОЧЯЧЕК		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	20.03.2026
2	Збір і узагальнення інформації по роботі	01.05.2026	01.05.2026
3	Написання першого розділу	15.05.2026	15.05.2026
4	Розробка технічного та робочого проекту	29.05.2026	29.05.2026
5	Написання спеціального розділу	05.06.2026	05.06.2026
6	Розрахунок економічної частини	08.06.2026	08.06.2026
7	Написання розділу охорони праці	09.06.2026	09.06.2026
8	Виконання графічної частини	10.06.2026	10.06.2026
9	Оформлення роботи	11.06.2026	11.06.2026
10	Погодження нормоконтролю	15.06.2026	15.06.2026
11	Попередній захист роботи	16.06.2026	16.06.2026
12	Захист роботи	23.06.2026	23.06.2026

7. Дата видачі завдання: 05 березня 2026 р.

Студент

\_\_\_\_\_

( підпис )

\_\_\_\_\_

( ім'я та прізвище )

Керівник роботи

\_\_\_\_\_

( підпис )

\_\_\_\_\_

( ім'я та прізвище )

## ЗМІСТ

АНОТАЦІЯ .....	7
ВСТУП .....	8
1 ЗАГАЛЬНИЙ РОЗДІЛ .....	9
1.1 Аналітичний огляд існуючих рішень .....	9
1.2 Технічне завдання .....	12
1.2.1 Найменування та область застосування .....	12
1.2.2 Призначення розробки .....	13
1.2.3 Вимоги до функціоналу web-сайту .....	14
1.2.4 Вимоги до програмної документації .....	16
1.2.5 Техніко-економічні показники .....	17
1.2.6 Стадії та етапи розробки .....	18
1.2.7 Порядок контролю та прийому .....	19
2 РОЗРОБКА ТЕХНІЧОГО ТА РОБОЧОГО ПРОЄКТУ .....	20
2.1 Розробка структури сайту і web-сторінок .....	20
2.2 Створення та верстка сторінок сайту .....	23
2.3 Розробка структури бази даних сайту .....	33
2.4 Програмування сайту .....	38
2.4.1 Написання клієнтської частини .....	44
2.4.2 Написання admin- частини .....	45
2.5 Тестування web- сайту .....	46
3 СПЕЦІАЛЬНИЙ РОЗДІЛ .....	56
3.1 Інструкція з розміщення сайту в Інтернеті .....	56
3.2 Інструкція з обслуговування та наповнення сайту .....	57
3.3 Інструкція з популяризації та підтримки сайту .....	58
4 ЕКОНОМІЧНИЙ РОЗДІЛ .....	60

<b>2026.КВР.122.421.04.00.00 ПЗ</b>				
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>
Розроб.		В'юнник О.О.		
Перевір.		Марціяш Г.Я.		
Реценз.				
Н. Контр.		Приймак В.А.		
Затверд.				
Розробка вебзастосунку для автоматизації роботи кінотеатру «Grand Vision» <b>Пояснювальна записка</b>				
		<b>Лім.</b>	<b>Арк.</b>	<b>Аркушів</b>
			5	97
<b>ВСП ТФК ТНТУ КН-421</b> <b>м. Тернопіль</b>				

4.1	Визначення стадій технологічного процесу та загальної тривалості проведення НДР.....	60
4.2	Визначення витрат на оплату праці та відрукувань на соціальні заходи	58
4.3	Розрахунок витрат на електроенергію .....	61
4.4	Розрахунок суми амортизаційних відрахувань для розробки вебзастосунку для автоматизації кінотеатру «Grand Vision» .....	63
4.5	Обчислення накладних .....	63
4.6	Складання кошторису витрат та визначення собівартості розробки вебзастосунку для автоматизації кінотеатру «Grand Vision» .....	65
4.7	Розрахунок ціни розробки вебзастосунку для автоматизації кінотеатру «Grand Vision» .....	65
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень .....	66
5	ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ	68
5.1	Охорона праці в Україні. Законодавчі акти з охорони праці .....	68
5.2	Основні визначення електробезпеки .....	72
	ВИСНОВКИ.....	76
	ПЕРЕЛІК ПОСИЛАНЬ.....	78
	ДОДАТКИ.....	80
	Додаток А – Лістинг BookingController.....	80
	Додаток Б – Лістинг BookingService .....	82
	Додаток В – Лістинг SeatHoldService .....	88
	Додаток Г – Лістинг Booking .....	91
	Додаток Д – Лістинг Session .....	92
	Додаток Е – Лістинг User .....	94
	Додаток Ж – Лістинг Movie .....	96

## АНОТАЦІЯ

Тема кваліфікаційної роботи: розробка вебзастосунку для автоматизації роботи кінотеатру «Grand Vision».

Метою роботи є розробка web-сайту для автоматизації діяльності мережі кінотеатрів, що забезпечує перегляд афіші фільмів, пошук і фільтрацію кінострічок, вибір сеансу та місця в залі, оформлення електронного квитка, авторизацію користувачів, роботу особистого кабінету та адміністрування даних.

Пояснювальна записка складається з п'яти розділів. У роботі виконано аналіз існуючих рішень, сформовано технічне завдання, розроблено структуру web-сайту, базу даних, клієнтську, серверну й адміністративну частини та проведено тестування. Клієнтська частина реалізована з використанням React та Vite, серверна частина — на основі Spring Boot, база даних — PostgreSQL.

Обсяг пояснювальної записки — 97 сторінок.

Графічна частина містить структурну схему web-сайту, схему використання прецедентів, ER-діаграму бази даних та техніко-економічні показники програмного продукту.

Qualification work topic: Development of a web application for automating the operation of the “Grand Vision” cinema.

The purpose of the work is to develop a website for automating cinema activities: browsing movies, searching and filtering the catalog, selecting sessions and seats, ordering electronic tickets, user authorization, personal account use, and data administration.

The explanatory note consists of five sections. It includes analysis of existing solutions, technical specification, website structure, database, client, server and administrative parts, and testing. The client part is developed with React and Vite, the server part with Spring Boot, and the database with PostgreSQL.

The explanatory note contains 97 pages.

The graphical part includes the website structure diagram, use case diagram, ER diagram, and technical and economic indicators.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

На сьогоднішній день індустрія кінопрокату та надання розважальних послуг переживає етап активного переходу до цифрових сервісів. Сучасний клієнт вимагає максимальної зручності та швидкості при отриманні послуг, віддаючи перевагу онлайн-обслуговуванню. В умовах високої конкуренції це змушує кінотеатри відходити від традиційних моделей касових продажів і переходити до впровадження комплексних веб-систем. Актуальність теми даної кваліфікаційної роботи зумовлена тим, що значна частина існуючих на вітчизняному ринку рішень побудована на застарілих багатосторінкових архітектурах. Такі системи не здатні забезпечити високу швидкість відмальовування інтерфейсу та стабільність роботи під час пікових навантажень, наприклад, у дні очікуваних кінопрем'єр. Кожна дія користувача супроводжується повним перезавантаженням сторінки, що погіршує клієнтський досвід та створює надмірне навантаження на сервер. Перехід до сучасної архітектури односторінкових застосунків є інженерним стандартом, що ефективно вирішує вказані проблеми, підвищує швидкість і зручність роботи веб-сайту.

Об'єктом розробки кваліфікаційної роботи є процес комплексної автоматизації операційної та комерційної діяльності кінотеатру «Grand Vision» за рахунок впровадження новітніх інформаційних технологій.

Предметом розробки виступають методи, алгоритми обробки даних та програмні засоби, необхідні для реалізації надійної клієнт-серверної архітектури при створенні вебзастосунку онлайн-бронювання квитків.

Метою проєкту є розробка кроссплатформеного, інтуїтивно зрозумілого та стабільного вебзастосунку, який надасть глядачам можливість зручного перегляду актуальної афіші, ознайомлення з фільмами та швидкого безконтактного бронювання місць у режимі реального часу. Окрім клієнтського функціоналу, важливою складовою мети є створення багатофункціональної, захищеної панелі адміністратора для керівного персоналу, що дозволить здійснювати управління розкладом сеансів, редагувати базу фільмів та проводити аналітичний моніторинг.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

# 1 ЗАГАЛЬНИЙ РОЗДІЛ

## 1.1 Аналітичний огляд існуючих рішень

Сучасний ринок індустрії розваг та надання кінопослуг неможливо уявити без високотехнологічних web-рішень. Стрімкий розвиток електронної комерції (e-commerce) призвів до того, що понад 70% усіх квитків на кіносеанси реалізуються саме через мережу Інтернет. Для обґрунтування доцільності розробки інформаційної системи кінотеатру «Grand Vision» було проведено детальний аналіз існуючих на ринку України та світу аналогів. Дослідження фокусувалося на архітектурних рішеннях, зручності користувацького інтерфейсу (UI/UX) та швидкодії систем.

У якості основних об'єктів для порівняльного аналізу було обрано лідерів українського ринку: платформи мереж «Multiplex», «Планета Кіно», а також сервіси-агрегатори (наприклад, «vkin.ua»).

Аналіз інформаційної системи мережі «Multiplex»

Мережа «Multiplex» володіє однією з найбільших систем онлайн-бронювання в Україні. До основних переваг їхнього web-сайту можна віднести:

- наявність продуманої програми лояльності, яка інтегрована безпосередньо в особистий кабінет користувача;
- підтримку різноманітних платіжних шлюзів (Apple Pay, Google Pay);
- високу швидкість обробки транзакцій.

*Недоліки системи:* головна сторінка сайту перевантажена важкими анімаційними елементами та рекламними банерами, що призводить до високого споживання оперативної пам'яті (RAM) та зниження швидкодії (просідання FPS) на слабких пристроях або мобільних телефонах минулих поколінь. Крім того, процес вибору місць іноді супроводжується затримками при оновленні статусу крісла (вільне/зайняте), що в пікові години навантаження призводить до виникнення колізій.

Аналіз інформаційної системи мережі «Планета Кіно»

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Web-додаток «Планети Кіно» є прикладом масштабного монолітного рішення, яке орієнтоване на преміальний сегмент глядачів.

*Переваги:* система відрізняється деталізованою візуалізацією залів (зокрема технологій IMAX, 4DX), наявністю інтерактивного 3D-перегляду місць та інтеграцією з мобільним застосунком.

*Недоліки системи:* інтерфейс користувача є досить специфічним і має високий поріг входу для нових або літніх клієнтів. Навігація між вибором фільму, кінотеатру та сеансу розділена на занадто велику кількість кроків (сторінок), що змушує клієнта довго чекати на повне перезавантаження DOM-дерева браузера. Відсутність концепції Single Page Application (SPA) в деяких модулях сайту робить користувацький досвід менш "безшовним".

Аналіз платформ-агрегаторів (на прикладі vkino.ua)

Такі платформи збирають розклади з різних незалежних кінотеатрів.

*Переваги:* величезний вибір локацій, єдина точка входу для пошуку будь-якого фільму в місті.

*Недоліки:* агрегатори не мають власної бази даних місць, тому при спробі купити квиток користувача часто перенаправляють на сторонні, застарілі iFrame-віджети або зовнішні сайти кінотеатрів. Це руйнує цілісність сесії, знижує рівень довіри до платформи та ускладнює процес повернення коштів у разі скасування сеансу.

Проведений аналіз існуючих рішень виявив ряд типових проблем, серед яких перевантаженість інтерфейсів зайвими елементами, висока затримка при оновленні стану кінозалу та складна багатоетапна процедура оформлення квитка.

З огляду на це, розробка web-сайту кінотеатру «Grand Vision» є обґрунтованою та актуальною задачею. На противагу проаналізованим аналогам, нова система проектується з урахуванням сучасних підходів до розробки програмного забезпечення.

Використання SPA-архітектури на базі бібліотеки React забезпечує швидкі переходи між каталогом фільмів та формою бронювання без повного перезавантаження сторінки, що ефективно вирішує проблему «важких» сайтів

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

конкурентів. Застосування серверної частини на основі фреймворку Spring Boot із використанням легковагового REST API забезпечує швидку передачу даних у форматі JSON, що мінімізує навантаження на мережу клієнта. Окрім того, мінімалістичний UI/UX дизайн, реалізований за допомогою Tailwind CSS, дозволяє зосередити увагу користувача виключно на контенті, зводячи процес бронювання до трьох логічних кроків: вибір сеансу, вибір місця та здійснення оплати.

Таким чином, створюваний програмний продукт враховує помилки конкурентів, пропонуючи замовнику сучасне, масштабоване та оптимізоване рішення, яке відповідає сучасним підходам веб-розробки. Результати порівняльного аналізу архітектур для кінотеатрів наведено в таблиці 1.1.

Таблиця 1.1 — Порівняльний аналіз архітектур MPA та SPA для кінотеатрів

Критерій порівняння	«Multiplex»	«Планета Кіно»	Агрегатори (vkinо)	Розроблювана ІС «Grand Vision»
Технологія побудови інтерфейсу	Багатосторінковий (MPA)	Багатосторінковий (MPA)	Застарілі iFrame	Single Page Application (React)
Швидкодія та завантаження (UI)	Низька (перевантажено)	Середня	Висока	Висока (Tailwind CSS)
Архітектура серверної частини	Монолітна	Монолітна	Монолітна	REST API (Spring Boot)
Бронювання та викуп місць	Так	Так	Перенаправлення	Так (власний модуль)
Безпека авторизації	Стандартні сесії	Стандартні сесії	Стандартні сесії	Криптографічна (JWT-токени)
Генерація електронного PDF-квитка	Так	Так	Ні	Так (з унікальним QR-кодом)

## 1.2 Технічне завдання

Метою створення вебзастосунку є комплексна автоматизація операційної та комерційної діяльності кінотеатру «Grand Vision». Система повинна проєктуватися на базі архітектури Single Page Application (SPA) для забезпечення стабільної обробки пікових запитів користувачів у режимі реального часу. У рамках виконання роботи необхідно реалізувати автоматизацію продажу електронних квитків, управління розкладом сеансів, ведення обліку відвідуваності та динамічне ціноутворення із миттєвим оновленням статусу глядацьких місць для уникнення подвійного бронювання.

### 1.2.1 Найменування та область застосування

Повне найменування розробки — вебзастосунок для автоматизації роботи кінотеатру «Grand Vision». Скорочена назва програмного продукту — ІС «Grand Vision».

Область застосування розробки охоплює сферу надання послуг та індустрію розваг, зокрема організаційну та операційну діяльність сучасних кінотеатрів. Упровадження даного програмного рішення спрямоване на комплексну цифровізацію процесів взаємодії між клієнтом та підприємством. Завдяки використанню сучасних вебтехнологій, система є кросплатформною та може застосовуватися на будь-яких пристроях користувачів із доступом до мережі Інтернет через стандартні браузері без необхідності встановлення додаткового програмного забезпечення.

Цільовим середовищем експлуатації програмного продукту є інформаційна інфраструктура закладу кінопрокату. З боку обслуговування відвідувачів система застосовується для надання зручних інструментів віддаленого онлайн-бронювання. Це дозволяє глядачам самостійно ознайомлюватися з актуальним репертуаром, обирати бажані сеанси та здійснювати резервування або покупку електронних квитків у режимі реального часу, що суттєво знижує навантаження на класичні каси

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

кінотеатру.

У контексті внутрішнього адміністрування область застосування системи охоплює робочі місця персоналу та керівництва кінотеатру. Програмний продукт виступає основним інструментом для оптимізації обліку репертуару, гнучкого управління розкладом показів та систематизації даних про продажі квитків. Крім того, застосунок використовується для безперервного моніторингу заповнюваності залів та збору статистичної інформації, що в сукупності дозволяє підвищити загальний рівень керованості бізнес-процесами підприємства та покращити якість надання послуг.

Архітектурні рішення, закладені на етапі проектування, передбачають гнучкість та можливість подальшого горизонтального масштабування системи. Це дозволить у перспективі інтегрувати нові філії кінотеатру або додаткові зали до єдиної інформаційної мережі підприємства без необхідності глибокого рефакторингу базового коду.

### 1.2.2 Призначення розробки

Основне призначення web-сайту, що проектується, полягає у створенні єдиного інтерактивного середовища для ефективної взаємодії між глядачами та підприємством кінопрокату. З точки зору клієнта майбутня система має забезпечити швидкий, безпечний та зручний доступ до актуальної інформації про кінопрем'єри, розклад сеансів, жанри, трейлери, вікові обмеження та інші відомості, необхідні для вибору фільму. Передбачається, що користувач зможе переглядати доступні сеанси, обирати зручний час показу та здійснювати вибір місця на інтерактивній схемі залу без необхідності особистого звернення до каси кінотеатру.

З точки зору адміністрації кінотеатру web-сайт має бути призначений для централізованого керування основними інформаційними ресурсами кінотеатру. Зокрема, передбачається можливість оновлення бази фільмів, формування розкладу показів, керування кінозалами, встановлення вартості квитків залежно від

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

формату сеансу та перегляду статистичних даних. Такий підхід має зменшити кількість рутинних операцій, пов'язаних із ручним веденням розкладу та обробкою звернень клієнтів, а також сприяти оптимізації роботи персоналу.

У комерційному контексті проєктований web-сайт має забезпечити технічну основу для автоматизації процесу продажу квитків і підвищення якості обслуговування глядачів. Використання такого програмного засобу дозволить скоротити черги у фізичних касах, пришвидшити оформлення замовлень, знизити ймовірність помилок під час вибору місць та забезпечити збереження інформації про придбані квитки в електронному вигляді.

Окрім автоматизації процесу бронювання, майбутня інформаційна система може бути використана для формування структурованої клієнтської бази та подальшого аналізу попиту на окремі фільми, сеанси й формати показу. Отримані дані можуть застосовуватися адміністрацією кінотеатру для планування репертуару, оцінювання популярності кінострічок, визначення пікових годин відвідуваності та підготовки маркетингових пропозицій. Таким чином, призначення розробки полягає у створенні сучасного web-засобу, який має перевести основні процеси взаємодії між кінотеатром і глядачем у цифровий формат.

### **1.2.3 Вимоги до функціоналу web-сайту**

Програмний продукт повинен забезпечувати комплексний набір функцій, який логічно розділяється на модулі відповідно до ролей користувачів у системі (Role-Based Access Control).

Функціонал для неавторизованих користувачів (гостей) повинен передбачати можливість перегляду головної сторінки з актуальною афішею та новинками прокату. Користувачі повинні мати безперешкодний доступ до детальної інформації про кожен фільм, що включає опис, акторський склад, тривалість, мову, віковий рейтинг та можливість перегляду трейлера. Для зручності навігації має бути реалізовано використання фільтрів та пошуку по каталогу

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

фільмів за жанром, датою чи назвою, а також має бути доступний перегляд розкладу сеансів для обраного фільму на конкретні дати.

Для авторизованих користувачів (клієнтів) має забезпечуватися безпечна реєстрація та авторизація в системі. Ця цільова група повинна отримувати доступ до інтерактивної схеми кінозалу для візуального вибору вільних місць та подальшого оформлення процесу бронювання із чіткою фіксацією обраних крісел за конкретним користувачем. Після успішного оформлення система повинна автоматично відправляти email-повідомлення з даними квитка. Крім того, клієнти повинні мати доступ до особистого кабінету для перегляду історії бронювань, завантаження електронного PDF-квитка з QR-кодом та мати можливість залишати відгуки чи оцінки переглянутим кінострічкам.

Функціональні вимоги для адміністратора повинні включати надання доступу до захищеної панелі керування (Dashboard), де має бути реалізовано повний спектр CRUD-операцій (створення, читання, оновлення, видалення) для управління каталогом фільмів. Адміністратор повинен здійснювати управління розкладом, зокрема створення нових сеансів, прив'язку їх до конкретних кінозалів, встановлення формату показу (2D, IMAX) та вартості квитків. Також мають бути передбачені спеціалізовані інструменти для безперервного моніторингу статистики проданих квитків та контролю заповнюваності залів.

До комплексу нефункціональних вимог розробки належить забезпечення належного рівня безпеки, продуктивності та адаптивності. Ідентифікація авторизованих користувачів у системі повинна реалізовуватися з використанням JWT-токенів, а доступ до адміністративного інтерфейсу має жорстко обмежуватися відповідно до ролі. Для промислового розгортання системи має бути передбачено використання захищеного протоколу HTTPS та винесення конфіденційних параметрів у змінні середовища. Час відгуку серверної частини на стандартні API-запити не повинен перевищувати 300 мс для забезпечення плавності роботи інтерфейсу. З боку кросбраузерності та адаптивності, клієнтська частина повинна коректно відображатися на всіх сучасних пристроях (смартфони, планшети, ПК) та безпомилково підтримуватися актуальними версіями браузерів Chrome, Safari,

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Firefox та Edge.

#### 1.2.4 Вимоги до програмної документації

Склад програмних документів визначається залежно від архітектурної складності програмного виробу, а також з урахуванням кваліфікації технічного персоналу, що обслуговуватиме web-сайт у майбутньому. Уся документація повинна розроблятися у відповідності до стандартів ЄСПД (Єдина система програмної документації) та базуватися на сучасних практиках інженерії програмного забезпечення. Однією з ключових вимог до вихідного коду програми є його самодокументованість. Усі основні програмні модулі, зокрема контролери, сервісні класи та репозиторії, повинні містити необхідні внутрішні коментарі у стандартному форматі JavaDoc. Ця документація має детально пояснювати призначення кожного методу, описувати вхідні параметри, а також специфікувати типи даних, що повертаються, задля полегшення подальшого супроводу, тестування та масштабування інформаційної системи.

Окрема увага приділяється специфікації програмних інтерфейсів. Архітектура REST API повинна бути чітко та структуровано задокументована із зазначенням усіх доступних ендпоінтів, підтримуваних HTTP-методів, а також точних форматів тіла запитів (Request Body) та відповідей сервера (Response Body) у форматі JSON. Крім того, для забезпечення успішного впровадження проєкту розробник зобов'язаний надати повний пакет експлуатаційної документації. Цей пакет має обов'язково включати детальну «Інструкцію з розгортання» для системного адміністратора, яка містить покроковий опис процесу конфігурації вебсервера, розгортання бази даних PostgreSQL, налаштування змінних середовища та безпосереднього запуску виконуваного .jar файлу. Для кінцевих користувачів системи з боку адміністрації кінотеатру додатково розробляється «Інструкція користувача», що регламентує порядок роботи із захищеною панеллю керування та детально описує основні сценарії управління контентом.

Для зручності подальшого масштабування проєкту документація також

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

повинна містити візуальні схеми архітектури, структурні ER-діаграми бази даних та діаграми прецедентів. Опис REST API рекомендується підтримувати у стандартизованому форматі, що значно спростить можливу майбутню інтеграцію з мобільними додатками або сторонніми сервісами партнерів.

### 1.2.5 Техніко-економічні показники

Розробка інформаційної системи онлайн-бронювання квитків для кінотеатру «Grand Vision» повинна бути обґрунтованою та доцільною з економічної точки зору. Головною метою проєктування є створення високорентабельного програмного продукту, що дозволить оптимізувати початкові інвестиції на його розробку та впровадження при максимізації подальшої фінансової вигоди для підприємства. Відповідно до вимог життєвого циклу розробки програмного забезпечення, на початковому етапі формується виключно загальний прогноз майбутніх витрат без детальної калькуляції.

Передбачається, що під час створення вебзастосунку основна частина капіталовкладень буде спрямована на фонд оплати праці інженерно-технічного персоналу (розробників, тестувальників) та відповідні обов'язкові відрахування на соціальні заходи. До супутніх статей витрат, які формуватимуть планову собівартість проєкту, належатимуть витрати на спожиту електроенергію, амортизацію задіяної обчислювальної техніки, витрати на оренду хмарних серверів для розгортання бази даних, а також загальновиробничі накладні витрати. Точний розрахунок загальної трудомісткості, повної собівартості та відпускної ціни програмного виробу буде наведено у відповідному економічному розділі роботи.

