

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Онлайн кінотеатр»

Виконав студент групи ІПЗкп-20-1
КОЙДА Владислав Миколайович

Керівник роботи:
ЧИЖМОТРЯ Олексій Володимирович

Рецензент:
КРАВЧЕНКО Світлана Миколаївна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Завідувач кафедри ІПЗ

_____ Тетяна ВАКАЛЮК

"01" березня 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КОЙДА Владислав Миколайович**

Керівник роботи: **ЧИЖМОТРЯ Олексій Володимирович**

Тема роботи: **«Онлайн кінотеатр»**,

затверджена Наказом закладу вищої освіти від **«01» березня 2024 р. № 87/с**

Вихідні дані для роботи: **використання веб-орієнтованих технологій для створення сайту кінотеатру**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів,
що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря О. В.	03.03.2024 р.	03.03.2024 р.
2	Чижмотря О. В.	03.03.2024 р.	03.03.2024 р.
3	Чижмотря О. В.	03.03.2024 р.	03.03.2024 р.

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання. Опрацювання літературних джерел.	3 березня 2024 – 5 березня 2024	Виконано
2	Проектування структури системи	6 березня 2024 – 9 березня 2024	Виконано
3	Написання програмного коду	10 березня 2024 - 21 березня 2024	Виконано
4	Тестування	22 березня 2024 – 23 березня 2024	Виконано
5	Аналіз одержаних результатів	24 березня 2024 – 25 березня 2024	Виконано
6	Оформлення пояснювальної записки	26 березня 2024 – 28 березня 2024	Виконано
7	Захист	30 березня 2024	

Реферат

Випускна кваліфікаційна робота складається з веб-сайту кінотеатру та пояснювальної записки. Пояснювальна записка до випускної роботи містить 75 сторінок, 26 ілюстрацій та 1 таблицю, список літературних джерел містить 17 найменувань.

Метою кваліфікаційної роботи є створення веб-сайту кінотеатру, який надає користувачам зручну платформу для пошуку фільмів, перегляду розкладу сеансів та придбання квитків. Сайт забезпечує можливість перегляду інформації про фільми, а також отримання рекомендацій щодо фільмів на основі новизни фільмів. Крім того, адміністратори кінотеатру можуть управляти розкладом сеансів, додавати нові фільми та відслідковувати статистику відвідування. Основною метою проекту є створення зручного та ефективного сервісу для кінолюбителів, які шукають якісний досвід перегляду кіно.

КЛЮЧОВІ СЛОВА: ВЕБ-САЙТ, BOOTSTRAP, SYMFONY, PHP.

					ІПЗ.КР.Б – 121 – 24 – ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	Пояснювальна записка «Онлайн кінотеатр»	Літ.	Арк.	Аркушів	
Розроб.		В.М. Койда							
Керівник		О.В. Чижмотря					4	75	
Консульт.						Житомирська політехніка група ІПЗкп-20-1			
Рецензент.		С.М. Кравченко							
Зав. каф.		Т.А. Вакалюк							

Abstract

The graduation thesis consists of a cinema website and an explanatory note. The explanatory note to the final thesis contains 75 pages, 26 illustrations and 1 table, the list of literary sources contains 17 titles.

The goal of the thesis project is to create a cinema website that provides a convenient platform for users to search for films, view the schedule of screenings and purchase tickets. The site provides the ability to view movie information and receive movie recommendations based on movie recency. In addition, cinema administrators can manage the schedule of screenings, add new films and track attendance statistics. The main goal of the project is to create a convenient and efficient service for moviegoers who are looking for a quality movie-watching experience.

KEYWORDS: WEBSITE, BOOTSTRAP, SYMFONY, PHP.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ЗАДАЧІ ТА ОСОБЛИВОСТЕЙ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ САЙТУ ПОШУКУ РОБОТИ	9
1.1. Постановка задачі.....	9
1.2. Аналіз аналогів програмного продукту.....	11
1.3. Вибір архітектури веб-сайту пошуку роботи.....	14
1.4. Інструментальні засоби реалізації програмного продукту.....	17
Висновки до розділу 1.....	21
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ ПОШУКУ РОБОТИ	24
2.1. Визначення варіантів використання системи.....	24
2.2. Розробка бази даних системи.....	28
2.3. Проектування та реалізація бази даних роботи системи	35
2.4. Реалізація веб-додатку	38
Висновки до розділу 2	50
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-САЙТОМ ПОШУКУ РОБОТИ	51
3.1. Порядок встановлення та налаштування системи	51
3.2. Структура інтерфейсу веб-сайту пошуку роботи	53
3.3. Тестування роботи веб-сайту.....	64
Висновки до розділу 3	66
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	69
ДОДАТКИ.....	71

Вступ

Актуальність теми: У сучасному світі уникнути черг є досить важливим, це економить багато часу для інших справ. А тому розробка платформи для замовлення квитків є актуальним. Ця платформа дасть змогу клієнтам замовити швидко та зручно квиток на фільм, який вас зацікавив, без необхідності походу в кінотеатр, з будь-якої точки світу. Це може стати чудовою можливістю зробити подарунок для дорогої людини. А квиток без прив'язки до часу та місця в залі відкриває ряд можливостей для клієнтів кінотеатру.