Висока економічна ефективність впровадженої системи підтверджуватиметься позитивними показниками чистої теперішньої вартості та рентабельності. Очікується, що прийнятний прогнозований термін окупності капітальних інвестицій буде досягнуто завдяки комплексному економічному ефекту. Цей ефект включає повну автоматизацію касових операцій, оптимізацію робочого часу персоналу кінотеатру, ліквідацію витрат на друк паперових квитків,

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

а також суттєве збільшення загального обсягу продажів за рахунок цілодобового доступу клієнтів до онлайн-бронювання.

### 1.2.6 Стадії та етапи розробки

Процес створення програмного комплексу базується на ітеративній моделі життєвого циклу розробки ПЗ (SDLC) та поділяється на кілька послідовних стадій. Початковим етапом є збір вимог та аналітика, що включає детальний аналіз предметної області, дослідження існуючих аналогів на ринку (зокрема, мереж Multiplex та Планета Кіно), формування технічного завдання, визначення ключових ролей користувачів та розробку UML-діаграм прецедентів. Наступним кроком є етап проєктування, під час якого здійснюється розробка ER-діаграми реляційної бази даних із нормалізацією таблиць до третьої нормальної форми, проєктування архітектури REST API та створення візуальних макетів інтерфейсу (Wireframes).

Безпосереднє програмування починається з розробки серверної частини (Backend), що передбачає налаштування середовища, написання бізнес-логіки на базі фреймворку Java Spring Boot, конфігурацію системи авторизації через Spring Security, реалізацію алгоритмів бронювання та інтеграцію з СУБД PostgreSQL. Далі виконується розробка клієнтської частини (Frontend) з ініціалізацією проєкту на базі React та Vite, версткою компонентів за допомогою Tailwind CSS, підключенням маршрутизації та налаштуванням HTTP-клієнта для взаємодії з серверним API. Завершальні стадії охоплюють комплексне тестування та впровадження. Етап тестування включає перевірку API-запитів через інструментарій Postman, ручне End-to-End тестування сценаріїв користувача та усунення виявлених дефектів. Фінальний етап полягає у підготовці збірки проєкту для продакшену, розгортанні системи на хостингу, оформленні супровідної документації та офіційній передачі продукту в експлуатацію.

Важливим аспектом є те, що весь процес написання програмного коду супроводжується обов'язковим використанням систем контролю версій (зокрема, Git) для забезпечення надійності, відстеження змін та цілісності проєкту.

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

Додатково на етапі розгортання передбачається можливість базового налаштування процесів безперервної інтеграції (CI/CD), що дозволить значно спростити та автоматизувати доставку нових оновлень на робочий сервер у майбутньому.

Після завершення розробки виконується перевірка працездатності системи, усунення виявлених недоліків і підготовка програмного комплексу до передачі замовнику. На фінальному етапі оцінюється стабільність роботи вебзастосунку та коректність оформлення квитків.

### 1.2.7 Порядок контролю та прийому

Прийом продукту в експлуатацію здійснюється на основі результатів комплексного тестування, що підтверджує його відповідність технічному завданню. Тестування проводиться у підготовленому локальному середовищі. Насамперед перевіряється production-збірка клієнтської частини на базі Vite та компіляція серверної частини Java Spring Boot за допомогою Maven на предмет відсутності критичних помилок.

Далі проводиться інтеграційне тестування API через Postman для валідації REST-запитів, JSON-відповідей та передачі JWT-токенів. Функціональне тестування імітує реальні дії користувача: реєстрацію, вибір сеансу, тимчасове блокування місця, оформлення замовлення та генерацію PDF-квитка. Додатково тестується інтерфейс для підтвердження адаптивності верстки на мобільних і десктопних пристроях та коректності навігації.

Продукт вважається готовим до впровадження за умови проходження основних тестових сценаріїв без виникнення критичних збоїв (Fatal Errors) та повної відповідності фактичного функціоналу затвердженому технічному завданню.

Після усунення виявлених недоліків програмний комплекс передається замовнику для фінальної дослідної експлуатації. Фінальним етапом є перевірка стабільності вебзастосунку та підписання акта приймання-передачі.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

### 2.1 Розробка структури сайту і web-сторінок

Програмний комплекс для автоматизації діяльності мережі кінотеатрів «Grand Vision» побудований за клієнт-серверною архітектурою, що передбачає розподіл системи на окремі функціональні рівні. Такий підхід дає змогу відокремити інтерфейс користувача, логіку обробки запитів та рівень збереження даних, що відповідає принципу розділення відповідальності компонентів програмної системи [2].

Клієнтська частина вебзастосунку реалізована у вигляді односторінкового застосунку на базі React та Vite. Вона відповідає за відображення сторінок сайту, навігацію між розділами, взаємодію користувача з формами, вибір місць у залі, перегляд афіші, оформлення бронювання та роботу особистого кабінету. Серверна частина розроблена на базі Java Spring Boot і виконує обробку HTTP-запитів, перевірку даних, авторизацію користувачів, створення бронювань, формування електронних квитків та взаємодію з базою даних. Для збереження інформації використовується реляційна система керування базами даних PostgreSQL [3].

Обмін даними між клієнтською та серверною частинами здійснюється через REST API. Передавання інформації відбувається у форматі JSON, що забезпечує зручну взаємодію між frontend- і backend-частинами системи. Такий формат обміну даними є доцільним для вебзастосунків, оскільки він підтримується більшістю сучасних клієнтських і серверних технологій [4].

Структура вебзастосунку «Grand Vision» сформована з урахуванням основних сценаріїв роботи користувача та адміністратора. До складу користувацької частини входять сторінки перегляду афіші, пошуку та фільтрації фільмів, перегляду детальної інформації про фільм, вибору міста, сеансу та місця в залі, оформлення замовлення, імітації оплати, перегляду QR-квитка та завантаження PDF-квитка. Окремо передбачено сторінки авторизації, реєстрації та виходу з облікового запису.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

Адміністративна частина вебзастосунку призначена для керування даними системи. Через адміністративну панель адміністратор може переглядати статистику, керувати фільмами, залами та сеансами. Такий розподіл сторінок дозволяє відокремити публічні можливості сайту від службових функцій адміністрування, що підвищує зручність користування системою та спрощує її подальший супровід.

Довідкова частина вебзастосунку містить інформаційні сторінки, необхідні для ознайомлення користувача з роботою кінотеатру. До неї належать сторінки «Про кінотеатр», «Контакти», «Правила» та «Публічна оферта». Наявність таких сторінок забезпечує повноту інформаційного наповнення сайту та робить вебзастосунок придатним не лише для бронювання квитків, а й для загального інформування відвідувачів.

Структурна схема вебзастосунку відображає логіку побудови основних сторінок, їх підпорядкування та взаємозв'язки між користувацькою, адміністративною, довідковою частинами й модулем авторизації. Структурна схема сайту зображена у графічній частині 2026.КВР.122.421.04.00.00 СС [2].

Для відображення функціональної взаємодії користувачів із системою визначено основних акторів: гостя, авторизованого користувача та адміністратора. Гість має доступ до перегляду афіші, пошуку фільмів, ознайомлення з розкладом сеансів та інформаційними сторінками. Авторизований користувач додатково отримує можливість вибору місця, оформлення бронювання, перегляду історії замовлень, завантаження PDF-квитка та використання QR-квитка. Адміністратор має розширені права доступу до службових функцій, пов'язаних із керуванням фільмами, залами, сеансами та переглядом статистики.

Діаграма варіантів використання демонструє розподіл функціональних можливостей між основними акторами системи та показує, які сценарії доступні кожному типу користувача. Діаграма варіантів використання вебзастосунку «Grand Vision» наведена у графічній частині 2026.КВР.122.421.04.00.00 СС [4].

Розмежування функціональних можливостей між акторами системи визначає логіку доступу до окремих модулів вебзастосунку. Неавторизований

					<i>2026.КВР.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

користувач може переглядати відкриту інформацію, однак не має доступу до операцій, що змінюють стан системи. Після авторизації користувач отримує можливість створювати бронювання, переглядати власні квитки та працювати з особистим кабінетом. Адміністратор системи має доступ до адміністративної панелі, через яку здійснюється підтримка актуальності даних програмного комплексу.

Для узагальнення прав доступу основних акторів системи сформовано матрицю ролей, у якій наведено рівні доступу, доступні функції та основні обмеження для кожного типу користувача. Така матриця дозволяє систематизувати функціональні можливості вебзастосунку та показати розподіл прав між гостем, авторизованим користувачем і адміністратором.

Розмежування доступу є необхідним, оскільки кожна категорія користувачів виконує в системі різні дії. Гість працює лише з відкритою частиною сайту: переглядає афішу, шукає фільми, ознайомлюється з розкладом сеансів та довідковою інформацією. На цьому рівні користувач не змінює стан системи, тому для нього не передбачено можливості бронювання або перегляду персональних квитків.

Авторизований користувач після входу в систему отримує доступ до функцій, пов'язаних з оформленням замовлення. Він може вибрати місто, сеанс і місце в залі, виконати бронювання, переглянути сформований електронний квиток та завантажити його у форматі PDF. Крім того, особистий кабінет забезпечує доступ до історії замовлень, що робить використання системи зручнішим для постійних клієнтів.

Адміністратор має доступ до службової частини вебзастосунку, де здійснюється підтримка актуальності даних. До його функцій належать керування фільмами, залами, сеансами, містами, а також перегляд бронювань і статистичних показників. Такий розподіл прав дозволяє відокремити публічні функції сайту від адміністративних операцій, які впливають на інформаційне наповнення програмного комплексу. Зведену матрицю ролей та прав доступу акторів системи наведено в таблиці 2.1.

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 — Матриця ролей та прав доступу акторів системи

Роль / актор	Рівень доступу	Доступні функції системи	Основні обмеження
Гість	Публічний доступ без авторизації	Перегляд афіші фільмів, пошук і фільтрація фільмів, перегляд інформації про фільм, перегляд доступних кінотеатрів, ознайомлення з розкладом майбутніх сеансів	Не може бронювати квитки, тимчасово резервувати місця, переглядати власні квитки та залишати відгуки
Авторизований користувач	Клієнтський доступ після входу в систему	Вибір міста та сеансу, перегляд схеми залу, вибір місця, тимчасове резервування місця, оформлення бронювання, генерація PDF-квитка, отримання квитка на email, перегляд історії покупок, додавання відгуків до фільмів	Не має доступу до адміністративної панелі, не може створювати або редагувати фільми, зали, сеанси та переглядати загальну статистику продажів
Адміністратор	Привілейований доступ до адміністративної панелі	Повний доступ до клієнтського функціоналу, керування фільмами, створення та видалення сеансів, керування залами й містами, перегляд усіх бронювань, перегляд статистики кінотеатру, аналіз доходу та кількості проданих квитків	Не виконує системне адміністрування сервера, не змінює структуру бази даних через інтерфейс сайту, не має доступу до конфіденційних паролів користувачів

## 2.2 Створення та верстка сторінок сайту

Інтерфейс користувача вебзастосунку «Grand Vision» розроблено з урахуванням принципів адаптивної верстки, зручної навігації та візуальної

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

цілісності всіх сторінок. Основною метою створення інтерфейсу було забезпечення зрозумілого доступу користувача до афіші фільмів, розкладу сеансів, сторінки бронювання, особистого кабінету та довідкової інформації. Для реалізації клієнтської частини використано бібліотеку React, а стилізацію елементів виконано засобами Tailwind CSS [9].

Структура frontend-частини побудована на основі окремих сторінок і повторно використовуваних компонентів. Такий підхід дозволяє розділити інтерфейс на логічні частини, спростити супровід програмного коду та забезпечити єдиний стиль оформлення. До основних сторінок вебзастосунку належать головна сторінка, каталог фільмів, сторінка деталей фільму, сторінка бронювання, особистий кабінет користувача, довідкові сторінки та адміністративна панель.

Компонент HomePage виконує роль головної сторінки та є центральною точкою входу до вебзастосунку. Його структуру сформовано за принципом логічного зонування: у верхній частині розміщена навігаційна панель Navbar, що містить логотип, перемикач мови, перемикач теми та елементи переходу до основних розділів сайту. Центральний блок сторінки використовується для візуального представлення кінотеатру «Grand Vision» і короткого ознайомлення користувача з основними можливостями сервісу.

Головна сторінка також містить інформаційні блоки, що підкреслюють переваги вебзастосунку: перегляд актуальних сеансів, вибір зручного формату показу, доступ до електронних квитків та швидкий перехід до афіші. Така структура дозволяє користувачеві швидко зорієнтуватися в інтерфейсі та перейти до перегляду доступних фільмів.

Окрему роль у структурі головної сторінки відіграє навігаційне меню, яке забезпечує перехід до основних розділів вебзастосунку без зайвого перевантаження інтерфейсу. Розміщення ключових елементів у верхній частині сторінки дає змогу користувачеві швидко перейти до афіші, довідкових сторінок або особистого кабінету. Завдяки єдиній стилістиці кнопок, карток і текстових блоків головна сторінка формує цілісне перше враження про сервіс та задає загальний візуальний стиль усього вебзастосунку.

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

Візуальне представлення головної сторінки вебзастосунку наведено на рисунку 2.1.

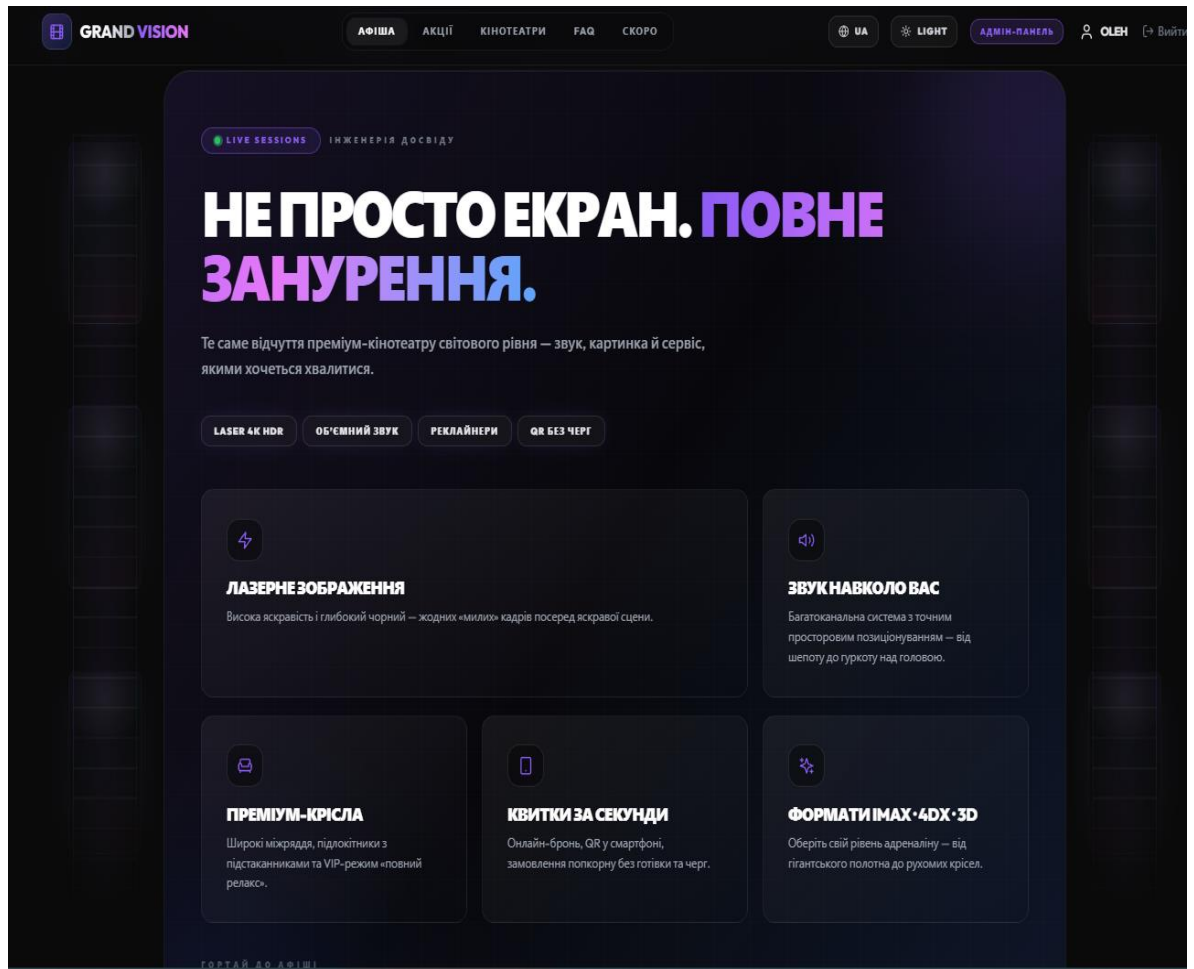


Рисунок 2.1 — Інтерфейс головної сторінки

Окремим розділом користувацької частини є інтерактивна афіша, у якій зосереджено основний функціонал перегляду доступних фільмів. Вона включає рядок швидкого пошуку, фільтрацію за жанрами та адаптивну сітку карток фільмів із відображенням рейтингу і доступних форматів показу. Такий спосіб подання інформації дає змогу користувачеві швидко знайти потрібну кінострічку та перейти до перегляду її детальної інформації.

Картки фільмів мають однакову структуру, що забезпечує візуальну впорядкованість каталогу та полегшує порівняння доступних кінострічок. Кожна картка містить основні відомості про фільм і елемент переходу до сторінки деталей.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

Завдяки адаптивній сітці кількість карток в одному рядку змінюється залежно від ширини екрана, тому сторінка залишається зручною для перегляду на різних пристроях. Інтерфейс каталогу фільмів наведено на рисунку 2.2.

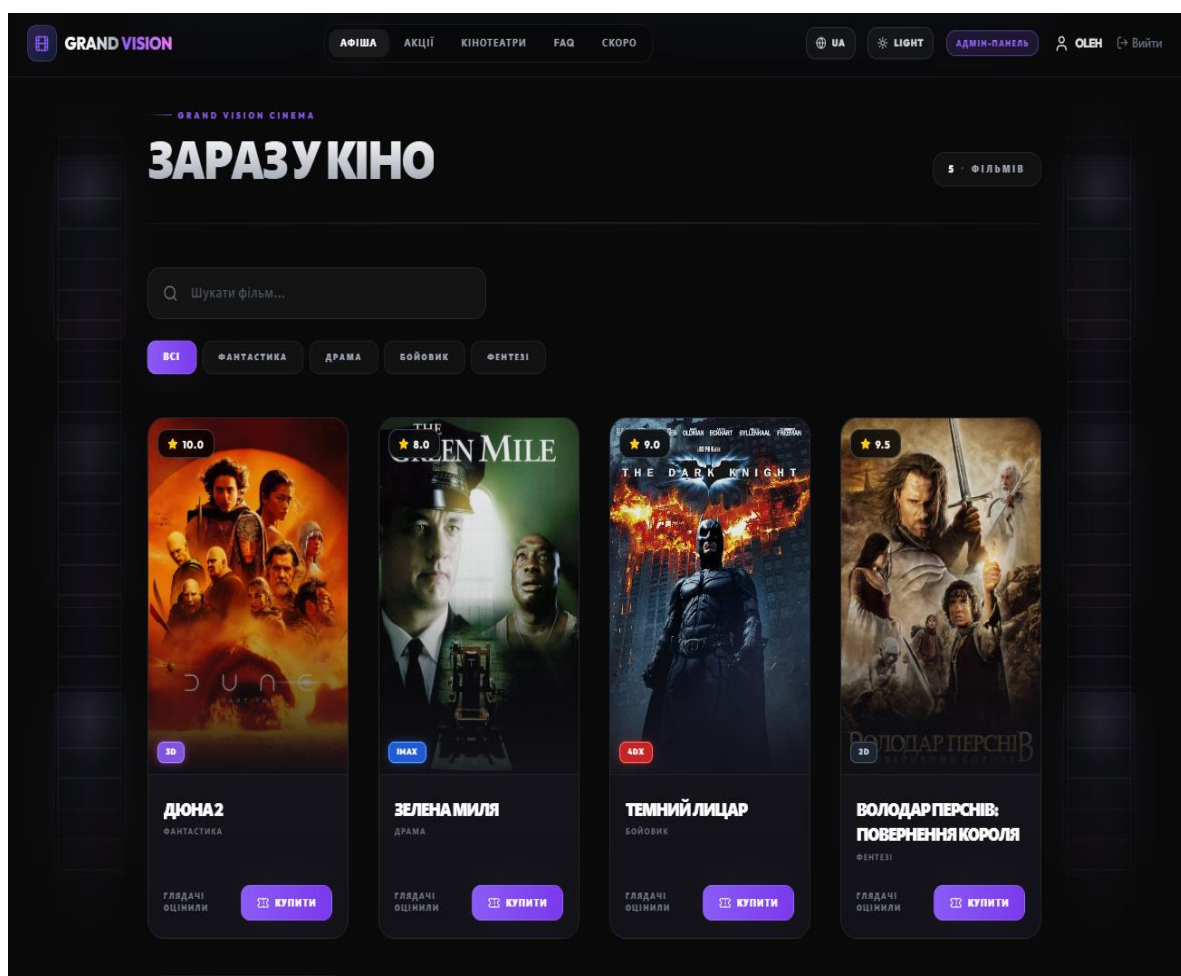


Рисунок 2.2 — Інтерфейс інтерактивного каталогу фільмів

Після вибору конкретного фільму користувач переходить на сторінку деталей кінострічки. Компонент `MovieDetailsPage` забезпечує відображення розширеної інформації про фільм, зокрема постера, назви, жанру, тривалості, рейтингу, короткого опису та доступних сеансів. Така структура сторінки дозволяє користувачеві ознайомитися з основними відомостями про кінострічку перед переходом до бронювання.

Окремим блоком на сторінці розміщено розклад доступних сеансів. Сеанси подаються у вигляді інтерактивних елементів із зазначенням часу показу, формату

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

та вартості квитка. Це спрощує вибір потрібного показу та забезпечує логічний перехід від перегляду опису фільму до сторінки бронювання місця. Візуальне представлення сторінки деталей фільму з розкладом сеансів наведено на рисунку 2.3.

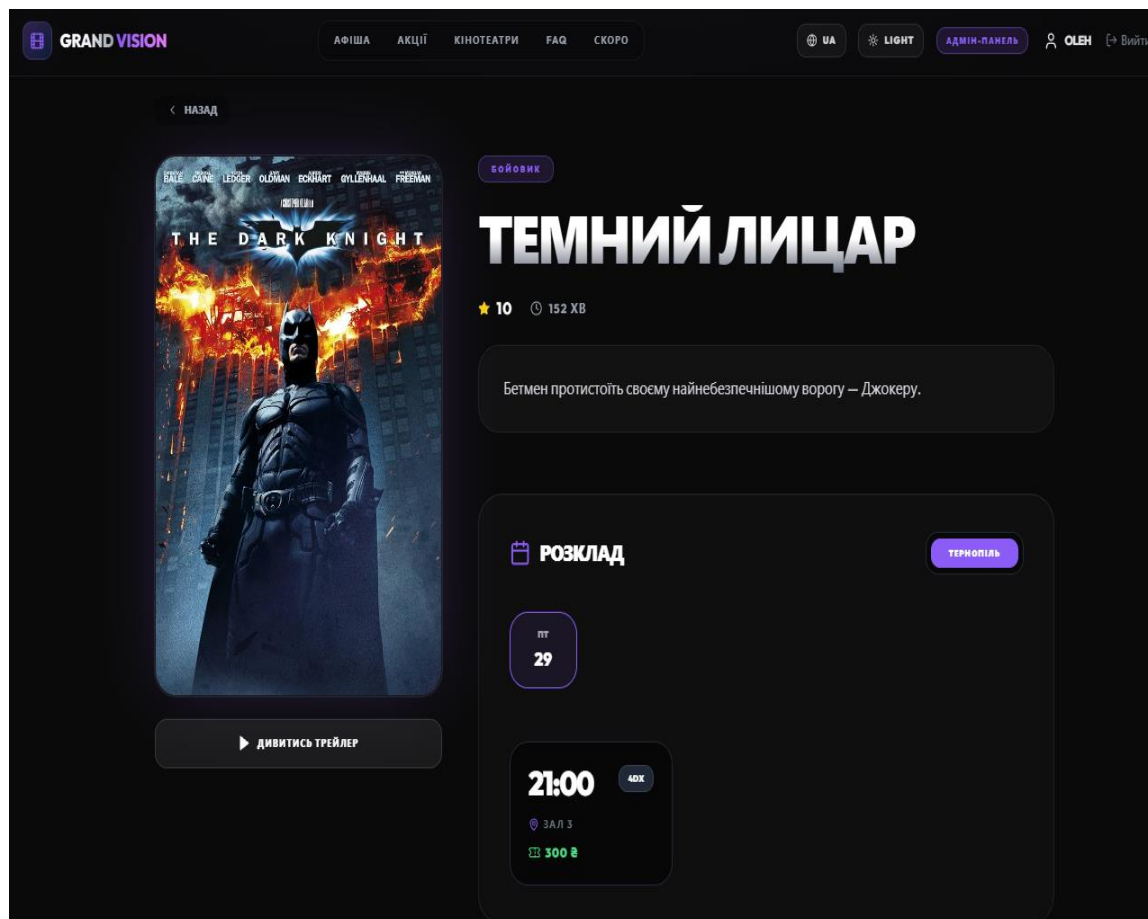


Рисунок 2.3 — Інтерфейс сторінки деталей фільму з розкладом

У нижній частині сторінки деталей фільму передбачено додаткові інформаційні блоки, які доповнюють опис кінострічки. Зокрема, користувач може переглянути блок відгуків, оцінки глядачів та інформацію про акторський склад. Таке компонування сторінки дає змогу поєднати довідкову інформацію про фільм із елементами користувацької взаємодії.

Блок відгуків дозволяє відображати оцінки та коментарі користувачів, що допомагає іншим відвідувачам сформуванати попереднє враження про фільм. Інформація про акторів подається у вигляді окремих карток, що підтримує єдину

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

візуальну стилістику сторінки. Інтерфейс сторінки деталей фільму з відгуками та акторським складом наведено на рисунку 2.4.

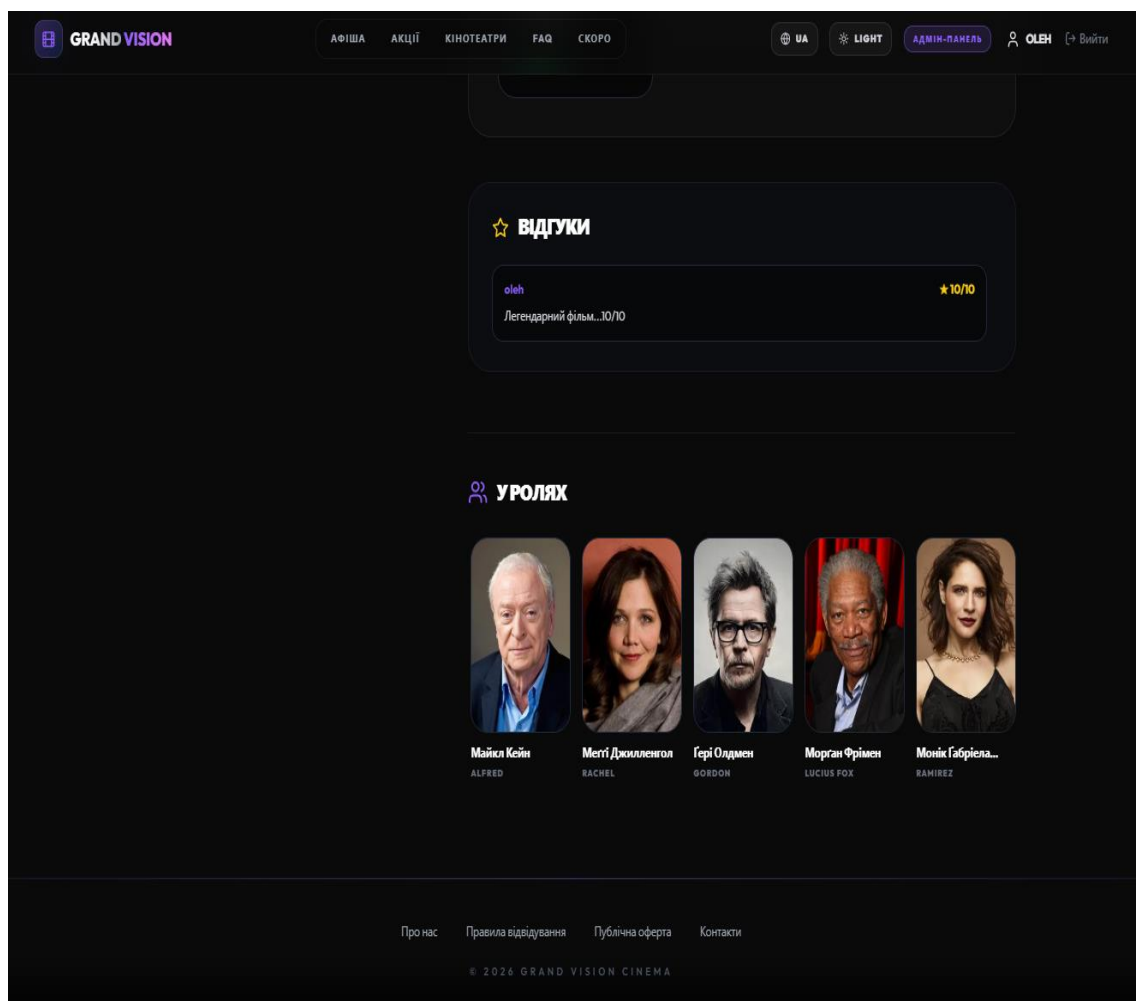


Рисунок 2.4 — Інтерфейс сторінки деталей фільму з відгуками та акторами

Наступним важливим елементом користувацької частини є сторінка бронювання, реалізована у компоненті BookingPage. Вона призначена для переходу від перегляду інформації про фільм до вибору конкретного сеансу та місця в глядацькому залі. Структура сторінки поділена на інформаційний блок із даними про фільм і робочу область, у якій користувач взаємодіє з параметрами бронювання.

У верхній частині сторінки бронювання відображається постер фільму, його назва, базова інформація про сеанс і доступні параметри вибору. Основну частину інтерфейсу займає карта кінозалу, представлена у вигляді сітки місць. Таке

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

компонування дозволяє поєднати інформацію про обраний фільм із візуальним вибором місця в одному інтерфейсному просторі. Інтерфейс мапи кінозалу з вибором місця наведено на рисунку 2.5.

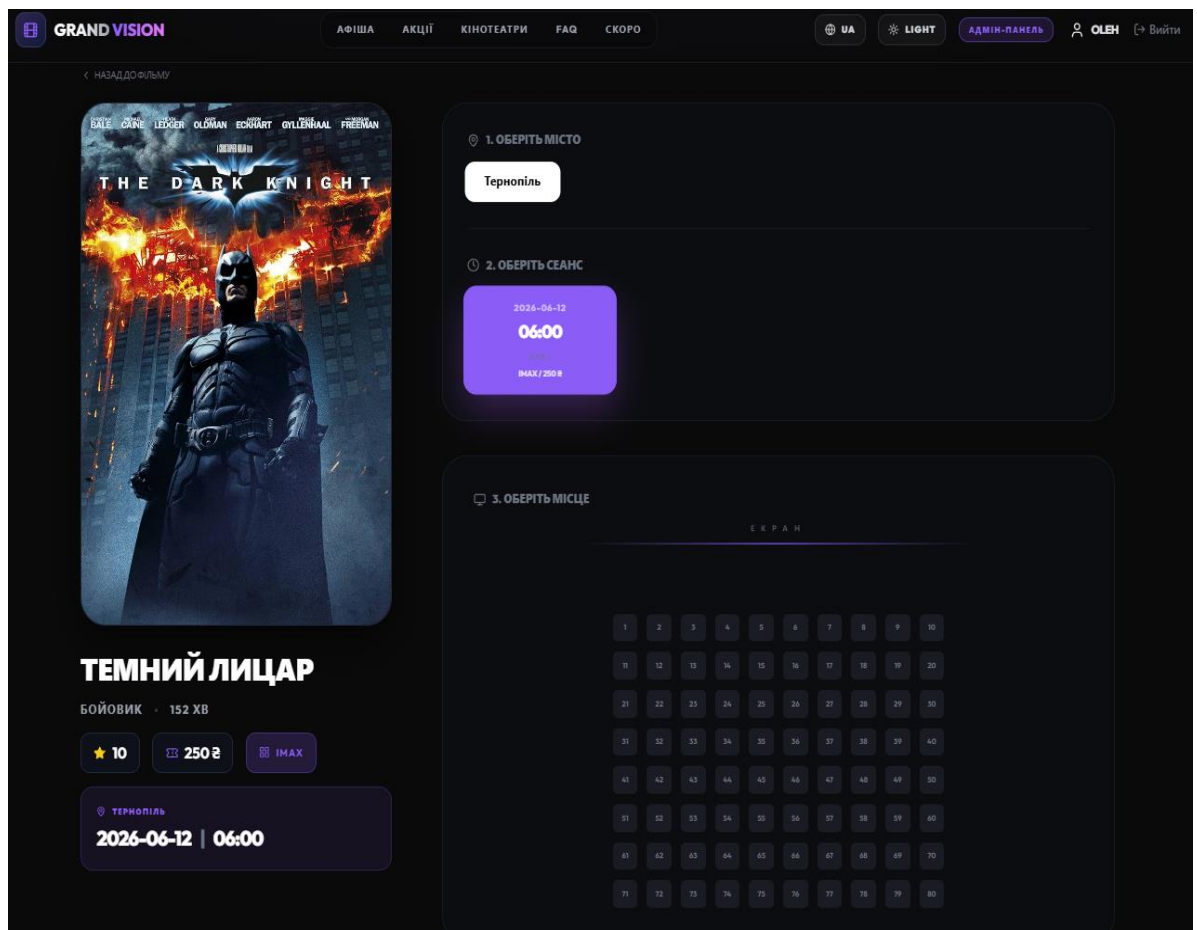


Рисунок 2.5 - Інтерфейс мапи кінозалу

Окремою сторінкою користувацької частини є особистий кабінет, призначений для перегляду інформації про замовлення та електронні квитки. Його інтерфейс побудовано у вигляді карток, у яких відображаються основні дані про придбані квитки: назва фільму, дата й час сеансу, номер залу, місце та статус оплати. У верхній частині сторінки також подано коротку статистику замовлень користувача, що дає змогу швидко оцінити кількість оформлених квитків.

Така структура особистого кабінету забезпечує зручний доступ до історії покупок і дозволяє користувачеві швидко знайти потрібний квиток. Візуальне представлення особистого кабінету користувача наведено на рисунку 2.6.

					<b>2026.КВР.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

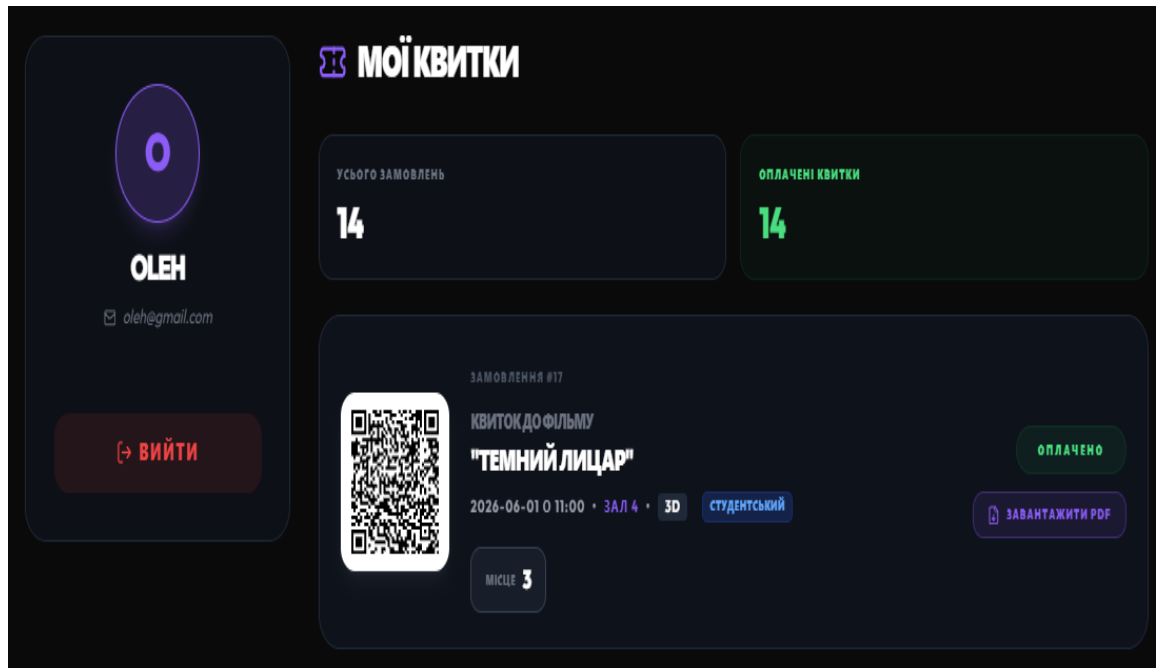


Рисунок 2.6 — Особистий кабінет користувача

Адміністративна частина вебзастосунку реалізована у вигляді окремої панелі керування. Компонент AdminPage містить бічне навігаційне меню та робочу область, у якій відображаються модулі адміністрування. На головному екрані адміністративної панелі подано статистичні показники роботи системи, зокрема кількість проданих квитків, кількість сеансів, фільмів та інші узагальнені дані. Інтерфейс адміністративної панелі зі статистикою наведено на рисунку 2.7.

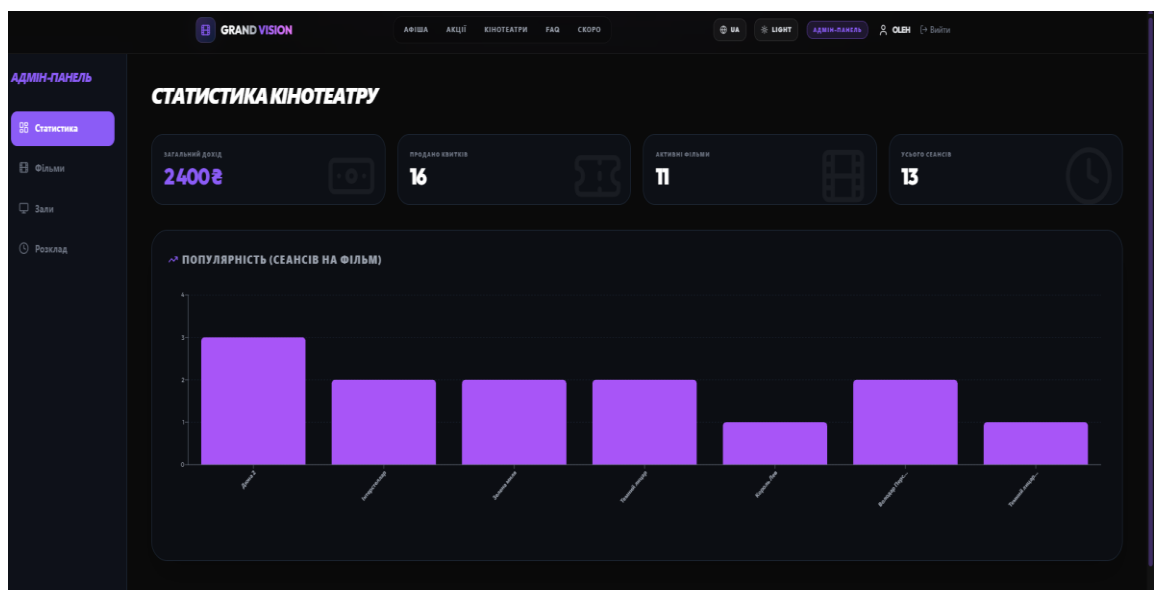


Рисунок 2.7 — Дашборд панелі адміністратора зі статистикою кінотеатру

Для наповнення репертуару кінотеатру використовується модуль керування фільмами. Він містить форму для введення основних даних про кінострічку, зокрема назви, жанру, тривалості, опису, посилання на постер і трейлер. Нижче розміщено список уже доданих фільмів із можливістю їх редагування або видалення. Інтерфейс керування фільмами наведено на рисунку 2.8.

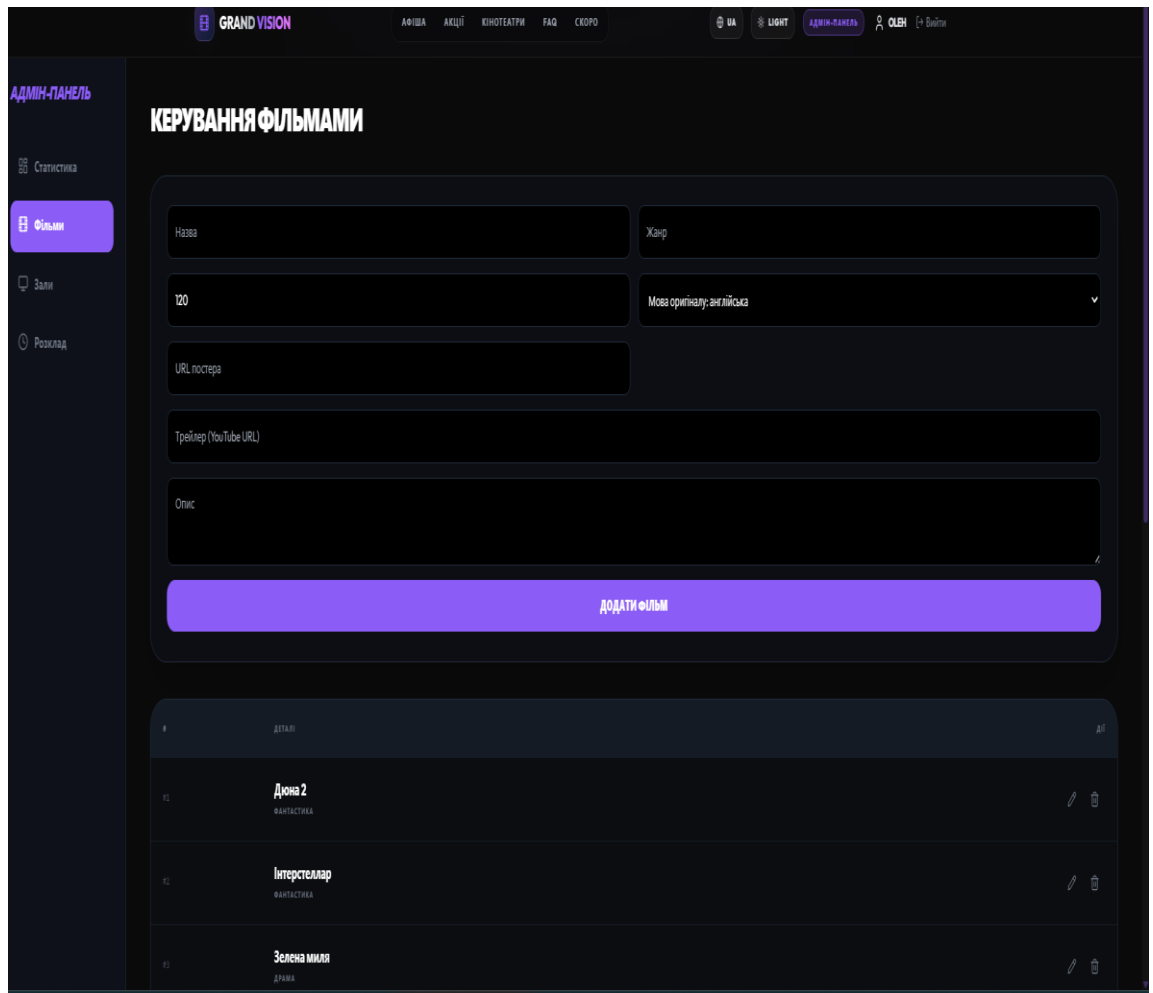


Рисунок 2.8 — Керування фільмами

Модуль керування залами призначений для створення та редагування інформації про кінозали. У формі адміністратор зазначає назву залу, місто та місткість, що надалі використовується під час формування схеми місць на сторінці бронювання. Інтерфейс керування залами наведено на рисунку 2.9.

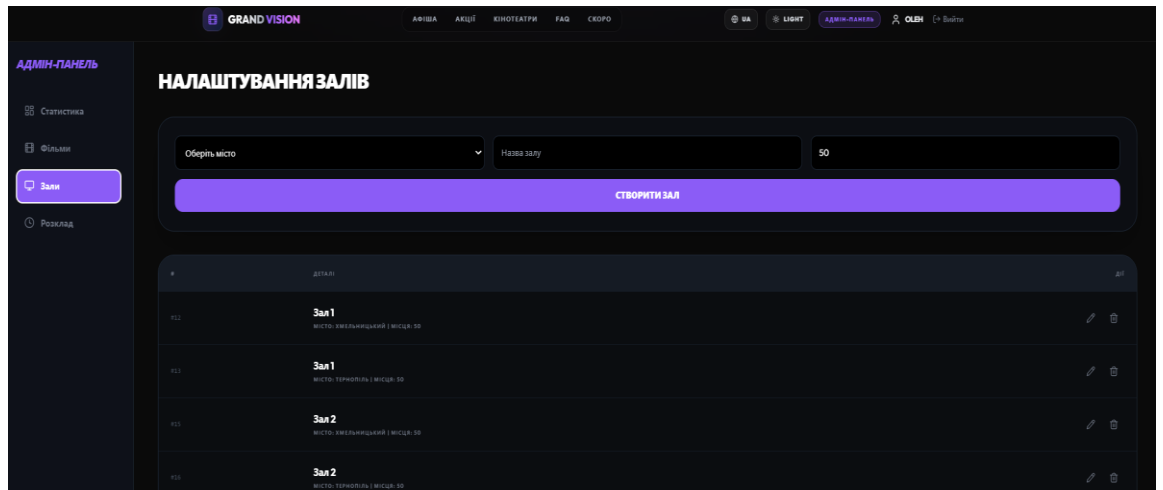


Рисунок 2.9 — Керування залами

Ключовим елементом адміністративної частини є модуль керування розкладом сеансів. У ньому адміністратор може прив'язати фільм до конкретного залу, встановити дату й час показу, формат сеансу та базову вартість квитка. Створені сеанси відображаються у вигляді окремих карток, що спрощує перегляд і подальше керування розкладом. Інтерфейс розкладу сеансів наведено на рисунку 2.10.