Метою даної кваліфікаційної роботи є: розробка платформи замовлення квитків в кінотеатр, який дасть можливість користувачам замовляти квитки на фільми, без необхідності походу в кінотеатр. Завдання полягає в створенні зручного інтерфейсу, що дасть змогу швидко та легко замовляти квитки на доступні фільми.

Об'єктом дослідження є: розробка, проектування та реалізація веб-сайту кінотеатру.

Предметом дослідження є: використання веб-орієнтованих технологій для створення сайту кінотеатру.

Методи дослідження: в роботі було використано методи об'єктно орієнтованого проектування, веб-програмування та дизайну

У першому розділі кваліфікаційної роботи досліджуються теоретичні аспекти, пов'язані з розробкою веб-сайту для кінотеатру. В даному розділі проводиться аналіз інформаційної бази та літературних джерел, де визначаються ключові завдання впровадження інформаційних технологій у контексті даного дослідження. Також проводиться огляд існуючих аналогів та формулюється архітектура майбутнього веб-додатку для кінотеатру.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Другий розділ роботи розглядає аспекти проектування та розробки веб-сайту кінотеатру. В цьому розділі докладно визначаються функціональні вимоги до сайту. Детально описується процес проектування та розробки додатку, а також надається визначення функціональних вимог до веб-сайту. Підсвічується етап проектування та розробки веб-додатку, де також розглядаються використовувані технології для розробки веб-додатків. Також висвітлюється обґрунтування вибору конкретного інструментарію для створення веб-сайту кінотеатру.

У другому розділі також докладно висвітлюється архітектура розробленого веб-додатку, його функціональні можливості та особливості реалізації. В цьому розділі будуть розглянуті аспекти, пов'язані з використанням баз даних для зберігання та обробки інформації, а також інших необхідних технологій, що забезпечують ефективну роботу сайту кінотеатру.

Третій розділ містить діаграми розгортання та компонентів системи, які сприяють легшому розумінню архітектури та функціональності веб-сайту кінотеатру. Ці діаграми допомагають наочно відобразити, як система взаємодіє та які компоненти відповідають за різні аспекти роботи веб-сайту.

В третьому розділі також надається структура інтерфейсу адміністратора, проходить етап тестування з виправленням виявлених помилок.

РОЗДІЛ 1. АНАЛІЗ ЗАДАЧІ ТА ОСОБЛИВОСТЕЙ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ САЙТУ ПОШУКУ РОБОТИ

1.1. Постановка задачі

Постановка задачі полягає у конкретизації цілей і завдань проекту, які необхідно вирішити для створення сайту пошуку роботи на Symfony. Основна мета полягає у створенні ефективного і зручного інструменту для клієнтів кінотеатру, яка дасть змогу швидко і зручно замовити квитки для подальшого походу в кінотеатр без необхідності стояти в чергах.

В рамках дослідження та розробки цього сайту, поставлено наступні конкретні задачі:

1. Розробка функціональності сайту кінотеатру: Сайт повинен мати зручний інтерфейс вибору фільмів для замовлення квитків в кінотеатр.
2. Оптимізація та розширення функціоналу: Важливим аспектом є оптимізація сайту, забезпечення його швидкодії та ефективності. Також враховуючи розвиток кіноіндустрії, сайт повинен мати можливість розширення функціоналу та адаптації до нових вимог та технологій.
3. Розробка адміністративної панелі: Сайт повинен мати адміністративну панель, яка дозволить модераторам керувати фільмами, користувачами, сеансами, переглядати квитки і мати ряд інших функцій, для зручного керування сайтом.