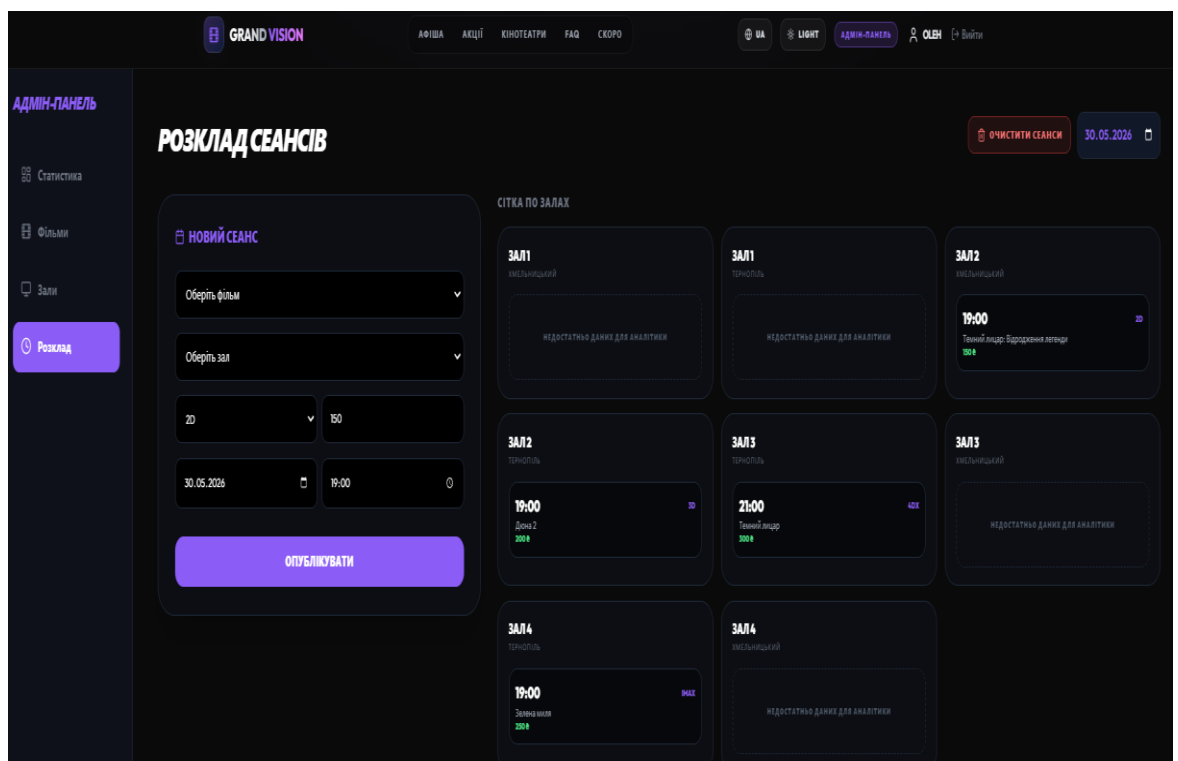


Рисунок 2.10 — Розклад сеансів

Отже, у процесі створення та верстки сторінок вебзастосунку було реалізовано повний набір інтерфейсів для користувацької та адміністративної частин системи. Сторінки мають єдиний стиль оформлення, логічну структуру та адаптивну верстку, що забезпечує зручну роботу з вебзастосунком на різних пристроях.

### 2.3 Розробка структури бази даних сайту

Серверна частина вебзастосунку «Grand Vision» використовує багаторівневу архітектуру з розділенням класів за функціональним призначенням. Структура backend-частини включає контролери, об'єкти передачі даних, сутності, репозиторії, сервіси та компоненти безпеки. Основними класами рівня збереження даних є JPA-сутності, які відображають структуру реляційних таблиць у базі даних PostgreSQL [7].

Проектування структури бази даних виконувалося з урахуванням основних інформаційних об'єктів предметної області: користувачів, фільмів, міст, кінозалів, сеансів, бронювань та відгуків. Такий набір сутностей забезпечує збереження даних, необхідних для роботи користувацької та адміністративної частин вебзастосунку. ER-діаграму бази даних вебзастосунку «Grand Vision» винесено до графічної частини 2026.КВР.122.421.04.00.00 БД [4].

Побудова структури бази даних здійснювалася з урахуванням взаємозв'язків між основними процесами вебзастосунку. Користувач взаємодіє з каталогом фільмів, обирає сеанс, виконує бронювання місця та отримує електронний квиток. Адміністратор, у свою чергу, керує фільмами, кінозалами та розкладом показів. Тому структура бази даних повинна забезпечувати збереження як публічної інформації для відображення на сторінках сайту, так і службових даних, необхідних для оформлення замовлень.

Кожна таблиця бази даних відповідає окремій сутності предметної області та має чітке призначення. Для забезпечення цілісності даних між таблицями використовуються первинні та зовнішні ключі. Це дозволяє пов'язати бронювання

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

з конкретним користувачем і сеансом, сеанс — із фільмом та залом, а зал — із відповідним містом. Такий підхід зменшує дублювання інформації та спрощує подальше оновлення даних через адміністративну панель.

Для збереження облікових записів користувачів використовується таблиця `users`. Вона містить дані, необхідні для авторизації користувача, визначення його ролі та прив'язки оформлених бронювань до конкретного облікового запису. Специфікацію сутності `users` наведено в таблиці 2.2.

Таблиця 2.2 — Специфікація сутності `users` (Користувачі)

Назва поля	Тип даних	Обмеження	Короткий опис
<code>id</code>	<code>bigint</code>	PK, Auto Increment, Not Null	Унікальний ідентифікатор користувача
<code>username</code>	<code>varchar(255)</code>	Not Null	Ім'я користувача в системі
<code>email</code>	<code>varchar(255)</code>	Not Null	Email користувача для авторизації та отримання квитків
<code>password</code>	<code>varchar(255)</code>	Not Null	Пароль користувача у захешованому вигляді
<code>is_admin</code>	<code>boolean</code>	Default false	Ознака адміністратора системи

Для збереження інформації про кінострічки використовується таблиця `movies`. Вона містить основні дані про фільм, які відображаються в афіші та на сторінці деталей фільму: назву, жанр, опис, посилання на постер, тривалість, рейтинг, трейлер і мову оригінального показу. Наявність окремих полів для медіаконтенту дозволяє зберігати посилання на постери та трейлери без дублювання самих файлів у базі даних. Поле `rating` використовується для відображення узагальненої оцінки фільму на основі користувацьких відгуків. Специфікацію сутності `movies` наведено в таблиці 2.3.

Таблиця 2.3 — Специфікація сутності *movies* (Фільми)

Назва поля	Тип даних	Обмеження	Короткий опис
id	bigint	PK, Auto Increment, Not Null	Унікальний ідентифікатор фільму
title	varchar(255)	Not Null	Назва кінострічки
genre	varchar(255)	Not Null	Жанр фільму
description	varchar(255)	Nullable	Текстовий опис сюжету фільму
image_url	varchar(255)	Nullable	Посилання на графічний файл постера
duration	integer	Not Null	Тривалість фільму у хвилинах
rating	double precision	Default 0.0	Середній рейтинг на основі відгуків
trailer_url	varchar(255)	Nullable	Посилання на офіційний трейлер (YouTube)
language	varchar(16)	Default 'en'	Код мови оригінального показу

Для територіального розподілу кінотеатрів використовується таблиця *cities*. Вона виконує роль довідника міст і дозволяє групувати кінозали та сеанси відповідно до обраної користувачем локації. Специфікацію сутності *cities* наведено в таблиці 2.4.

Таблиця 2.4 — Специфікація сутності *cities* (Міста)

Назва поля	Тип даних	Обмеження	Короткий опис
id	bigint	PK, Auto Increment, Not Null	Унікальний ідентифікатор міста
name	varchar(255)	Not Null	Назва міста, у якому розташований кінотеатр або кінозали

Інформація про кінозали зберігається в таблиці *halls*. Вона містить назву залу, його місткість та посилання на місто, у якому розташований зал. Поле *capacity* використовується для побудови схеми місць на сторінці бронювання. Специфікацію сутності *halls* наведено в таблиці 2.5.

Таблиця 2.5 — Специфікація сутності halls (Кінозали)

Назва поля	Тип даних	Обмеження	Короткий опис
id	bigint	PK, Auto Increment, Not Null	Унікальний ідентифікатор залу
name	varchar(255)	Not Null	Назва або номер кінозалу
capacity	integer	Not Null	Загальна кількість місць у залі
city_id	bigint	FK	Посилання на місто, у якому розташований зал

Для формування розкладу показів використовується таблиця sessions. Вона поєднує фільм, кінозал, дату й час початку сеансу, формат показу та базову вартість квитка. Саме ця сутність забезпечує зв'язок між репертуаром кінотеатру та доступними для бронювання сеансами. Специфікацію сутності sessions наведено в таблиці 2.6.

Таблиця 2.6 — Специфікація сутності sessions (Сеанси)

Назва поля	Тип даних	Обмеження	Короткий опис
id	bigint	PK, Auto Increment, Not Null	Унікальний ідентифікатор сеансу
movie_id	bigint	FK, Not Null	Зовнішній ключ, посилання на фільм
hall_id	bigint	FK, Not Null	Зовнішній ключ, посилання на кінозал
start_time	timestamp	Not Null	Точна дата та час початку сеансу
format	varchar(10)	Default '2D'	Формат показу (2D, 3D, IMAX, 4DX)
price	double precision	Not Null, Default 150.0	Базова вартість квитка на сеанс

Для фіксації оформлених замовлень, обраних місць та збереження пільгових

					<b>2026.КВР.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

статусів клієнтів використовується таблиця `bookings`, специфікацію якої наведено в таблиці 2.7. Комбінація полів `session_id` та `seat_number` є важливою для роботи бекенд-алгоритму запобігання подвійному бронюванню (колізіям) під час паралельних запитів користувачів у години пікових навантажень. У свою чергу, булевий прапорець `is_student` слугує тригером для сервісного шару системи, що використовується для відображення пільгового статусу та розрахунку кінцевої вартості замовлення в інтерфейсі клієнтської частини додатку. Наявність зовнішнього ключа `user_id` забезпечує жорстку прив'язку кожного проданого квитка до конкретного облікового запису, що дозволяє формувати детальну історію покупок в особистому кабінеті. Крім того, саме агреговані дані з цієї таблиці виступають технічною основою для програмної генерації фінального PDF-квитка з унікальним QR-кодом після успішної оплати.

Таблиця 2.7 — Специфікація сутності `bookings` (Бронювання)

Назва поля	Тип даних	Обмеження	Короткий опис
<code>id</code>	<code>bigint</code>	PK, Auto Increment, Not Null	Унікальний ідентифікатор замовлення
<code>seat_number</code>	<code>integer</code>	Not Null, UNIQUE	Унікальний ідентифікатор заброньованого квитка
<code>is_student</code>	<code>boolean</code>	Not Null	Ознака застосування пільгового тарифу
<code>session_id</code>	<code>bigint</code>	FK	Зовнішній ключ, посилання на сеанс
<code>user_id</code>	<code>bigint</code>	FK	Зовнішній ключ, посилання на покупця

Для реалізації соціально-інтерактивного функціоналу та збору зворотного зв'язку від відвідувачів кінотеатру використовується сутність `reviews`, опис полів

якої наведено в таблиці 2.8. Зв'язки із таблицями користувачів (`user_id`) та фільмів (`movie_id`) гарантують цілісність даних на рівні СКБД PostgreSQL, запобігаючи створенню відгуків на неіснуючі об'єкти. Числове поле `stars` використовується для збереження оцінки користувача в діапазоні від 1 до 10 і є основним джерелом даних, на основі яких серверна логіка автоматично перераховує та оновлює глобальний рейтинг кінострічки в таблиці `movies` після кожного успішного додавання нового коментаря.

Таблиця 2.8 — Специфікація сутності `reviews` (Відгуки)

Назва поля	Тип даних	Обмеження	Короткий опис
<code>id</code>	<code>bigint</code>	PK, Auto Increment, Not Null	Унікальний ідентифікатор відгуку
<code>user_id</code>	<code>bigint</code>	FK, Not Null	Посилання на автора коментаря
<code>movie_id</code>	<code>bigint</code>	FK, Not Null	Посилання на об'єкт рецензування (фільм)
<code>stars</code>	<code>integer</code>	Not Null	Оцінка фільму від 1 до 10 зірок
<code>comment</code>	<code>varchar(1000)</code>	Nullable	Текстовий коментар користувача

## 2.4 Програмування сайту

Програмування вебзастосунку «Grand Vision» виконувалося з розділенням клієнтської та серверної частин. Серверна частина реалізована на базі Java Spring Boot [6] і містить контролери, сервіси, репозиторії, DTO-класи, сутності бази даних та компоненти безпеки. Клієнтська частина створена з використанням React [5], Vite [8] та Tailwind CSS [9] і містить сторінки, компоненти інтерфейсу, допоміжні модулі та сервіси для взаємодії з REST API.

Файлова структура проєкту побудована за модульним принципом. У backend-частині окремо виділено пакети `controller`, `service`, `entity`, `repository`, `dto` та `security`, що спрощує супровід коду та відокремлює бізнес-логіку від рівня представлення даних. У frontend-частині код розподілено між сторінками, компонентами, сервісами, хуками та допоміжними модулями. Деревоподібну

структуру вихідного коду проєкту наведено на рисунку 2.11.

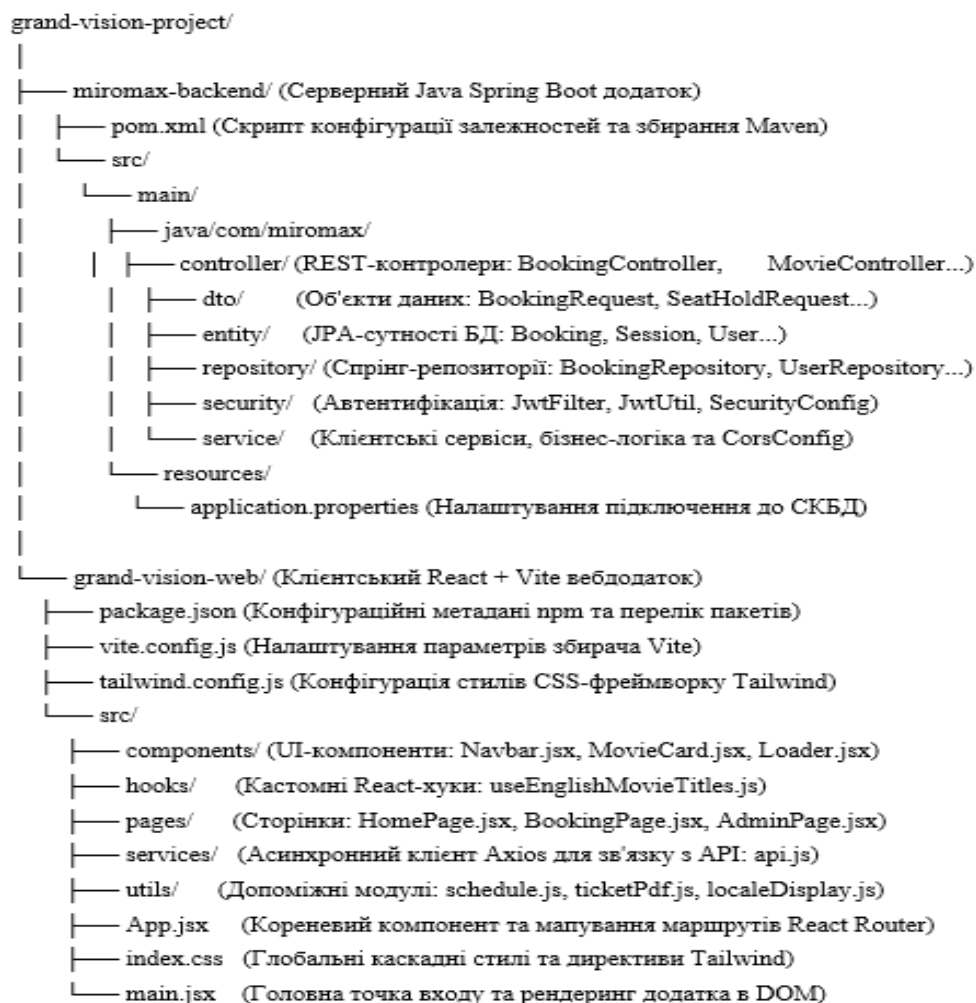


Рисунок 2.11 — Деревоподібна структура вихідного коду проєкту

Одним із ключових модулів серверної частини є модуль бронювання квитків. Він забезпечує тимчасове утримання місця, перевірку доступності крісла, створення бронювання, прив'язку замовлення до користувача та збереження інформації про придбаний квиток у базі даних. У реалізації цього модуля задіяно контролер `BookingController`, сервіс `BookingService`, репозиторій `BookingRepository`, DTO-клас `BookingRequest` та сутність `Booking`.

Для візуального представлення зв'язків між основними класами модуля бронювання було побудовано UML-діаграму. Вона демонструє взаємодію контролера, сервісного шару, репозиторію, DTO-об'єктів та сутностей бази даних. UML-діаграму класів модуля бронювання наведено на рисунку 2.12.

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

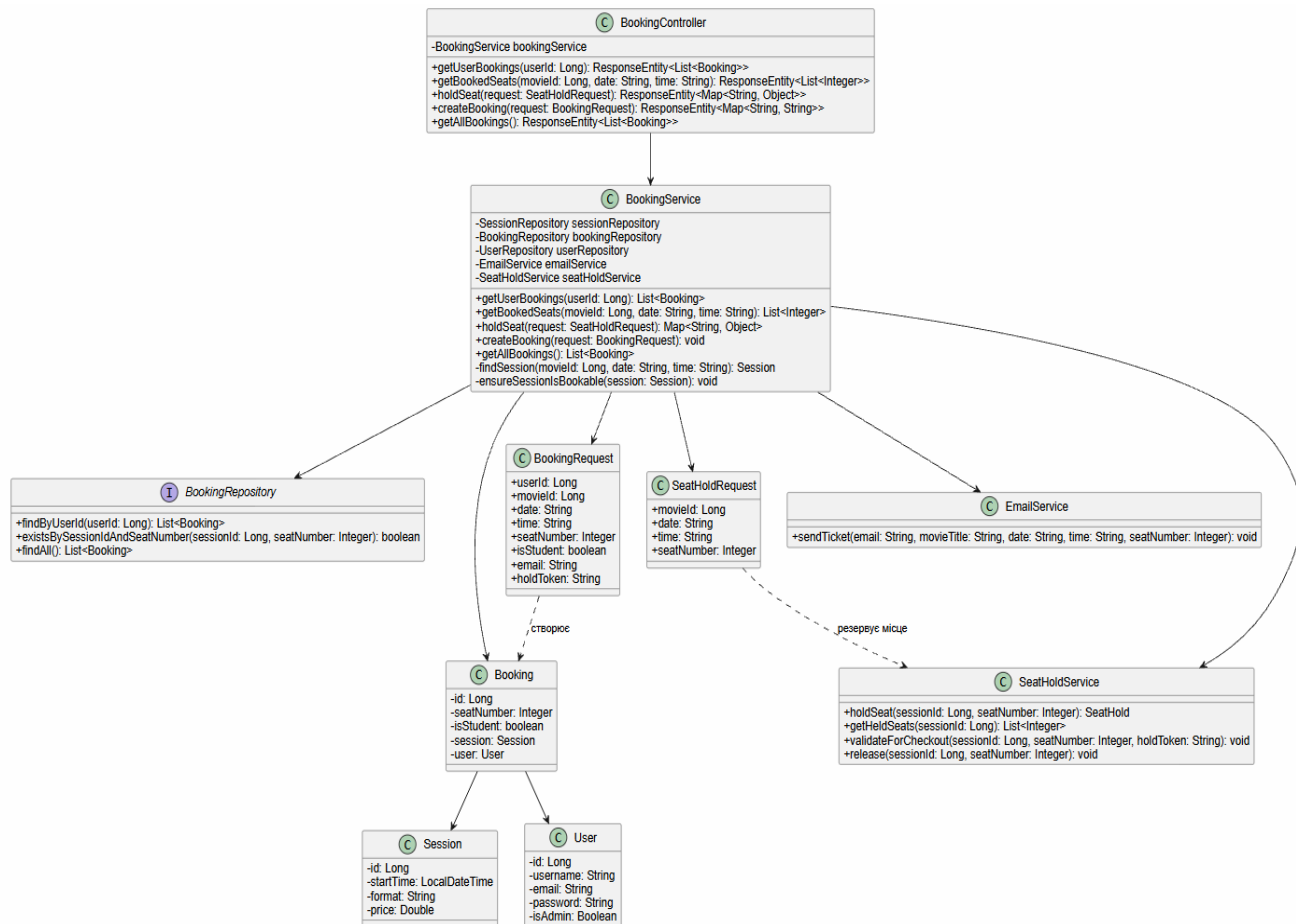


Рисунок 2.12 — UML-діаграма класів модуля бронювання

Взаємодія між клієнтською частиною та сервером під час оформлення квитка реалізована через набір REST-запитів. Спочатку користувач обирає місце в кінозалі, після чого клієнтська частина надсилає запит на тимчасове утримання місця. У відповідь сервер повертає `holdToken`, який використовується під час фінального підтвердження бронювання.

Основна логіка оформлення квитка виконується під час надсилання POST-запиту на створення бронювання. Сервер перевіряє коректність переданих даних, наявність активного утримання місця, доступність сеансу та відсутність повторного бронювання того самого місця. Після успішної перевірки створюється запис у таблиці `bookings`, який пов'язує користувача з конкретним сеансом і вибраним місцем. Такий підхід забезпечує узгодженість даних між інтерфейсом бронювання, особистим кабінетом користувача та адміністративною частиною системи. Алгоритм оформлення бронювання наведено на рисунку 2.13.

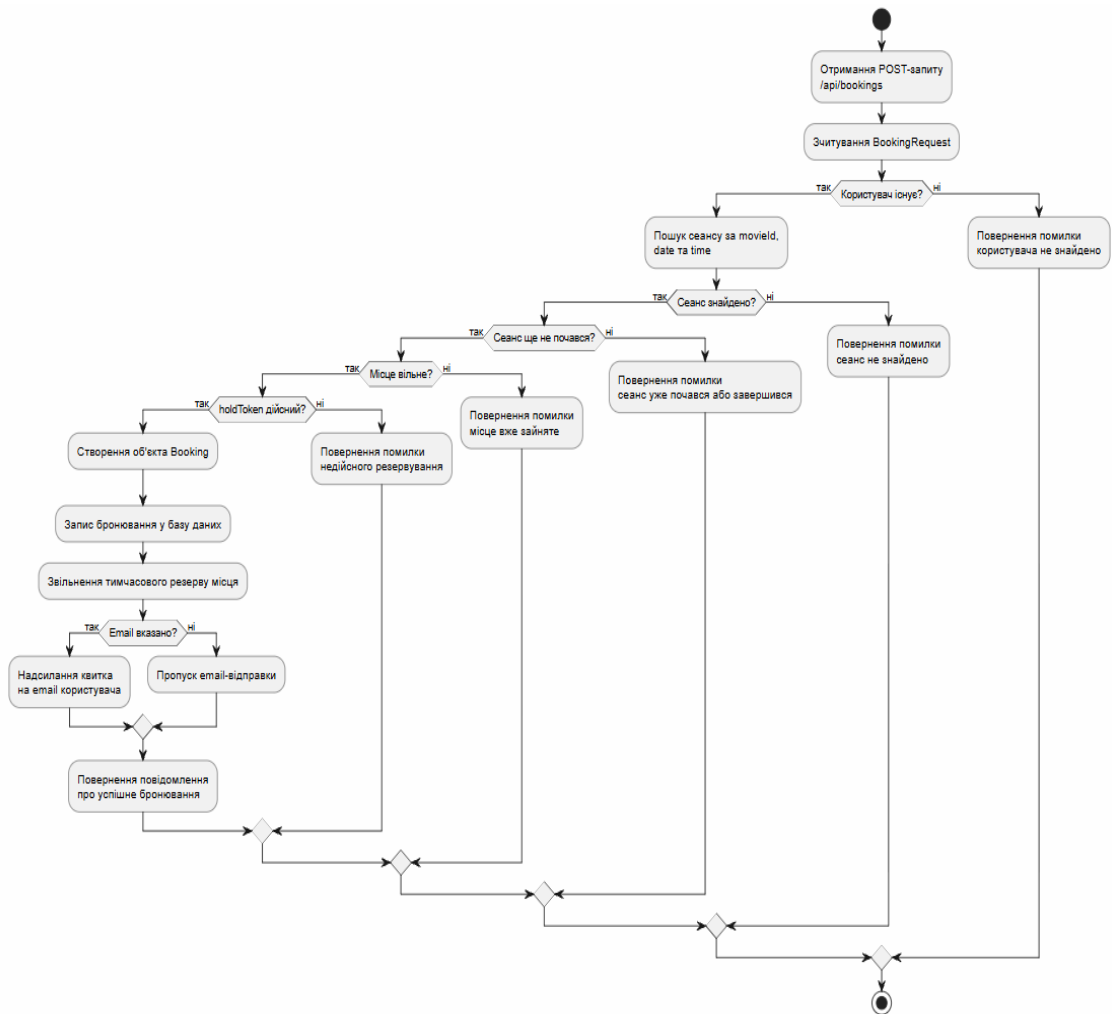


Рисунок 2.13 — алгоритм оформлення бронювання

Повна програмна реалізація класу BookingController з відповідними REST-ендпоінтами наведена у додатку А. У тексті пояснювальної записки подано опис призначення основних методів контролера та специфікацію REST API, що використовується під час оформлення бронювання.

Контролер BookingController виконує роль проміжної ланки між клієнтською частиною та сервісним рівнем серверного застосунку. Він приймає HTTP-запити від інтерфейсу користувача, передає отримані дані до BookingService та повертає клієнтській частині результат виконання операції. Завдяки такому підходу логіка обробки бронювання не змішується з логікою приймання запитів, що відповідає принципу розділення відповідальності.

Під час оформлення квитка контролер забезпечує кілька послідовних операцій: отримання списку зайнятих місць для вибраного сеансу, тимчасове

утримання місця, створення фінального бронювання та отримання історії замовлень користувача. Окремі методи також використовуються адміністративною частиною для перегляду всіх бронювань у системі. Специфікацію REST API ендпоінтів контролера BookingController наведено в таблиці 2.9.

Таблиця 2.9 — Специфікація REST API ендпоінтів контролера BookingController

HTTP-метод	URL-маршрут	Вхідні параметри	Призначення	Структура відповіді
GET	/api/bookings/user/{userId}	JWT-токен	Історія квитків користувача	Список об'єктів Booking
GET	/api/bookings	Query-параметри: movieId, date, time	Зайняті місця сеансу	Масив цілих чисел (номерів місць), наприклад: [1, 5, 12]
POST	/api/bookings/holds	JSON: movieId, date, time, seatNumber	Утримання місця	JSON об'єкт із holdToken та expiresAt
POST	/api/bookings	JSON: userId, movieId, date, time, seatNumber, isStudent, email, holdToken	Створення бронювання	Текстове повідомлення про успішне бронювання місця
GET	/api/bookings/all	Без параметрів	Усі бронювання для адмінпанелі	Список усіх об'єктів Booking

Наведена специфікація REST API демонструє основні точки взаємодії клієнтської частини з серверним модулем бронювання. Кожен ендпоінт виконує

окрему функцію в межах загального процесу оформлення квитка: отримання історії замовлень, перевірку зайнятих місць, тимчасове утримання вибраного місця та остаточне створення бронювання.

Особливу роль у роботі контролера відіграє механізм тимчасового блокування місця, оскільки він запобігає одночасному бронюванню одного й того самого крісла кількома користувачами. Після вибору місця клієнтська частина отримує `holdToken`, який передається під час фінального підтвердження замовлення.

Для підтвердження практичної реалізації REST-взаємодії між клієнтською та серверною частинами було розроблено стандартизовані JSON-структури для обміну даними. Приклади таких запитів та відповідей сервера наведено в додатку Б. Наведені структури демонструють формат запитів на створення фільмів та сеансів, а також формат відповідей сервера, що містять актуальний контент та службову інформацію про пагінацію.

Під час проєктування JSON-об'єктів для створення фільму особлива увага приділялася обов'язковим полям, таким як `title` та `genre`, які використовуються для формування афіші та клієнтських фільтрів. Поле `duration` визначає тривалість фільму для розрахунку розкладу, а поля `imageUrl` та `trailerUrl` забезпечують коректне відображення медіаконтенту. При створенні сеансів параметри `date` та `time` об'єднуються на серверній стороні у поле `startTime` типу `LocalDateTime`, що гарантує цілісність даних у базі.

Структура відповідей сервера, що використовується для динамічного відображення каталогу, включає масив об'єктів `content` та службові дані пагінації (`totalElements`, `totalPages`), що дозволяє фронтенд-додатку ефективно керувати навігацією. Запропонована структура JSON-відповідей забезпечує високу швидкість парсингу на стороні клієнта та мінімізує обсяг переданих даних, що є критично важливим для швидкодії вебзастосунку. Такий підхід до організації API-інтерфейсу дозволяє легко масштабувати проєкт та впроваджувати нові типи медіаконтенту без порушення цілісності базових схем обміну даними.

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

## 2.4.1 Написання клієнтської частини

Програмування клієнтської частини (frontend) web-сайту кінотеатру виконано за допомогою сучасної JavaScript-бібліотеки React. Вибір саме цієї технології зумовлений її високою продуктивністю завдяки використанню механізму віртуального DOM (Virtual DOM), що дозволяє мінімізувати кількість прямих, ресурсомістких звернень до браузерного API та суттєво пришвидшити рендеринг користувацького інтерфейсу. У зв'язці з React було використано новітній інструмент збирання Vite, який, на відміну від класичних бандлерів (наприклад, Webpack), базується на нативних ES-модулях. Це забезпечує надзвичайно високу швидкість гарячого перезавантаження модулів (Hot Module Replacement – HMR) під час розробки, що значно оптимізує робочий процес програміста та зводить до мінімуму час очікування при внесенні змін до вихідного коду.

Архітектура застосунку суворо базується на компонентному підході. Кожен логічний елемент інтерфейсу (наприклад, картка фільму, навігаційне меню, форма авторизації, інтерактивна схема залу для вибору місць) виділений в окремий, незалежний та інкапсульований компонент. Це не лише сприяє максимальному повторному використанню коду згідно з принципом DRY (Don't Repeat Yourself), але й значно полегшує подальшу підтримку, тестування та масштабування всієї системи.

Для взаємодії із серверним API та забезпечення динамічного оновлення контенту реалізовано систему асинхронних HTTP-запитів. Клієнтський додаток ефективно обробляє повний життєвий цикл мережевих запитів: ініціалізацію з'єднання для отримання каталогу фільмів чи розкладу сеансів, безпечну передачу даних форм бронювання, відображення індикаторів завантаження (спінерів) для покращення UX (User Experience) та надійну обробку можливих серверних помилок.

Локальний стан компонентів керується за допомогою стандартизованих хуків React, зокрема useState для збереження та мутації динамічних даних у пам'яті браузера, та useEffect для виконання побічних ефектів, таких як первинне

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

завантаження даних під час монтування компонента у DOM-дерево.

Для маршрутизації та навігації між сторінками застосовано спеціалізовану бібліотеку React Router DOM, що дозволяє повною мірою реалізувати концепцію Single Page Application (SPA – односторінковий застосунок). Завдяки цьому архітектурному рішенню переходи між різними розділами сайту (від головної сторінки до детального опису фільму чи кошика) відбуваються миттєво, без повного перезавантаження сторінки браузера. Окрім того, система маршрутизації включає механізми захисту шляхів (Protected Routes), які надійно обмежують доступ до адміністративних панелей для звичайних відвідувачів, автоматично перенаправляючи їх на сторінку авторизації у разі спроби несанкціонованого входу.

#### 2.4.2 Написання admin- частини

Розробка адміністративної частини системи реалізована як інтегрований модуль загальної архітектури застосунку з окремим інтерфейсом керування контентом. Основне обмеження доступу до адміністративної панелі здійснюється на рівні клієнтської маршрутизації: інтерфейс перевіряє роль поточного користувача та приховує адміністративні сторінки від осіб без відповідних привілеїв.

Доступ до адміністративної панелі обмежується насамперед на рівні клієнтської маршрутизації (frontend) за допомогою захищених маршрутів (наприклад, компонент AdminRoute), які враховують поточну роль користувача. Зі свого боку, JWT-токени [10] використовуються для надійної ідентифікації авторизованого користувача при взаємодії з REST API бекенду. Така архітектура дозволяє гнучко керувати правами доступу до інтерфейсу управління репертуаром та розкладом.

До основного функціоналу admin-частини належить комплексне управління всім інформаційним наповненням (контентом) web-сайту кінотеатру. Адміністратор системи отримує доступ до розширених електронних форм (CRUD-

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

операцій) для додавання нових кінострічок до репертуару. Цей процес включає заповнення метаданих фільму, завантаження графічних постерів, визначення жанрової приналежності, тривалості показу, трейлеру.

Окремим, технологічно складним блоком є модуль управління розкладом показів. Він надає адміністратору інструментарій для створення нових сеансів, їх логічної прив'язки до конкретних фізичних кінозалів із врахуванням їх місткості та топології посадкових місць. У цьому ж модулі виконується встановлення дат, часу початку та конфігурація базової вартості квитків залежно від формату показу (ранковий чи вечірній сеанс, формат 2D, 3D або IMAX).

Усі зміни репертуару, налаштування розкладу та цінової політики, які вносяться через адміністративну панель, миттєво зберігаються у реляційній базі даних PostgreSQL. Процес збереження відбувається з суворим дотриманням принципів транзакційної цілісності (ACID), що гарантує відсутність частково збережених або пошкоджених даних у разі програмних збоїв. Завдяки реактивній природі клієнтського застосунку, оновлена інформація відразу стає доступною для перегляду та бронювання клієнтами на публічній частині web-сайту без необхідності ручного кешування чи перезапуску сервера.

## 2.5 Тестування web- сайту

Тестування вебзастосунку «Grand Vision» виконувалося з метою перевірки коректності роботи основних функціональних сценаріїв, стабільності взаємодії клієнтської та серверної частин, а також правильності збереження даних у базі даних. Перевірка проводилася шляхом ручного функціонального тестування інтерфейсу користувача, перевірки REST-запитів до серверної частини та аналізу результатів виконання операцій у базі даних PostgreSQL.

Основну увагу під час тестування було приділено сценаріям, які є ключовими для роботи кінотеатру: перегляду афіші, вибору фільму, вибору сеансу, бронюванню місця, імітації оплати, формуванню електронного квитка та відображенню замовлення в особистому кабінеті користувача. Окремо

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

перевірялася адміністративна частина системи, зокрема додавання фільмів, створення залів, формування розкладу сеансів і перегляд статистики.

На першому етапі було перевірено роботу сторінки авторизації та реєстрації користувача. Під час введення облікових даних система виконує перевірку коректності заповнення полів, після чого користувач отримує доступ до особистого кабінету та функцій бронювання. Інтерфейс авторизації користувача наведено на рисунку 2.14.

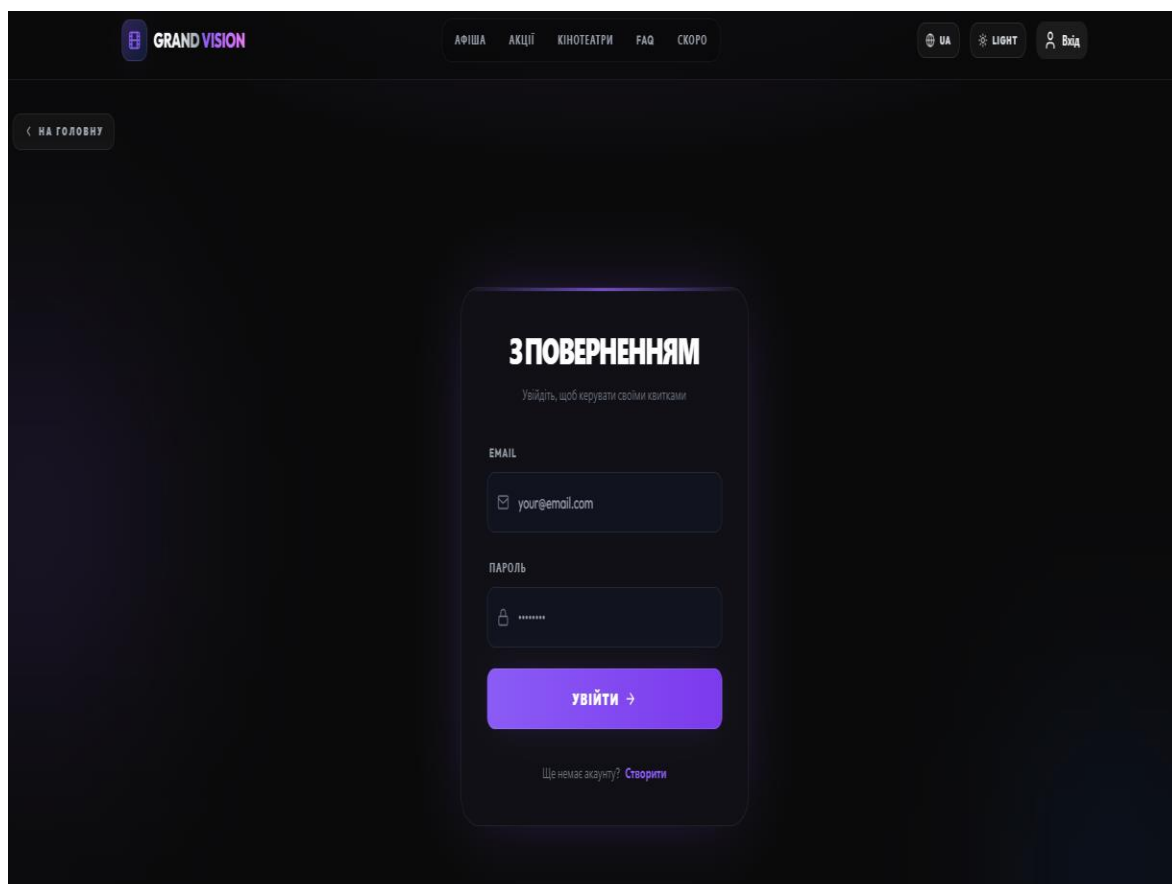


Рисунок 2.14 — Інтерфейс авторизації користувача

Наступним етапом тестування була перевірка сценарію вибору сеансу та місця в кінозалі. Після переходу зі сторінки фільму користувач обирає доступний сеанс, після чого система відображає карту залу з місцями. Перевірка показала, що після вибору місця воно візуально виділяється в інтерфейсі, а в нижньому блоці сторінки відображається номер обраного місця та сума до сплати. Результат вибору місця наведено на рисунку 2.15.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

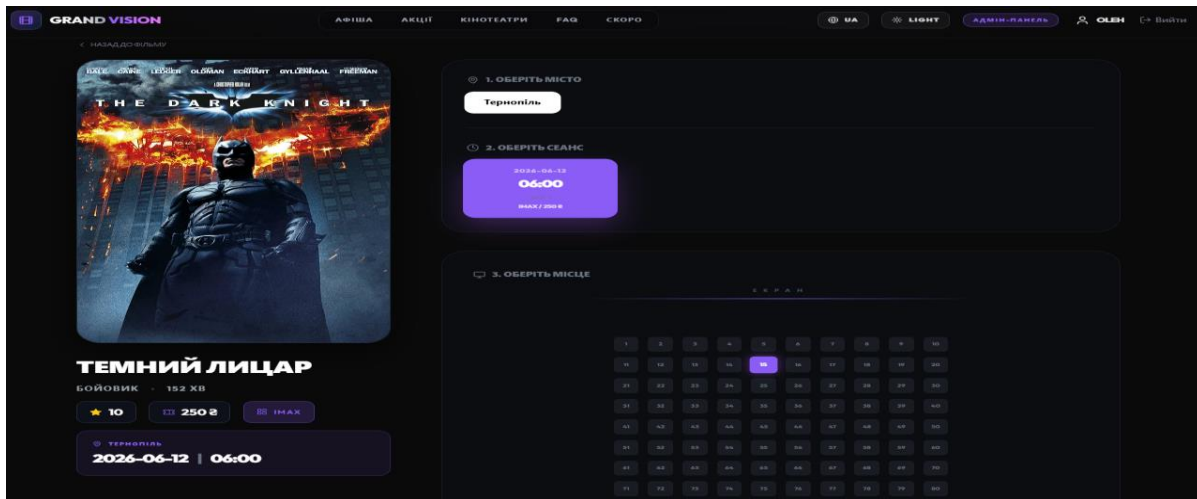


Рисунок 2.15 — Перевірка вибору місця в кінозалі

Окремо перевірено роботу механізму перерахунку вартості квитка при застосуванні пільгової категорії. У разі вибору відповідного параметра система змінює фінальну суму замовлення без перезавантаження сторінки. Такий підхід підтверджує коректну роботу клієнтської логіки розрахунку вартості квитка. Результат перерахунку вартості при застосуванні студентської знижки наведено на рисунку 2.16.

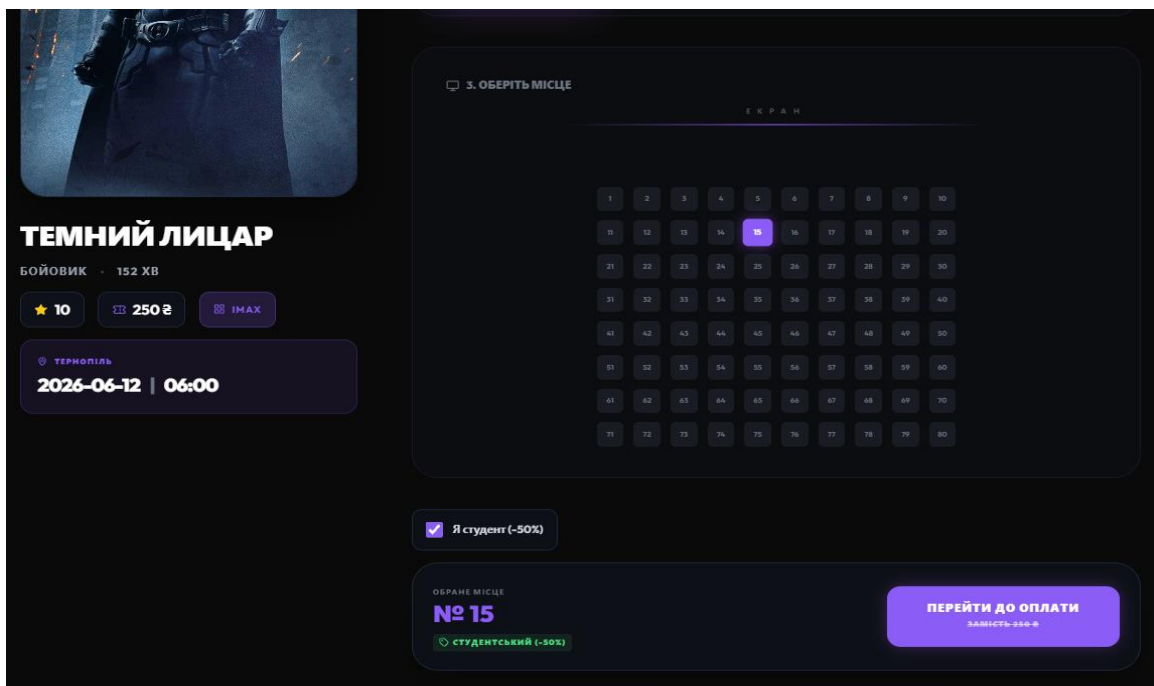


Рисунок 2.16 — Перерахунок вартості при застосуванні студентської знижки

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

Після підтвердження вибору місця користувач переходить до модального вікна оформлення замовлення. У цьому вікні вводиться електронна пошта для отримання квитка та реквізити платіжної картки для імітації платіжного етапу. Під час тестування перевірено коректність відображення суми до сплати, номера місця та роботи полів введення. Інтерфейс модального вікна оформлення бронювання наведено на рисунку 2.17.

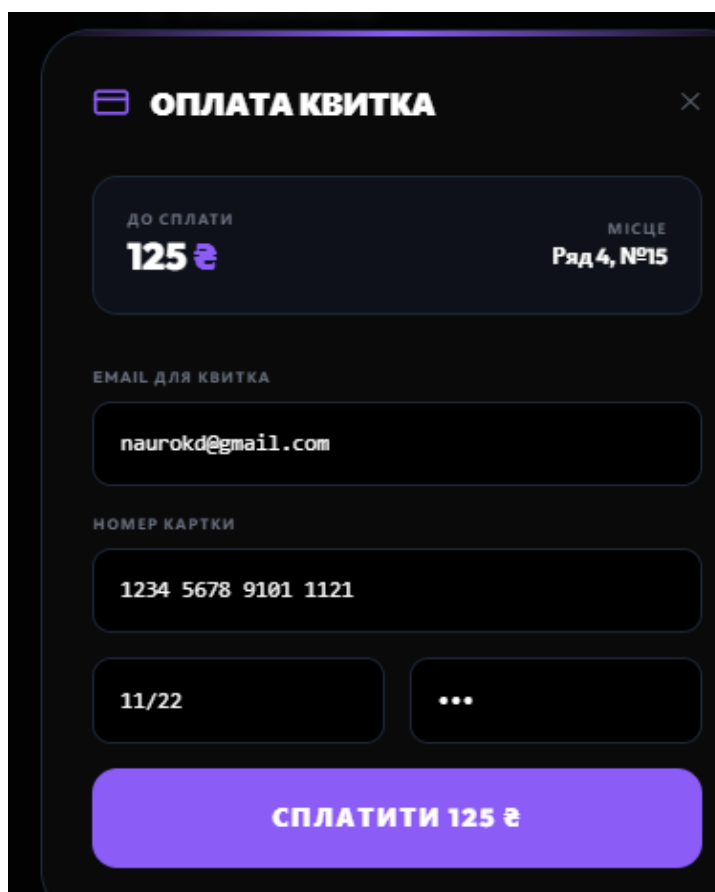


Рисунок 2.17 — Модальне вікно оформлення бронювання квитка

Після підтвердження бронювання було перевірено збереження електронного квитка в особистому кабінеті користувача. У профілі відображається картка замовлення з назвою фільму, датою та часом сеансу, номером залу, обраним місцем, статусом оплати та QR-кодом. Також передбачено можливість завантаження PDF-квитка, що підтверджує завершення процесу оформлення замовлення. Результат збереження електронного квитка в особистому кабінеті наведено на рисунку 2.18.