Не функціональні вимоги до сайту - це вимоги, що не стосуються безпосередньо функціональності системи, але визначають його якості, характеристики та параметри використання. Основна відмінність між функціональними та не функціональними вимогами полягає в тому, що функціональні вимоги описують, що система повинна робити, а не функціональні - як вона повинна це робити.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Не функціональні вимоги включають:

1. Надійність: Вимоги до стабільності, надійності та доступності системи. Наприклад, максимальний час відновлення після збою, частота виникнення помилок, можливість резервного копіювання даних та відновлення роботи системи.
2. Безпека: Вимоги до захисту конфіденційності, цілісності та доступності даних. Наприклад, необхідність шифрування даних, обмеження доступу до конфіденційної інформації, захист від несанкціонованого доступу та зламу системи.
3. Сприйняття: Час відповіді системи для запитів не повинен перевищувати 10 секунд

Результатом виконання поставленого завдання буде розроблений функціональний сайт, який забезпечує можливість користувачам замовляти квитки на фільми, які їх зацікавлять. Сайт повинен мати зручний та інтуїтивно зрозумілий інтерфейс та добре організовану навігацію.

1.2. Аналіз аналогів програмного продукту

Для порівняння веб-сайтів, які займаються продажем квитків в кінотеатр в Житомирі, буде корисним провести аналіз наступних сайтів: kinoukraina.zt.ua, frankocinema.zt.ua.

Кінотеатр Україна - це сучасний 2-х зальний кінотеатр європейського рівня, оснащений новітнім обладнанням: кінопроекційна CHRISTIE, DOLBY Digital, які відповідають всім світовим стандартам, комфортними кріслами з підставками для напоїв і попкорну та зручними диванами.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

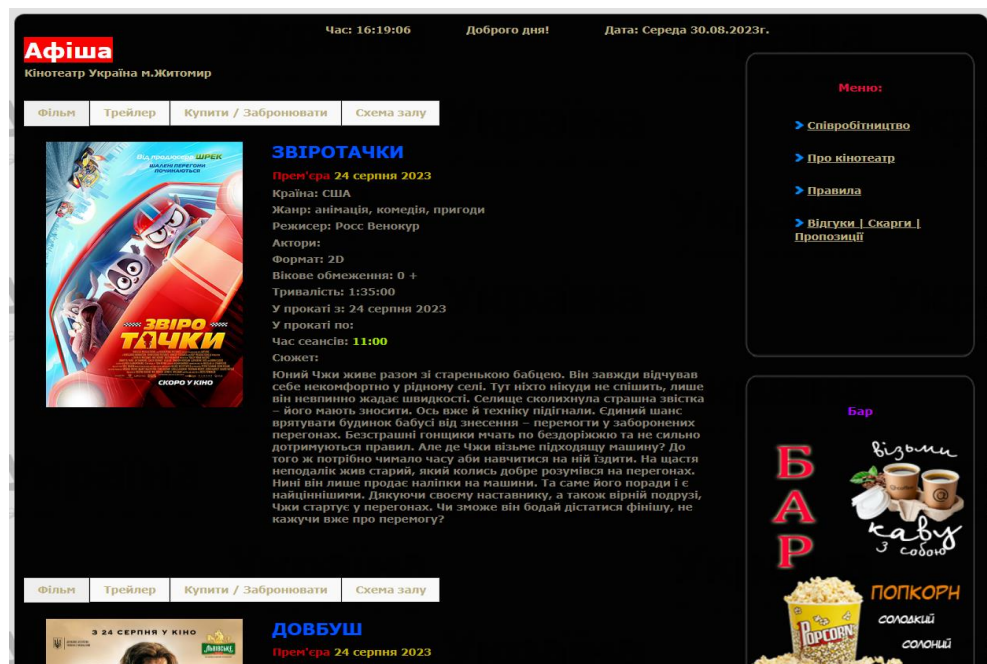


Рисунок 1.1 - Інтерфейс kinoukraina.zt.ua

Основні переваги сайту kinoukraina.zt.ua з точки зору програміста:

- Технічна підтримка та оновлення: Н сайті працює ефективна технічна підтримка, що дозволяє вчасно вирішувати будь-які технічні проблеми, які користувачі можуть зустріти.
- Простота та зручність користування: Сайт має інтуїтивно зрозумілий та зручний інтерфейс, що робить його простим та зручним у використанні для користувачів.

Недоліки сайту kinoukraina.zt.ua з точки зору програміста:

- Швидкість роботи сайту. На деяких пристроях веб-сайт завантажується дуже повільно. Це може бути зв'язано з недостатнім оптимізуванням коду та зображень.
- Робота з мобільними пристроями. Не всі сторінки сайту адаптовані для роботи з мобільними пристроями.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

- Навігація. Деякі користувачі можуть знайти навігацію сайту не зовсім зручною та не інтуїтивно зрозумілою. Це може призвести до того, що користувачі не зможуть знайти потрібну їм інформацію.