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

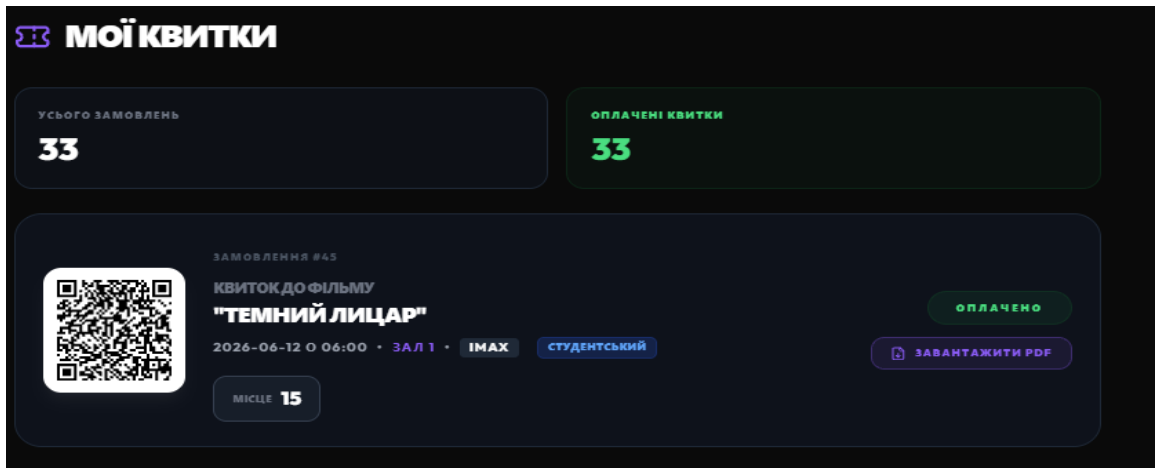


Рисунок 2.18 — Перевірка збереження електронного квитка в особистому кабінеті

Додатково було перевірено можливість завантаження електронного квитка у форматі PDF. Після натискання кнопки «Завантажити PDF» система формує окремий файл квитка, який містить основні дані замовлення: назву фільму, дату й час сеансу, номер місця та QR-код для перевірки квитка. Такий формат є зручним для збереження на пристрої користувача або подальшого друку. Згенерований PDF-квиток з QR-кодом наведено на рисунку 2.19.



Рисунок 2.19 — Згенерований PDF-квиток

Для перевірки адміністративного модуля керування фільмами було виконано тестове додавання нової кінострічки. На першому етапі адміністратор заповнює форму додавання фільму, вказуючи назву, жанр, опис, тривалість, рейтинг, мову, посилання на постер і трейлер. Заповнення форми додавання фільму наведено на рисунку 2.20.

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

**КЕРУВАННЯ ФІЛЬМАМИ**

Пірати Карибського моря: Прокляття «Чорної перлини» Пригоди

143 Мова оригіналу: англійська

[https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Pirates\\_of\\_the\\_Caribbean\\_movie.jpg](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Pirates_of_the_Caribbean_movie.jpg)

[https://www.youtube.com/watch?v=sq\\_J3vFXEQ](https://www.youtube.com/watch?v=sq_J3vFXEQ)

американський фантастичний пригодницький фільм 2003 року. Режисер – Гор Вербінокі. Перший із серії фільмів «Пірати Карибського моря».

**ДОДАТИ ФІЛЬМ**

Рисунок 2.20 — Заповнення форми додавання фільму

Після натискання кнопки збереження система передає введені дані на серверну частину, де створюється новий запис у базі даних. У разі успішного збереження фільм з'являється в адміністративному списку, що підтверджує коректність виконання операції додавання. Результат відображення доданого фільму наведено на рисунку 2.21.











#10	<b>Титанік</b> РОМАНТИКА	 
#11	<b>Початок (Insertion)</b> НАТЮРОВА ФАНТАСТИКА	 
#12	<b>Темний лицар: Відродження легенди</b> БОЙОВИК	 
#13	<b>Бетмен: Початок</b> БОЙОВИК	 
#14	<b>Пірати Карибського моря: Прокляття «Чорної перлини»</b> ПРИГОДИ	 

Рисунок 2.21 — Відображення доданого фільму в адміністративному списку

Після перевірки створення нового фільму було протестовано можливість редагування його даних. Адміністратор відкриває запис фільму зі списку, змінює необхідні поля та зберігає оновлену інформацію. Це дозволяє підтримувати актуальність репертуару без прямого втручання в базу даних. Перевірку редагування фільму наведено на рисунку 2.22.

					<b>2026.КВР.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

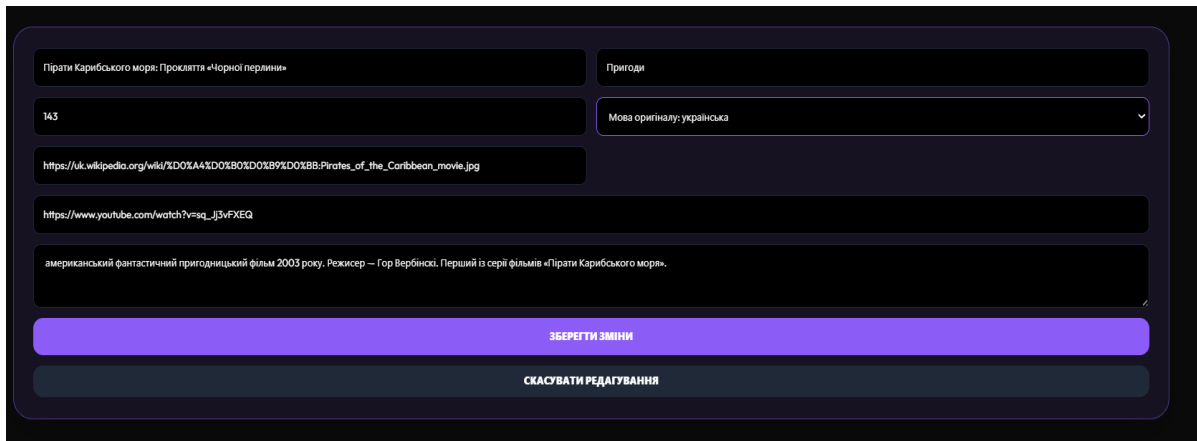


Рисунок 2.22 — Перевірка редагування даних фільму

Також було перевірено можливість видалення фільму з адміністративної панелі. Після натискання відповідної кнопки запис вилучається зі списку фільмів, що підтверджує коректну роботу функції видалення. Перевірку видалення фільму наведено на рисунку 2.23.

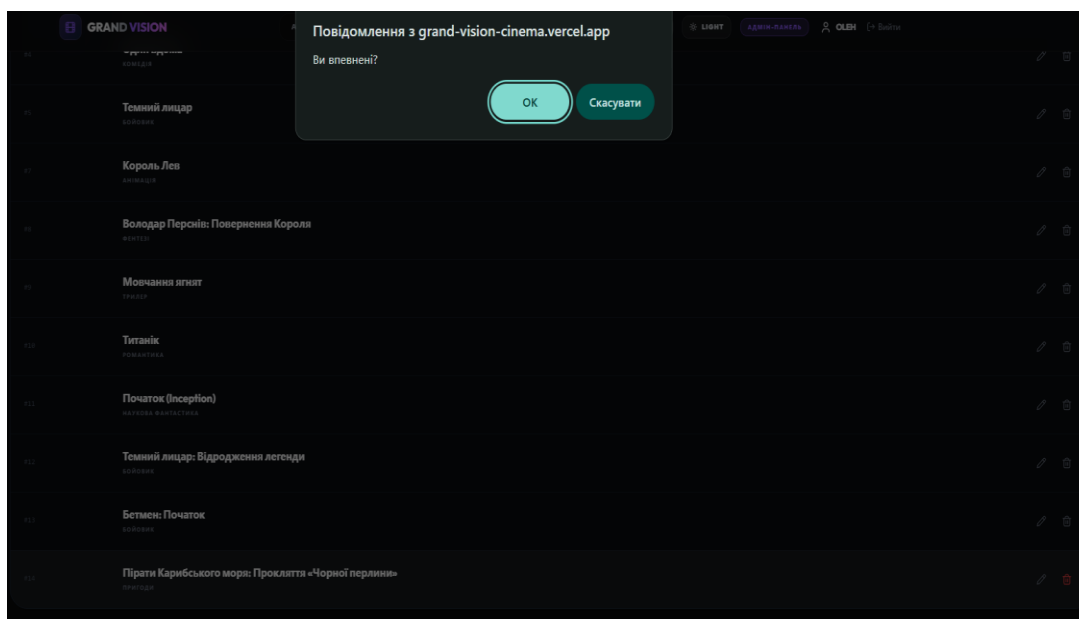


Рисунок 2.23 — Перевірка видалення фільму

Після створення фільму та налаштування відповідного сеансу було перевірено публікування сеансу в адміністративній панелі. Адміністратор вказує фільм, зал, дату, час показу, формат і вартість квитка, після чого сеанс додається до загального розкладу. Перевірку публікування сеансу наведено на рисунку 2.24.

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

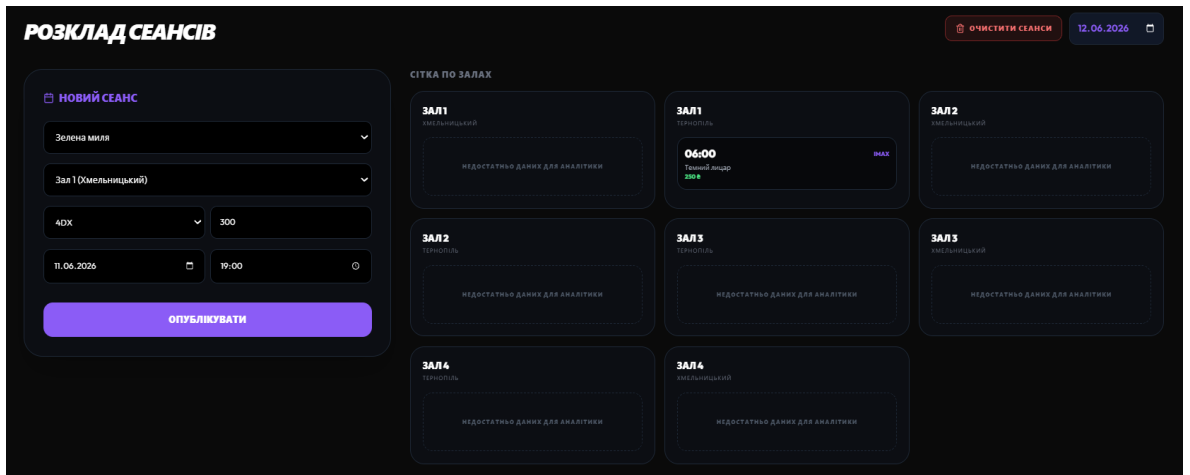


Рисунок 2.24 — Перевірка публікування сеансу в адміністративній панелі

Для перевірки зв'язку адміністративної та користувацької частин було відкрито афішу вебзастосунку. Доданий фільм відображається серед доступних кінострічок, що підтверджує коректну передачу даних із серверної частини до клієнтського інтерфейсу. Перевірку відображення фільму в афіші наведено на рисунку 2.25.

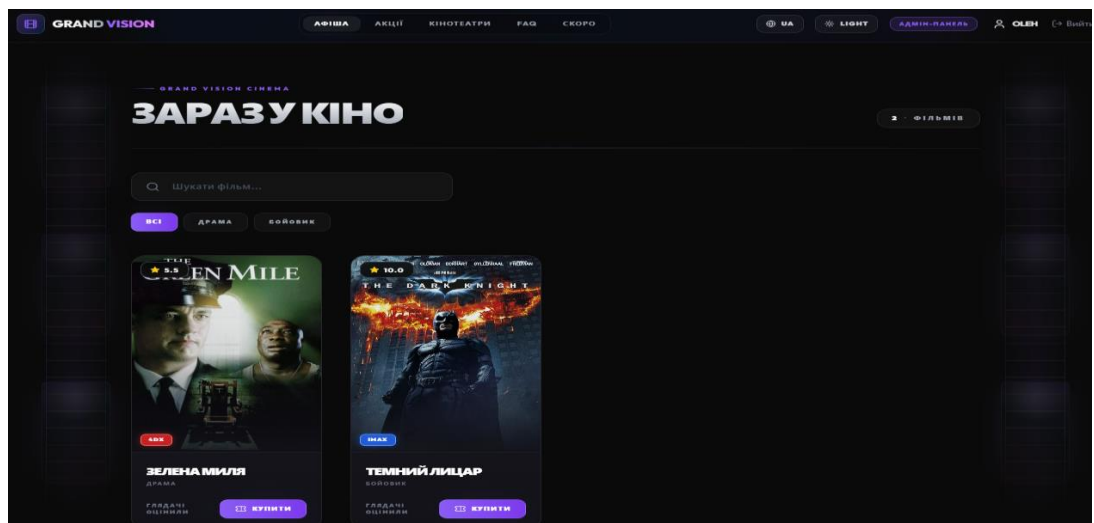


Рисунок 2.25 — Перевірка відображення доданого фільму в афіші

Наступним етапом було перевірено роботу пошуку за назвою фільму. У рядок пошуку введено назву доданої кінострічки, після чого система відобразила відповідну картку фільму. Це підтверджує коректну роботу механізму пошуку в каталозі. Перевірку пошуку фільму за назвою наведено на рисунку 2.26..

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

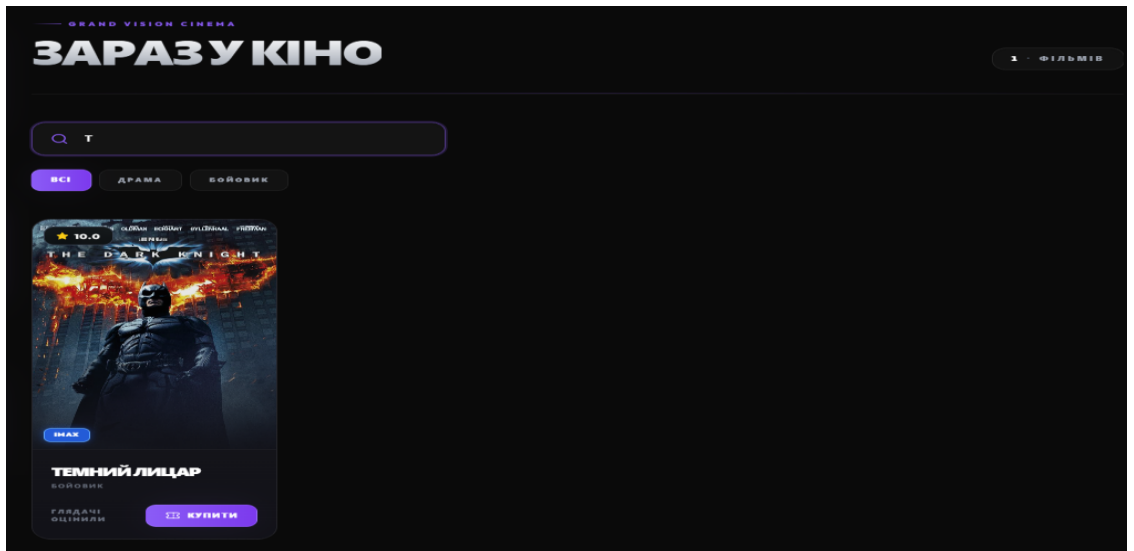


Рисунок 2.26 — Перевірка пошуку фільму за назвою

Також було перевірено роботу фільтрації фільмів за жанрами. Після вибору відповідної категорії в каталозі залишаються лише ті кінострічки, які відповідають обраному жанру. Це спрощує навігацію користувача в афіші та дає змогу швидше знайти потрібний фільм. Перевірку фільтрації фільмів за жанром наведено на рисунку 2.27.

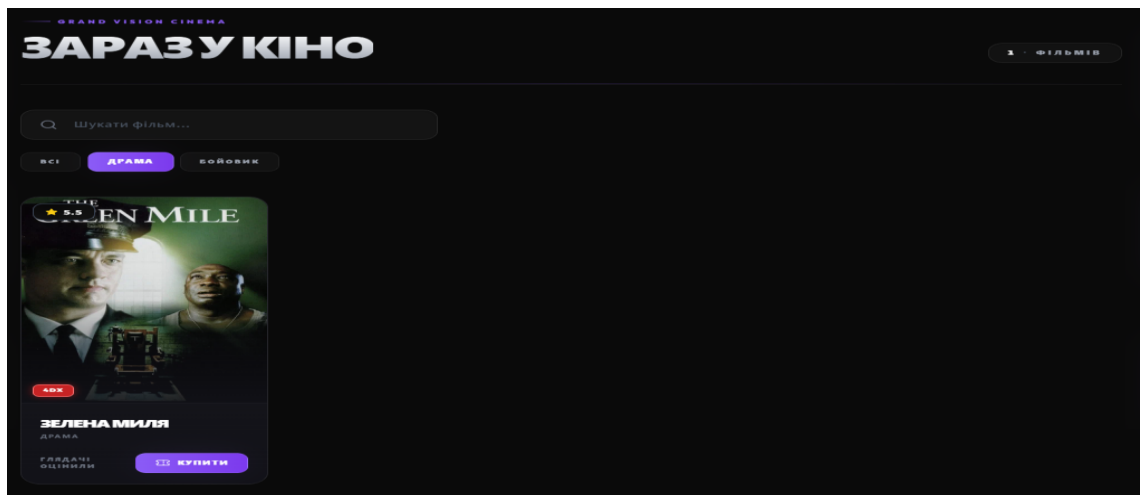


Рисунок 2.27 — Перевірка фільтрації фільмів за жанром

Інтеграційна перевірка взаємодії між клієнтською та серверною частинами виконувалася через надсилання HTTP-запитів до REST API. Для перевірки створення бронювання було сформовано POST-запит із даними користувача,

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

фільму, сеансу, обраного місця, ознаки студентського квитка та токена тимчасового резервування. Серверна частина обробляє запит через BookingController, передає дані до сервісного рівня та створює запис про бронювання в базі даних.

Під час тестування frontend-частини перевірено відображення карти кінозалу, вибір місця, перерахунок вартості, відкриття модального вікна оформлення квитка, збереження електронного квитка в особистому кабінеті та формування PDF-квитка з QR-кодом. Також перевірено адміністративні функції: додавання, редагування та видалення фільму, публікування сеансу й відображення доданих даних у користувацькій частині вебзастосунку.

Результати перевірок за ключовими функціональними сценаріями наведено у таблиці 2.10.

Таблиця 2.10 — Результати тестування функціональних сценаріїв

Сценарій тестування	Очікуваний результат	Статус
Реєстрація бронювання	Статус 200 ОК, запис у БД	Успішно
Повторне бронювання місця	Помилка 400/500, повідомлення	Успішно
Запит доступних місць	Повернення масиву Integer	Успішно
Валідація email користувача	Перевірка формату, повідомлення	Успішно

Проведене тестування підтвердило коректну роботу основних сценаріїв вебзастосунку: перегляду афіші, вибору місця, оформлення бронювання, збереження квитка та керування даними через адміністративну панель. Клієнтська частина стабільно взаємодіє із серверним API, а введені дані коректно зберігаються в базі даних PostgreSQL. Це свідчить про готовність програмного комплексу до демонстрації та подальшого використання.

### 3. СПЕЦІАЛЬНИЙ РОЗДІЛ

#### 3.1 Інструкція розміщення сайту в Інтернеті

Для забезпечення високої доступності, відмовостійкості та мінімізації фінансових витрат розгортання програмного комплексу «Grand Vision» реалізовано з використанням сучасних хмарних PaaS-платформ. Такий підхід дозволяє імплементувати повноцінний конвеєр безперервної інтеграції та розгортання з автоматичною синхронізацією коду з репозиторію GitHub. Вибір хмарної архітектури замість традиційних VPS дозволяє автоматично делегувати завдання з обслуговування інфраструктури провайдеру, зосередившись виключно на оптимізації програмного коду.

Зберігання реляційних даних забезпечується хмарною платформою Neon [11] (Serverless Postgres), яка надає безкоштовний рівень доступу та підтримує автоматичне масштабування залежно від поточного навантаження. Процес налаштування передбачає реєстрацію облікового запису на платформі Neon.tech та створення нового проєкту. Після генерації бази даних система надає рядок підключення (Connection String), що містить адресу хоста, логін та пароль. Отриманий рядок підключення необхідно зберегти як змінну середовища для серверного додатка, що гарантує приховування конфіденційних даних від відкритого вихідного коду. Додаткова перевага такого підходу полягає у можливості створення ізольованих гілок бази даних для кожного середовища (development, staging, production), що суттєво спрощує процес тестування нових функцій.

Хостинг серверної частини на Java Spring Boot здійснюється через сервіс Render [12], який нативно підтримує розгортання додатків на базі Maven. Підключення сервісу до GitHub-репозиторію miromax-backend у панелі керування Render дозволяє автоматизувати процес деплою. Параметри збірки налаштовуються шляхом визначення команди `mvn clean package -DskipTests`, тоді як для запуску додатка вказується команда `java -jar target/miromax-backend-0.0.1-`

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

SNAPSHOT.jar. У розділі налаштувань «Environment» додаються змінні середовища, зокрема `SPRING_DATASOURCE_URL`, секретні ключі для генерації JWT-токенів та параметри конфігурації CORS із вказанням дозволеного домену фронтенду. Завдяки такій інтеграції, після кожного коміту в гілку main, Render автоматично виконує збірку та оновлення бекенду без простоїв у роботі системи.

### 3.2 Інструкція з обслуговування та наповнення сайту

Ця інструкція визначає робочий процес адміністратора кінотеатру щодо управління мультимедійним контентом та бізнес-логікою через захищену панель керування. В основі роботи з системою лежить концепція безпечної авторизації та інтерактивного редагування репертуару.

Процедура входу до системи розпочинається з переходу на сторінку /login, де адміністратор вводить свої облікові дані. Система здійснює перевірку наданої інформації за записами в базі даних PostgreSQL, після чого генерує JWT-токен та формує відповідь із даними користувача, що підтверджують рівень адміністративних прав. Отримані дані зберігаються у локальному сховищі браузера (Local Storage). Під час наступних сесій клієнтська частина автоматично додає JWT-токен до HTTP-заголовка Authorization, а рівень доступу до окремих компонентів адмін-панелі додатково верифікується на рівні маршрутизації React-додатка. Використання саме JWT-токенів замість традиційних сесій забезпечує безстанність (stateless) архітектури та спрощує підтримку клієнт-серверної взаємодії при збільшенні кількості користувачів.

Управління каталогом фільмів здійснюється через модуль «Управління фільмами», що забезпечує відображення всіх наявних стрічок із підтримкою механізму пагінації для швидкого пошуку. Додавання нової позиції до репертуару реалізовано через інтерактивну форму, поля якої проходять обов'язкову валідацію на стороні клієнта. Адміністратор вносить основні метадані фільму, такі як назва, жанр, опис, тривалість, а також посилання на медіа-контент. Для оптимізації швидкості завантаження інтерфейсу рекомендується використовувати посилання

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

на зображення у сучасних форматах, таких як WebP. Після успішної валідації та натискання кнопки «Зберегти» транзакція передається на сервер, забезпечуючи миттєву актуалізацію даних у глобальному пошуку системи.

Формування розкладу сеансів відбувається у вкладці «Розклад сеансів», де адміністратор призначає кожен кінострічку до обраного кінозалу з урахуванням місткості приміщення. При виборі дати та часу початку показу система активує алгоритм автоматичної перевірки накладок, що запобігає виникненню часових колізій у межах одного залу. Також на цьому етапі встановлюється базова вартість квитка, яка може динамічно змінюватися залежно від формату показу, наприклад, 2D, 3D або преміум-форматів. Така гнучкість ціноутворення дозволяє адміністратору ефективно реагувати на сезонні коливання попиту, максимізуючи прибутковість кожного сеансу.

Технічне обслуговування бази даних значно спрощується завдяки використанню хмарної платформи Neon. Система автоматично створює інваріантні точки відновлення (Point-in-Time Recovery), що мінімізує ризики втрати даних та звільняє адміністратора від необхідності ручного створення дамів через стандартні утиліти на кшталт pg\_dump. У разі виникнення критичних збоїв або випадкового видалення записів, технічний фахівець має можливість відновити стан бази даних до будь-якого моменту часу через консоль керування хмарним сервісом, що забезпечує надійність та безперервність функціонування інформаційної системи кінотеатру.

### **3.3 Інструкція з популяризації та підтримки сайту**

Для успішного функціонування кінотеатру «Grand Vision» в умовах жорсткої конкуренції та залучення стабільного потоку нових клієнтів, після впровадження проєкту у продакшен рекомендовано реалізувати наступні перспективні заходи щодо пошукової оптимізації (SEO), цифрового маркетингу та постійного моніторингу технічних показників застосунку.

Заходи з популяризації розпочинаються з динамічної оптимізації метатегів.

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

Враховуючи SPA-архітектуру сайту, вкрай важливо використовувати бібліотеку типу React Helmet, яка дозволяє під час переходу на сторінку конкретного фільму динамічно змінювати теги заголовків та описів, вставляючи туди назву стрічки, що є критично необхідним для коректної індексації пошуковими ботами Google. Наступним кроком є впровадження семантичної мікророзмітки, а саме JSON-LD за стандартом Schema.org з використанням типів Movie, TheaterEvent та AggregateRating. Це дає змогу пошуковим системам формувати розширені сніпети, завдяки чому користувачі бачать рейтинг фільму, постер і розклад найближчих сеансів прямо у видачі Google ще до переходу на сайт. Також важливу роль відіграє соціальний маркетинг та інтеграція протоколу Open Graph, що забезпечує красиве відображення посилань на фільми при публікації у месенджерах та соціальних мережах. Комплексне застосування цих інструментів дозволяє вибудувати надійний фундамент для органічного зростання відвідуваності ресурсу без додаткових витрат на контекстну рекламу.

Моніторинг продуктивності та підтримка працездатності передбачає контроль показників Core Web Vitals через регулярний аудит сайту інструментом Google Lighthouse. Особлива увага приділяється метрикам LCP, що відповідає за час відмальовування найбільшого елемента, та CLS, яка контролює стабільність верстки при завантаженні. Використання Vercel [13] як хостингу фронтенду може покращити ці показники завдяки Edge-мережі кешування. Для веб-аналітики інтегруються Google Analytics 4 та Google Tag Manager з метою відстеження воронки продажів та аналізу шляху клієнта від афіші до оплати квитка, що дозволяє виявляти проблемні етапи транзакцій. Моніторинг помилок у реальному часі здійснюється за допомогою хмарних сервісів логування типу Sentry, що забезпечує перехоплення виключень на обох частинах системи та оперативну реакцію на критичні збої до того, як вони вплинуть на користувачів. Нарешті, управління безпекою та SecOps включає регулярний аудит залежностей проєкту через інструменти на кшталт GitHub Dependabot, що допомагає своєчасно оновлювати вразливих npm-пакетів та Maven-бібліотек для захисту конфіденційних даних клієнтів.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

## 4. ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даного дипломного проекту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки вебзастосунку для автоматизації роботи кінотеатру «Grand Vision». Результати розрахунків дозволять прийняти рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є вебзастосунок для автоматизації роботи кінотеатру «Grand Vision».

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

### 4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки вебзастосунку для автоматизації роботи кінотеатру «Grand Vision». Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Кер. проєкту Рm	8
		Інженер (І1)	8
2	Розробка технічного завдання	Кер. проєкту (Рm)	7
		Інженер (І1)	4
3	Дизайн інтерфейсу	Інженер (І1)	16
		Інженер (І2)	24
4	Розробка функціоналу	Інженер (І1)	42
5	Тестування та відладка	Тестувальник	8
6	Документування	Інженер (І1)	2
7	Розгортання та підтримка	Інженер (І2)	16
8	Управління проєктом	Кер. проєкту (Рm)	24
Разом			159

Сумарний час виконання операцій технологічного процесу становить 159 години.

#### 4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки вебзастосунку для автоматизації роботи кінотеатру «Grand Vision».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

					<b>2026.КВР.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c * K_r \quad (4.1)$$

де:  $T_c$  – тарифна ставка, грн. (приймаємо для керівника проекту (Pm) – 480 грн./год, інженера (I2) – 290 грн./год.), інженера (I1) – 130 грн./год., тестувальник – 110 грн./год.;  $K_r$  – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проекту (Pm)  $Z_{\text{осн.}} = 39 * 480 = 18\,720$  грн.

Інженера (I2)  $Z_{\text{осн.}} = 40 * 290 = 11\,600$  грн.

Інженера (I1)  $Z_{\text{осн.}} = 72 * 130 = 9\,360$  грн.

Тестувальник  $Z_{\text{осн.}} = 8 * 110 = 880$  грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн.}} = 18\,720 + 11\,600 + 9\,360 + 880 = 40\,560 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} * K_{\text{допл.}} \quad (4.2)$$

де:  $K_{\text{допл.}}$  – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проекту  $Z_{\text{дод.}} = 18\,720 * 0,1 = 1\,872$  грн.

Інженера (I2)  $Z_{\text{дод.}} = 11\,600 * 0,1 = 1\,160$  грн.

Інженера (I1)  $Z_{\text{дод.}} = 9\,360 * 0,1 = 936$  грн.

Тестувальник  $Z_{\text{дод.}} = 880 * 0,1 = 88$  грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод.}} = 1\,872 + 1\,160 + 936 + 88 = 4\,056 \text{ грн.}$$

Звідси загальні витрати на оплату праці ( $V_{\text{о.п.}}$ ) визначаються за формулою:

$$V_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

$$V_{\text{о.п.}} = 40\,560 + 4\,056 = 44\,616 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$V_{\text{ЄСВ}} = V_{\text{оп}} * 0,22 \quad (4.4)$$

$$V_{\text{ЄСВ}} = 44\,616 * 0,22 = 9\,815,52 \text{ грн.}$$

					<b>2026.КВР.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифна ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Кер. проекту (Pm)	480	39	18 720	1 872	4 530	25 122
2	Інженера (I2)	290	40	11 600	1 600	2 807	15 567
3	Інженера (I1)	130	72	9 360	936	2 265	12 561
4	Тестувальник	110	8	880	88	214	1 182
Разом				40 560	4 056	9816	54 432

Отже, загальні витрати на оплату праці становлять 54432 грн.

### 4.3. Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_B = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії (приймаємо 15,94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні: комп'ютер – 0,82 кВт/год.

$$Z_{ек} = 0,82 * 159 * 15,94 = 2078 \text{ грн.}$$

					<b>2026.КВР.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

Витрати на електроенергію становлять 2078 грн.

#### 4.4. Розрахунок суми амортизаційних відрахувань вебзастосування для автоматизації роботи кінотеатру «Grand Vision»

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B * H_A}{100\%} * T, \quad (4.6)$$

де: А – амортизаційні відрахування за звітний період, грн.;

Б<sub>В</sub> – балансова вартість групи основних фондів на початок звітного періоду, грн.;

Н<sub>А</sub> – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 45000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = \frac{45\ 000,00 * 0,04}{150} * 159 = 1\ 908 \text{ грн.}$$

#### 4.5. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

додаткової заробітної плати працівників.

$$H_B = V_{o.l.} * 0,2..0,6 \quad (4.7)$$

де:  $H_B$  – накладні витрати.

$$H_B = 44\,616 * 0,4 = 17\,846 \text{ грн.}$$

#### 4.6. Складання кошторису витрат та визначення собівартості вебзастосунку для автоматизації роботи кінотеатру «Grand Vision»

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.3.

Таблиця 4.3 - Кошторис витрат на розробку вебзастосунку

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	54 432	71,4
2.	Витрати на електроенергію	2 078	2,8
3.	Амортизаційні відрахування	1 908	2,6
4.	Накладні витрати	17 846	23,2
5.	Собівартість	76 264	100

Собівартість ( $C_B$ ) НДР розраховуємо за формулою:

$$C_B = V_{o.l.} + V_{c.z.} + Z_e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює  $C_B = 76\,264$  грн.

#### 4.7. Розрахунок ціни вебзастосунку для автоматизації роботи кінотеатру «Grand Vision»

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

$$Ц = Cв * (1 + P_{рен}) * (1 + ПДВ), \quad (4.9)$$

де:  $Cв$  – собівартість;  $P_{рен.}$  – рівень рентабельності;  $ПДВ$  – ставка податку на додану вартість.

$$Ц = 76\,264 * (1 + 0,3) * (1 + 0,2) = 118\,972 \text{ грн.}$$

#### 4.8. Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$ЧТВ = -Cв + \sum_{i=1}^t \frac{\Gamma_{п}}{(1+i)^i}, \quad (4.10)$$

де:  $Cв$  – собівартість розробки;  $\Gamma_{п}$  – грошовий потік за  $t$  – ий рік;  $t$  – відповідний рік проекту;  $i$  – величина дисконтної ставки (10...15%).

$$ЧТВ = -76\,264 + \frac{42\,708}{(1+0,1)^1} + \frac{42\,708}{(1+0,1)^2} + \frac{42\,708}{(1+0,1)^3} = 29\,427 \text{ грн}$$

Якщо  $ЧТВ \geq 0$ , то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + \frac{H_{в}}{\Gamma_{пр}} \quad (4.11)$$

де:  $T_{пв}$  – період до повного відшкодування витрат, років;  $H_{в}$  – невідшкодовані витрати на початок року, грн.;  $\Gamma_{пр}$  – грошовий потік на початок року, грн.

$$T_{ок} = 2 + \frac{2\,504}{42\,500} = 2,1 \text{ р.}$$

Всі дані внесемо в таблицю з Техніко-економічні показники проекту. Узагальнені техніко-економічні показники розробленого web-сайту наведено у графічній частині 2026.КВР.122.421.04.00.00 ТБ.

					<b>2026.КВР.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

Прибутковість проєкту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка вебзастосунку для автоматизації роботи кінотеатру «Grand Vision» є доцільною з економічної точки зору.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

## 5. ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

### 5.1 Охорона праці в Україні. Законодавчі акти з охорони праці

Охорона праці — це комплексна система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних засобів та заходів, спрямованих на збереження здоров'я, життя і працездатності людини у процесі її активної трудової діяльності. У сучасних умовах розвитку інформаційних технологій питання охорони праці набуває особливої актуальності, оскільки впровадження новітніх комп'ютерних систем та тривала робота за дисплейними терміналами створюють нові види професійних ризиків, які потребують жорсткого нормативного регулювання.

Правове регулювання відносин у сфері охорони праці в Україні базується на принципі безумовного пріоритету життя і здоров'я працівників по відношенню до результатів виробничої діяльності підприємства. Нормативно-правова база України з охорони праці складається з ієрархічної системи документів, головним з яких є Конституція України. Відповідно до статті 43 Конституції, кожен громадянин має фундаментальне право на належні, безпечні і здорові умови праці, а також на заробітну плату, не нижчу від визначеної законом.

Основним спеціалізованим документом є Закон України «Про охорону праці». Цей закон визначає основні положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності, на належні, безпечні і здорові умови праці, регулює за участю відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні. Згідно зі статтею 13 цього Закону, роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці. Роботодавець несе безпосередню відповідальність за організацію безпечних умов

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

праці, впровадження сучасних засобів техніки безпеки та запобігання професійним захворюванням. У свою чергу, стаття 14 Закону покладає на працівника обов'язок знати і виконувати вимоги нормативних актів про охорону праці, правила поведіння з машинами, механізмами, устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту.

Трудові відносини також регулюються Кодексом законів про працю України (КЗпП). Він встановлює норми робочого часу (не більше 40 годин на тиждень), регламентує час відпочинку, відпустки, а також містить окрему главу, присвячену виключно охороні праці. КЗпП забороняє вимагати від працівника виконання роботи, поєднаної з очевидною небезпекою для життя, а також в умовах, що не відповідають законодавству про охорону праці.

Організаційною основою управління охороною праці на підприємстві є проведення обов'язкових інструктажів. За характером і часом проведення вони поділяються на п'ять видів:

- Вступний інструктаж — проводиться з усіма працівниками, які щойно приймаються на роботу (постійну або тимчасову), незалежно від їх освіти, стажу роботи та посади.

- Первинний інструктаж — проводиться безпосередньо на робочому місці до початку виконання професійних обов'язків.

- Повторний інструктаж — проводиться на робочому місці з усіма працівниками не рідше ніж раз на півріччя (для робіт з підвищеною небезпекою — раз на квартал) з метою закріплення знань з техніки безпеки.

- Позаплановий інструктаж — проводиться при введенні в дію нових або переглянутих нормативно-правових актів з охорони праці, при зміні технологічного процесу, заміні або модернізації устаткування, приладів та інструментів.

- Цільовий інструктаж — проводиться при виконанні разових робіт, не передбачених трудовою угодою, ліквідації аварії або стихійного лиха.

Оскільки об'єктом даного дипломного проекту є розробка складного вебзастосунку для автоматизації роботи кінотеатру «Grand Vision», діяльність

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

розробника (техніка-програміста) під час проєктування архітектури бази даних, написання коду на Java та React, а також тестування програмного забезпечення належить до категорії розумової праці високої нервово-емоційної інтенсивності. Ця праця характеризується гіподинамією (тривалим перебуванням у вимушеній сидячій позі), значним напруженням зорового аналізатора та тривалою монотонією. Тому організація такого робочого місця суворо регулюється спеціалізованими підзаконними актами, зокрема НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» та ДСанПіН 3.3.2.007-98 «Державні санітарні правила та норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Відповідно до цих санітарних норм, робоче місце програміста має відповідати суворим вимогам щодо мікроклімату. Оскільки робота за комп'ютером належить до категорії легких фізичних робіт (категорія Ia, енерговитрати до 120 ккал/год), оптимальними показниками мікроклімату у робочому приміщенні є: температура повітря у холодний та перехідний періоди року повинна становити 22–24 °С, у теплий період — 23–25 °С; відносна вологість повітря має підтримуватися на рівні 40–60%; швидкість руху повітря не повинна перевищувати 0,1 м/с. Окрім температури, важливим є іонний склад повітря, оскільки працююча комп'ютерна техніка деіонізує середовище. Для підтримання оптимальних параметрів, забезпечення працівника чистим повітрям та відведення надлишкового тепла від серверного обладнання, робоче приміщення обов'язково обладнується системами припливно-витяжної вентиляції та побутовими чи промисловими кондиціонерами.

Особлива увага згідно із законодавством приділяється освітленню робочого місця, оскільки від 80 до 90% інформації програміст сприймає візуально, фокусуючись на дрібних елементах інтерфейсу та рядках вихідного коду. Приміщення повинно мати змішане освітлення: природне (через віконні прорізи) та штучне. Коефіцієнт природної освітленості (КПО) повинен бути не менше 1,5%. Робочий стіл розташовується таким чином, щоб природне світло падало на робочу поверхню переважно зліва. Рівень штучної освітленості на поверхні столу в зоні

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

розташування робочого документа повинен становити 300–500 люкс. Штучне освітлення забезпечується системою загального рівномірного освітлення з використанням сучасних LED-панелей або люмінесцентних світильників з електронною пускорегулювальною апаратурою (для мінімізації коефіцієнта пульсації світлового потоку, який не повинен перевищувати 5%). Категорично не допускається наявність прямого та відбитого блискоутворення (відблисків) на екрані монітора. Для уникнення цього вікна обов'язково обладнуються сонцезахисними конструкціями (жалюзі, ролети, штори). Крім освітлення, нормується рівень шуму: еквівалентний рівень звуку в приміщенні від працюючих кулерів системних блоків та серверного обладнання не повинен перевищувати 50 дБА.

Ергономіка робочого місця є критично важливим фактором для попередження розвитку професійних захворювань, таких як остеохондроз хребта, радикуліт та тунельний синдром зап'ястя (карпальний синдром). Конструкція робочого місця повинна забезпечувати підтримання оптимальної робочої пози. Висота робочої поверхні столу повинна регулюватися в межах 680–800 мм, а при нерегульованому столі — становити 725 мм. Ширина і глибина столу (не менше 1200 мм і 800 мм відповідно) повинні забезпечувати можливість зручного розміщення монітора, клавіатури, миші та документації. Під столом має бути забезпечений вільний простір для ніг (висотою не менше 600 мм, шириною не менше 500 мм). Робоче крісло обов'язково має бути підйомно-поворотним, стійким (на п'яти коліщатах), з можливістю плавного регулювання висоти сидіння та кута нахилу спинки. Екран монітора повинен знаходитися на відстані 600–700 мм від очей, а його верхній край — розташовуватися на рівні очей працівника або трохи нижче, щоб кут спостереження становив 10-20 градусів нижче горизонталі. Клавіатуру слід розташовувати на поверхні столу на відстані 100–300 мм від краю, щоб забезпечити можливість опори для передпліч.

Для збереження здоров'я програміста та запобігання розвитку комп'ютерного зорового синдрому (Computer Vision Syndrome), згідно з НПАОП 0.00-7.15-18, встановлюється суворий режим праці та відпочинку. При 8-

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

годинному робочому дні загальна тривалість регламентованих перерв повинна становити від 50 до 90 хвилин (наприклад, 10 хвилин після кожної години роботи). Безперервна робота за екраном дисплея не може перевищувати 2 години. Під час мікропауз та регламентованих перерв працівник зобов'язаний відводити погляд від монітора, виконувати спеціальні комплекси вправ для очей (фокусування на віддалених предметах, кругові рухи очними яблуками) та легку виробничу гімнастику для зняття статичної напруги з м'язів шийно-плечового відділу та спини.

## 5.2 Основні визначення електробезпеки

Електробезпека — це система організаційних та технічних заходів і засобів, що забезпечують комплексний захист людей від шкідливого і небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики. Електричний струм є особливо підступним і прихованим видом виробничої небезпеки, оскільки людина не має спеціальних органів чуття, здатних дистанційно виявити наявність напруги на металевих конструкціях обладнання до моменту безпосереднього дотику до них.

Під час виконання робіт над розробкою, компіляцією та тестуванням програмного комплексу кінотеатру, технік-програміст постійно експлуатує складну комп'ютерну техніку та периферійні пристрої. До них належать системні блоки, монітори, комутатори, сервери для розгортання баз даних PostgreSQL, лазерні принтери та джерела безперебійного живлення (UPS). Все це обладнання підключається до однофазної електричної мережі змінного струму напругою 220 В із частотою 50 Гц. Проходження такого електричного струму через тіло людини викликає складний комплекс фізіологічних впливів: термічну дію (опіки ділянок тіла, нагрівання кровоносних судин), електролітичну дію (розклад крові та інших органічних рідин, що порушує їх фізико-хімічний склад) та біологічну дію (роздратування і збудження живих тканин організму, що супроводжується мимовільними судомними скороченнями м'язів, зупинкою дихання та серцебиття).

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

Електротравми, які може отримати працівник, класифікують на місцеві та загальні. До місцевих електротравм належать електричні опіки (контактні або дугові), електричні знаки (плями на шкірі в місцях контакту з електродом), металізація шкіри (проникнення у верхні шари шкіри дрібних частинок металу внаслідок розплавлення під дією дуги), електроофтальмія (запалення зовнішніх оболонок очей від потужного ультрафіолетового випромінювання дуги) та механічні ушкодження (розриви шкіри та м'язів внаслідок судом). Загальні електротравми (електричний удар) є найбільш небезпечними, оскільки вони вражають весь організм. Ступінь ураження критично залежить від сили струму: змінний струм силою 0,6-1,5 мА є пороговим відчутним; струм 10-15 мА називають пороговим невідпускаючим (через судоми людина не може самостійно розтиснути руку і відірватися від струмоведучого провідника); струм 50 мА викликає порушення дихання; а струм понад 100 мА є фібриляційним і вважається смертельно небезпечним, оскільки за лічені секунди викликає хаотичне скорочення волокон серцевого м'яза та зупинку кровообігу.

Приміщення, в яких працюють програмісти (офіси, лабораторії), за рівнем безпеки ураження електричним струмом згідно з Правилами улаштування електроустановок (ПУЕ) класифікуються як приміщення без підвищеної безпеки. Це зумовлено тим, що в них відсутні фактори, що підвищують ризик (такі як висока вологість понад 75%, струмопровідний пил, хімічно активне середовище, струмопровідна підлога або можливість одночасного дотику людини до металевих корпусів електрообладнання та заземлених металоконструкцій будівель, наприклад, радіаторів опалення).

Незважаючи на відсутність підвищеної безпеки приміщення, наявність великої кількості електроприладів вимагає застосування надійних технічних засобів захисту від ураження електричним струмом. Основними з них є ізоляція струмоведучих частин, захисне заземлення, занулення та використання пристроїв захисного вимкнення (ПЗВ). Захисне заземлення — це навмисне електричне з'єднання з землею (або її еквівалентом) металевих неструмоведучих частин комп'ютерного обладнання (наприклад, металевого корпусу системного блоку), які

					<i>2026.КВР.122.421.04.00.00 ПЗ</i>	Арк.
						73
Зм.	Арк.	№ докум.	Підпис	Дата		

в нормальному стані не перебувають під напругою, але можуть опинитися під нею внаслідок випадкового пробією ізоляції. Головною метою заземлення є зниження напруги дотику до безпечного для людини рівня. Опір захисного заземлення для обчислювальної техніки в мережах до 1000 В не повинен перевищувати 4 Ом. Занулення перетворює замикання фази на корпус у коротке замикання, що призводить до миттєвого перегорання запобіжників або спрацювання автоматичних вимикачів, знеструмлюючи аварійну ділянку мережі. ПЗВ є сучасним високошвидкісним захисним апаратом, який безперервно контролює баланс струмів у фазному та нульовому проводах. У разі виникнення струму витоку (наприклад, при прямому дотику людини до оголеного проводу) ПЗВ автоматично знеструмлює мережу за час, що не перевищує 0,03 секунди, рятуючи життя. Крім того, на робочому місці проводиться боротьба зі статичною електрикою, яка накопичується на синтетичних поверхнях і може викликати збої в роботі електронних компонентів; для цього застосовують антистатичні покриття підлоги та підтримують нормативну вологість повітря.

З метою забезпечення надійної електробезпеки під час кодингу, тестування REST API та розгортання сервера, програміст зобов'язаний суворо дотримуватися правил безпечної експлуатації електроустановок. Перед початком роботи необхідно візуально перевірити цілісність ізоляції кабелів живлення, справність розеток та вилок. Категорично забороняється: експлуатувати електрообладнання з пошкодженим корпусом або іскрінням; залишати працюючу техніку без нагляду (крім серверів, розрахованих на цілодобову роботу); прокладати кабелі поблизу джерел тепла або заземляти їх меблями; знімати захисні кришки з блоків живлення, моніторів або системних блоків, якщо вони підключені до мережі. Підключення та відключення будь-яких інтерфейсних кабелів периферії повинно проводитися виключно при повністю вимкненому живленні обох пристроїв.

Крім електробезпеки, важливою складовою є пожежна безпека. Приміщення з обчислювальною технікою належать до категорії «В» за пожежною небезпекою, оскільки містять значну кількість твердих горючих матеріалів (документація, пластикові корпуси, ізоляція кабелів). Основними причинами

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

виникнення пожежі є коротке замикання, перевантаження електромережі та іскріння в несправних контактах. Для забезпечення безпеки приміщення обов'язково обладнуються автоматичною пожежною сигналізацією з димовими сповіщувачами та первинними засобами пожежогасіння.

У разі виникнення аварійної ситуації, такої як пожежа або ураження працівника електричним струмом, програміст повинен діяти за чітким і відпрацьованим алгоритмом. У випадку ураження людини струмом першочерговою та найголовнішою дією є негайне припинення дії електричного струму на потерпілого. Це робиться шляхом відключення напруги (вимкнення загального рубильника, автомата на електрощиті, або витягування вилки з розетки). Якщо швидко знеструмити обладнання неможливо, рятувальник повинен відокремити потерпілого від струмоведучих частин, дотримуючись власної безпеки: використовувати діелектричні рукавиці, сухий одяг, або відкинути провід сухою дерев'яною палицею. Після звільнення від дії струму необхідно негайно викликати швидку медичну допомогу за телефоном 103. До прибуття медиків потерпілому слід надати домедичну допомогу: покласти на рівну поверхню, забезпечити приплив свіжого повітря, розстебнути стискаючий одяг; у разі відсутності дихання та пульсу — негайно розпочати проведення серцево-легеневої реанімації (штучне дихання та непрямий масаж серця). У разі загоряння комп'ютерної техніки слід негайно знеструмити приміщення. Для гасіння електроустановок, що перебувають під напругою або щойно знеструмлені, дозволяється застосовувати виключно вуглекислотні (ВВК) або порошкові (ВП) вогнегасники. Використання води або пінних вогнегасників категорично заборонено, оскільки вони є провідниками струму і можуть призвести до ураження рятувальника. Паралельно з початком гасіння необхідно викликати пожежно-рятувальну службу за телефоном 101, сповістити керівництво та організувати евакуацію людей згідно із затвердженим планом, уникаючи паніки та не користуючись ліфтами.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						75
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було успішно розроблено, протестовано та підготовлено до впровадження сучасний вебресурс, призначений для комплексної автоматизації діяльності кінотеатру «Grand Vision». Актуальність обраного напрямку зумовлена загальносвітовими тенденціями до цифровізації бізнес-процесів та зростаючими вимогами споживачів до зручності й швидкості отримання послуг. Поставлена мета щодо створення інтерактивного середовища для клієнтів та надійних аналітичних інструментів управління для персоналу була досягнута в повному обсязі.

Проведений аналітичний огляд існуючих систем онлайн-бронювання дозволив довести технічну перевагу використання архітектури Single Page Application, перехід до якої забезпечив миттєву реакцію графічного інтерфейсу та суттєве зниження навантаження на серверну частину. Програмний комплекс був спроектований за класичною триланковою архітектурою, де клієнтська частина базується на бібліотеці React, а серверну логіку реалізовано за допомогою фреймворку Java Spring Boot із використанням бази даних PostgreSQL. У ході програмування були успішно впроваджені основні функціональні модулі: графічна мапа залу з бекенд-алгоритмом блокування крісел, система динамічного ціноутворення та механізм генерації електронних квитків. Безпеку системи забезпечено завдяки автентифікації на базі JSON Web Tokens, а також багаторівневному контролю доступу на рівні маршрутизації.

Спеціальний розділ роботи присвячено професійним регламентам розгортання системи у мережі Інтернет. Використання хмарних PaaS-платформ, зокрема Neon для бази даних та Render для бекенду, дозволило забезпечити високу доступність сервісу та створити технічну основу для безперервної інтеграції коду. Разом із цим, розроблена стратегія популяризації, що включає динамічну оптимізацію метатегів та семантичну мікророзмітку, створює умови для ефективного просування проєкту в пошукових системах.

Техніко-економічні розрахунки підтвердили високу фінансову ефективність

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						76
Зм.	Арк.	№ докум.	Підпис	Дата		

та комерційну доцільність IT-рішення. Планова собівартість проєкту становить 77 218 грн, при цьому прогнозований термін окупності капітальних інвестицій не перевищує 2,1 року, що є показником високої інвестиційної привабливості. Проведений аналіз охорони праці підтвердив дотримання всіх ергономічних та санітарних норм, необхідних для безпечної професійної діяльності програміста.

Загальним підсумком роботи є те, що практичне значення отриманих результатів полягає у суттєвому зменшенні навантаження на фізичні каси та підвищенні прибутковості підприємства за рахунок переведення клієнтів у цифрове середовище. Розроблена програмна архітектура закладає технічну основу для подальшого масштабування системи, інтеграції програм лояльності та розширення функціоналу, що відповідає потребам сучасного бізнесу.

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ПОСИЛАНЬ

1. Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І.Пулюя».
2. diagrams.net (draw.io). Онлайн-сервіс для створення структурних схем та блок-діаграм. URL: <https://app.diagrams.net/> (дата звернення: 16.05.2026).
3. W3Schools. Навчальний ресурс з веб-технологій (HTML, CSS, JavaScript). URL: <https://www.w3schools.com/> (дата звернення: 17.05.2026).
4. PlantUML. Інструмент для автоматичного створення діаграм з текстового опису. URL: <https://plantuml.com/> (дата звернення: 20.05.2026).
5. React. Офіційна документація бібліотеки для побудови користувацьких інтерфейсів. URL: <https://react.dev/> (дата звернення: 21.05.2026).
6. Spring Boot. Офіційна документація фреймворку для розробки серверних додатків на Java. URL: <https://spring.io/projects/spring-boot/> (дата звернення: 22.05.2026).
7. PostgreSQL. Документація реляційної системи керування базами даних. URL: <https://www.postgresql.org/docs/> (дата звернення: 23.05.2026).
8. Vite. Офіційна документація інструменту для автоматизації збирання фронтенд-проектів. URL: <https://vitejs.dev/guide/> (дата звернення: 25.05.2026).
9. Tailwind CSS. Документація фреймворку для швидкого налаштування дизайну інтерфейсу. URL: <https://tailwindcss.com/docs/> (дата звернення: 24.05.2026).
10. JWT (JSON Web Tokens). Офіційний стандарт безпечної передачі даних між сторонами. URL: <https://jwt.io/introduction/> (дата звернення: 26.05.2026).
11. Neon. Офіційний вебсайт хмарної платформи для безсерверної PostgreSQL. URL: <https://neon.tech/> (дата звернення: 27.05.2026).
12. Render. Хмарна PaaS-платформа для хостингу вебзастосунків. URL: <https://render.com/> (дата звернення: 28.05.2026).
13. Vercel. Хмарна платформа для розгортання фронтенд-інтерфейсів. URL: <https://vercel.com/> (дата звернення: 29.05.2026).

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
						78
Зм.	Арк.	№ докум.	Підпис	Дата		

14. React Helmet. Документація бібліотеки для керування метаданими в SPA. URL: <https://github.com/nfl/react-helmet/> (дата звернення: 30.05.2026).

15. Sentry. Платформа для моніторингу помилок та продуктивності застосунків у реальному часі. URL: <https://sentry.io/> (дата звернення: 31.05.2026).

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

## ДОДАТКИ

### Додаток А. Лістинг BookingController

```
package com.miromax.controller;

import com.miromax.dto.BookingRequest;
import com.miromax.dto.SeatHoldRequest;
import com.miromax.entity.Booking;
import com.miromax.service.BookingService;
import jakarta.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Map;

@RestController
@RequestMapping("/api/bookings")
public class BookingController {

    @Autowired
    private BookingService bookingService;

    @GetMapping("/user/{userId}")
    public ResponseEntity<List<Booking>> getUserBookings(@PathVariable Long
userId) {
        return ResponseEntity.ok(bookingService.getUserBookings(userId));
    }

    @GetMapping
    public ResponseEntity<List<Integer>> getBookedSeats(
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80

```

        @RequestParam Long movieId,
        @RequestParam String date,
        @RequestParam String time) {
    return ResponseEntity.ok(bookingService.getBookedSeats(movieId,
date, time));
    }

    @PostMapping("/holds")
    public ResponseEntity<Map<String, Object>> holdSeat(@Valid @RequestBody
SeatHoldRequest request) {
        return ResponseEntity.ok(bookingService.holdSeat(request));
    }

    @PostMapping
    public ResponseEntity<Map<String, String>> createBooking(@Valid
@RequestParam BookingRequest request) {
        bookingService.createBooking(request);
        return ResponseEntity.ok(Map.of(
            "message",
            "Seat " + request.seatNumber() + " was successfully booked."
        ));
    }

    @GetMapping("/all")
    public ResponseEntity<List<Booking>> getAllBookings() {
        return ResponseEntity.ok(bookingService.getAllBookings());
    }
}

```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

## Додаток Б. Лістинг BookingService

```
package com.miromax.service;

import com.miromax.dto.BookingRequest;
import com.miromax.dto.SeatHoldRequest;
import com.miromax.entity.Booking;
import com.miromax.entity.Session;
import com.miromax.entity.User;
import com.miromax.exception.ResourceNotFoundException;
import com.miromax.repository.BookingRepository;
import com.miromax.repository.SessionRepository;
import com.miromax.repository.UserRepository;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.transaction.support.TransactionSynchronization;
import
org.springframework.transaction.support.TransactionSynchronizationManager;

import java.time.LocalDateTime;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

@Service
public class BookingService {

    private static final Logger log =
    LoggerFactory.getLogger(BookingService.class);

    @Autowired
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

```

private SessionRepository sessionRepository;

@Autowired
private BookingRepository bookingRepository;

@Autowired
private UserRepository userRepository;

@Autowired
private EmailService emailService;

@Autowired
private SeatHoldService seatHoldService;

public List<Booking> getUserBookings(Long userId) {
    return bookingRepository.findByUserId(userId);
}

public List<Integer> getBookedSeats(Long movieId, String date, String
time) {
    Session session = findSession(movieId, date, time);
    ensureSessionIsBookable(session);

    List<Integer> unavailableSeats = session.getBookings().stream()
        .map(Booking::getSeatNumber)
        .collect(Collectors.toList());

    unavailableSeats.addAll(seatHoldService.getHeldSeats(session.getId()));

    return unavailableSeats.stream()
        .distinct()
        .toList();
}

```

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

```

public Map<String, Object> holdSeat(SeatHoldRequest request) {
    Session session = findSession(request.movieId(), request.date(),
request.time());
    ensureSessionIsBookable(session);

    if
(bookingRepository.existsBySessionIdAndSeatNumber(session.getId(),
request.seatNumber())) {
        throw new IllegalStateException("Seat " + request.seatNumber()
+ " is already booked.");
    }

    SeatHoldService.SeatHold hold = seatHoldService.holdSeat(
        session.getId(),
        request.seatNumber()
    );

    return Map.of(
        "holdToken", hold.token(),
        "expiresAt", hold.expiresAt().toString()
    );
}

@Transactional
public void createBooking(BookingRequest request) {
    log.info(
        "Creating booking: user={}, movie={}, seat={}",
        request.userId(),
        request.movieId(),
        request.seatNumber()
    );

    Session session = findSession(request.movieId(), request.date(),

```

					<b>2026.KBP.122.421.04.00.00 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

```

request.time());
    ensureSessionIsBookable(session);

    User user = userRepository.findById(request.userId())
        .orElseThrow(() -> new ResourceNotFoundException("User was
not found."));

    if
(bookingRepository.existsBySessionIdAndSeatNumber(session.getId(),
request.seatNumber())) {
        throw new IllegalStateException("Seat " + request.seatNumber()
+ " is already booked.");
    }

    seatHoldService.validateForCheckout(
        session.getId(),
        request.seatNumber(),
        request.holdToken()
    );

    Booking booking = new Booking();
    booking.setSeatNumber(request.seatNumber());
    booking.setSession(session);
    booking.setUser(user);
    booking.setStudent(request.isStudent());

    bookingRepository.save(booking);
    seatHoldService.release(session.getId(), request.seatNumber());

    sendTicketAfterCommit(
        request.email(),
        session.getMovie().getTitle(),
        request.date(),
        request.time(),

```

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

```

        request.seatNumber()
    );
}

public List<Booking> getAllBookings() {
    return bookingRepository.findAll();
}

private Session findSession(Long movieId, String date, String time) {
    LocalDateTime startDateTime = LocalDateTime.parse(date + "T" + time
+ ":00");

    return sessionRepository.findByMovieIdAndStartTime(movieId,
startDateTime)
        .orElseThrow(() -> new ResourceNotFoundException("Session
was not found."));
}

private void ensureSessionIsBookable(Session session) {
    if (!session.getStartTime().isAfter(LocalDateTime.now())) {
        throw new IllegalStateException("This session has already
started or ended.");
    }
}

private void sendTicketAfterCommit(
    String email,
    String movieTitle,
    String date,
    String time,
    int seatNumber) {
    if (email == null || email.isBlank()) {
        return;
    }
}

```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

```

TransactionSynchronizationManager.registerSynchronization(new
TransactionSynchronization() {
    @Override
    public void afterCommit() {
        emailService.sendTicket(email, movieTitle, date, time,
seatNumber);
    }
});
}
}

```

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

## Додаток В. Лістинг SeatHoldService

```
package com.miromax.service;

import org.springframework.stereotype.Service;

import java.time.Instant;
import java.util.List;
import java.util.Map;
import java.util.UUID;
import java.util.concurrent.ConcurrentHashMap;

@Service
public class SeatHoldService {

    private static final long HOLD_TTL_SECONDS = 10 * 60;

    private final Map<String, SeatHold> holds = new ConcurrentHashMap<>();

    public SeatHold holdSeat(Long sessionId, Integer seatNumber) {
        clearExpiredHolds();

        String key = key(sessionId, seatNumber);
        SeatHold existing = holds.get(key);

        if (existing != null && !existing.isExpired()) {
            throw new IllegalStateException(
                "Seat is temporarily reserved. Try another seat or wait
a few minutes."
            );
        }

        SeatHold hold = new SeatHold(
            UUID.randomUUID().toString(),
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

```

        sessionId,
        seatNumber,
        Instant.now().plusSeconds(HOLD_TTL_SECONDS)
    );

    holds.put(key, hold);

    return hold;
}

public List<Integer> getHeldSeats(Long sessionId) {
    clearExpiredHolds();

    return holds.values().stream()
        .filter(hold -> hold.sessionId().equals(sessionId))
        .map(SeatHold::seatNumber)
        .toList();
}

public void validateForCheckout(Long sessionId, Integer seatNumber,
String holdToken) {
    clearExpiredHolds();

    SeatHold hold = holds.get(key(sessionId, seatNumber));

    if (hold == null) {
        return;
    }

    if (!hold.token().equals(holdToken)) {
        throw new IllegalStateException("Seat is reserved by another
checkout.");
    }
}
}

```

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		89

```

public void release(Long sessionId, Integer seatNumber) {
    holds.remove(key(sessionId, seatNumber));
}

private void clearExpiredHolds() {
    holds.entrySet().removeIf(entry -> entry.getValue().isExpired());
}

private String key(Long sessionId, Integer seatNumber) {
    return sessionId + ":" + seatNumber;
}

public record SeatHold(String token, Long sessionId, Integer seatNumber,
Instant expiresAt) {
    boolean isExpired() {
        return Instant.now().isAfter(expiresAt);
    }
}
}

```

					<i>2026.KBP.122.421.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		90

## Додаток Г. Лістинг Booking

```
package com.miromax.entity;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import jakarta.persistence.*;
import lombok.Data;

@Data
@Entity
@Table(name = "bookings")
public class Booking {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private Integer seatNumber;

    private boolean isStudent;

    @ManyToOne
    @JoinColumn(name = "session_id")
    @JsonIgnoreProperties("bookings")
    private Session session;

    @ManyToOne
    @JoinColumn(name = "user_id")
    private User user;
}
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

## Додаток Д. Лістинг Session

```
package com.miromax.entity;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import jakarta.persistence.*;
import lombok.Data;

import java.time.LocalDateTime;
import java.util.List;

@Entity
@Table(name = "sessions")
@Data
public class Session {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne
    @JoinColumn(name = "movie_id", nullable = false)
    @JsonIgnoreProperties("sessions")
    private Movie movie;

    @ManyToOne
    @JoinColumn(name = "hall_id", nullable = false)
    @JsonIgnoreProperties("sessions")
    private Hall hall;

    private LocalDateTime startTime;

    @Column(length = 10)
    private String format = "2D";
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		92

```
@Column(nullable = false, columnDefinition = "float8 default 150.0")
private Double price = 150.0;

@OneToMany(mappedBy = "session", cascade = CascadeType.ALL)
@JsonIgnoreProperties("session")
private List<Booking> bookings;
}
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

## Додаток Е. Лістинг User

```
package com.miromax.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;
import jakarta.validation.constraints.Email;
import jakarta.validation.constraints.NotBlank;
import jakarta.validation.constraints.Size;
import lombok.Data;

import java.util.List;

@Entity
@Data
@Table(name = "users")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotBlank(message = "Username is required.")
    @Size(min = 3, message = "Username must contain at least 3 characters.")
    private String username;

    @NotBlank(message = "Email is required.")
    @Email(message = "Invalid email format.")
    private String email;

    @NotBlank(message = "Password is required.")
    @Size(min = 6, message = "Password must contain at least 6 characters.")
    private String password;
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		94

```
@Column(name = "is_admin")
private Boolean isAdmin = false;

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL)
@JsonIgnore
private List<Booking> bookings;
}
```

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

## Додаток Ж. Лістинг Movie

```
package com.miromax.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;
import jakarta.validation.constraints.NotBlank;
import lombok.Data;

import java.util.List;

@Entity
@Data
@Table(name = "movies")
public class Movie {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotBlank(message = "Title is required.")
    private String title;

    private String genre;
    private String description;
    private String imageUrl;
    private int duration;
    private Double rating = 0.0;
    private String trailerUrl;

    @Column(length = 16)
    private String language = "en";

    @OneToMany(mappedBy = "movie", cascade = CascadeType.ALL, orphanRemoval
= true)
```

					2026.КВР.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

```
@JsonIgnore
private List<Session> sessions;

public String getTrailerUrl() {
    return trailerUrl;
}

public void setTrailerUrl(String trailerUrl) {
    this.trailerUrl = trailerUrl;
}
}
```

					2026.KBP.122.421.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		97

## Таблиця техніко-економічних показників

№ п/п	Показник	Одиниці вимірювань	Значення
1.	Собівартість, грн.	грн	76 264
2.	Плановий прибуток або грошовий потік, грн.	грн	42 708
3.	Ціна, грн.	грн	118 972
4.	Чиста теперішня вартість, грн.	грн	29 427
5.	Термін окупності, рік	грн	2,1

					2026.KBP.122.42104.00.00 ТБ		
Зн.	Арх.	№ док.	Підп.	Дата	Розробка введасистеми для автоматизації роботи кінотеатру «Grand Vision»		
Розроб.	Ванчик О. О.				Лист	Маса	Масштаб
Перев.	Марціняк Г. Я.						
Т.контр.					Арх.	Архив	4
Рецензент					ВСП ТФК ТНТУ КН-421		
Н.контр.	Приймач В.А.				м. Тернопіль		
Затв.					Формат А1		

# Структурна схема прецедентів програми

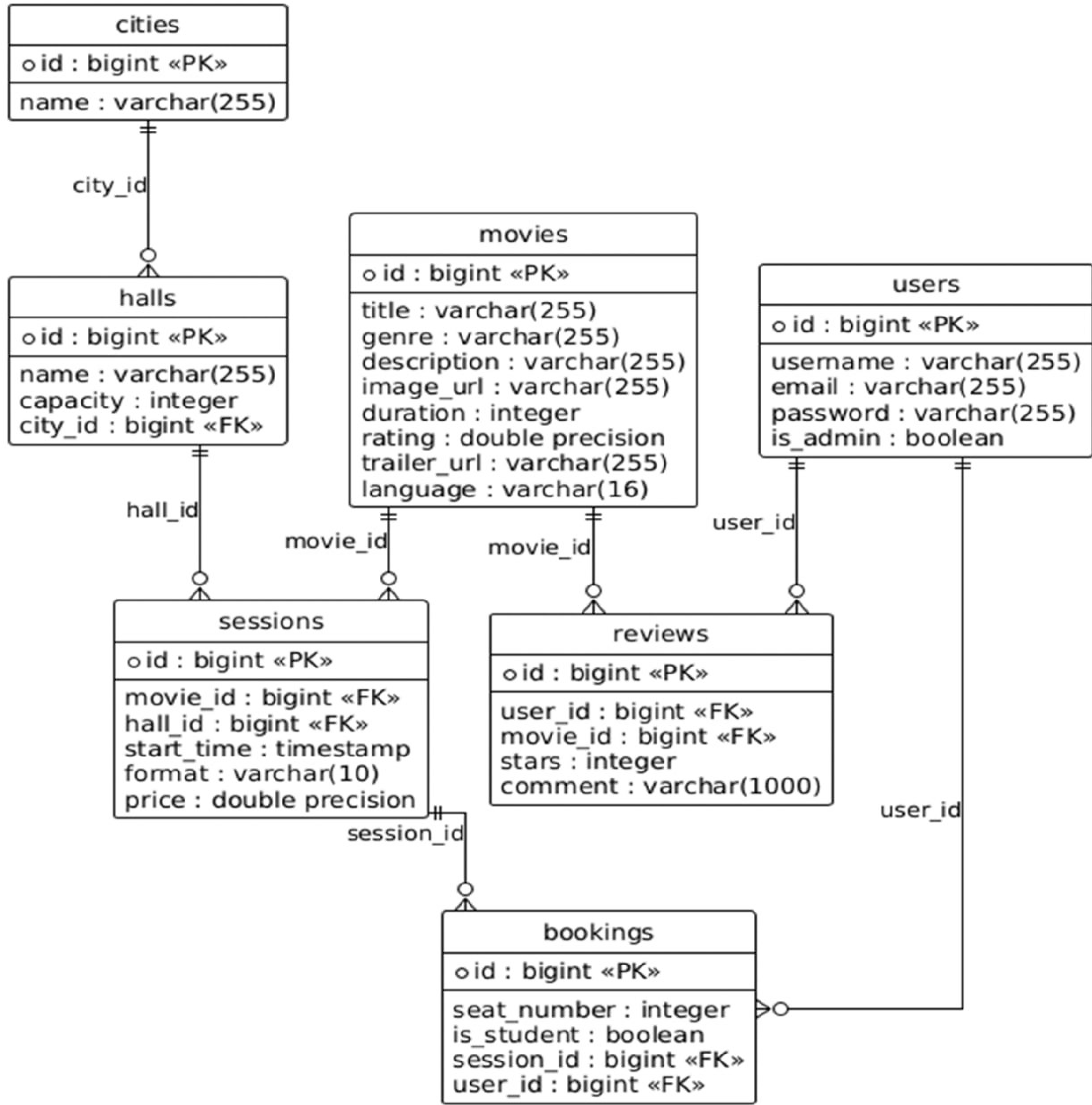


Перш. застос. Сторінка 1 з 1. Підп. і дата. Зам. наб. 1. Наб. № відг. Підп. і дата. № № аркуш. Зам. наб. 1. Наб. № відг. Підп. і дата. № № аркуш.

					2026.KBP.122.4.2104.00.00 CC		
Зм.	Арк.	№ аркуш.	Підп.	Дата	Розробка вебзастосунку для автоматизації роботи кінотеатру «Grand Vision»		
Розроб.	Ванчик О.О.				Лит.	Маса	Масштаб
Перев.	Марціняк Г.Я.						
Т.контр.					Архив	4	
Рецензент					ВСП ТФК ТНТУ м. Тернопіль		
Н.контр.	Приймак В.А.				Формат А1		
Залб.					Копія		



# ER-діаграма бази даних



Перш записати  
Сторінка  
Підпис і дата  
Зем. підс.  
Підп. і дата  
№№ в архіві

2026.KBP.122.4.2104.00.00 БД				
Зм.	Арх.	№ док.	Підп.	Дата
Разраб.	Венчик О.О.			
Перев.	Маричка Г.Я.			
Т.контр.				
Рецензент				
Н.контр.	Приймак В.А.			
Затв.				
Разработка вебзастосунку для автоматизації роботи кінотеатру «Grand Vision» ER-діаграма бази даних			Лист	Маса
			Архив	4
			ВСП ТФЖ ТНТУ	КН-421
			м. Тернопіль	
			Формат	A1