Кінотеатр має особливий статус – спеціалізованого дитячого кінотеатру, тому в репертуарі кінотеатру завжди передбачено фільми для маленьких глядачів, крім того існує система знижок для колективних замовлень від 20 осіб.

Сьогодні кінотеатр обладнаний сучасною системою цифрового кінопоказу NEC NC1200C. Дана система кінопоказу має надзвичайно високу яскравість зображення (9000 люменів) і дає можливість демонструвати найсучасніші кінофільми в форматі DCP, які виходять в прокат на території України. Точне та яскраве зображення, розмір екрану(9мХ4м), звукові спецефекти та зручні м'які крісла створюють в кінозалі атмосферу «присутності» і комфорту.

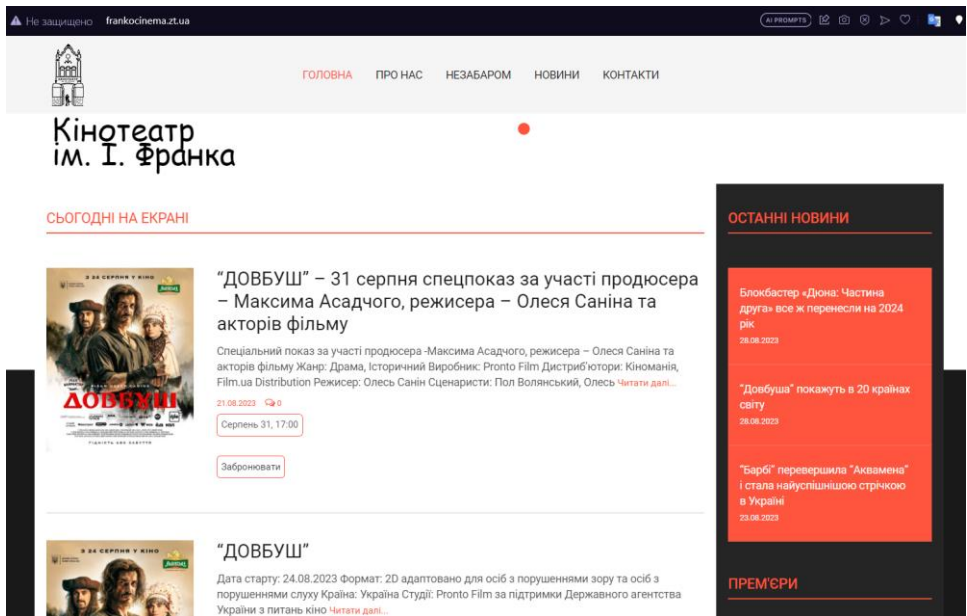


Рисунок 1.2 - Інтерфейс frankocinema.zt.ua

Оцінка frankocinema.zt.ua з точки зору програміста може бути наступною:

Плюси:

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- Зручна навігація: сайт frankocinema.zt.ua має зручну для користувачів навігацію по сайту. Це дає змогу легко та швидко знайти те, що необхідно.

Мінуси:

- Помилки в роботі: дуже часто сайт зависає, з'являються критичні помилки, не підвантажуються JavaScript і т.д.
- Відсутність регулярних оновлень: сайт може не оновлюватись регулярно, що може призвести до того, що він стане менш привабливим для користувачів.

Зважаючи на проведений аналіз сайтів frankocinema.zt.ua та kinoukraina.zt.ua, було зроблено висновок, що розроблений веб-сайт кінотеатру має ряд переваг. По-перше, сайт має інтуїтивно зрозумілий та легкий у використанні інтерфейс, що дозволить користувачам швидко орієнтуватися для замовлення квитків. Отже, розроблюваний веб-сайт пошуку роботи має конкурентні переваги порівняно зі своїми аналогами, що зробить його бажаним інструментом для замовлення квитків багатьом користувачам.

1.3. Вибір архітектури веб-сайту пошуку роботи

Для успішного розроблення сайту пошуку роботи використання відповідної архітектури є важливим кроком. Під час вибору архітектурного рішення було розглянуто кілька варіантів, включаючи Model-View-Controller (MVC), RESTful API та інші шаблони проектування.

MVC є одним з популярних шаблонів проектування, який забезпечує чітке розділення логіки бізнес-логіки, представлення даних та обробку запитів. Його переваги полягають у легкій модульності і розширюваності коду, покращеній повторній використовуваності та можливості паралельної роботи декількох розробників. Однак, також існує потенційний недолік у вигляді

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

можливості створення завеликих контролерів та складності управління великими проектами.

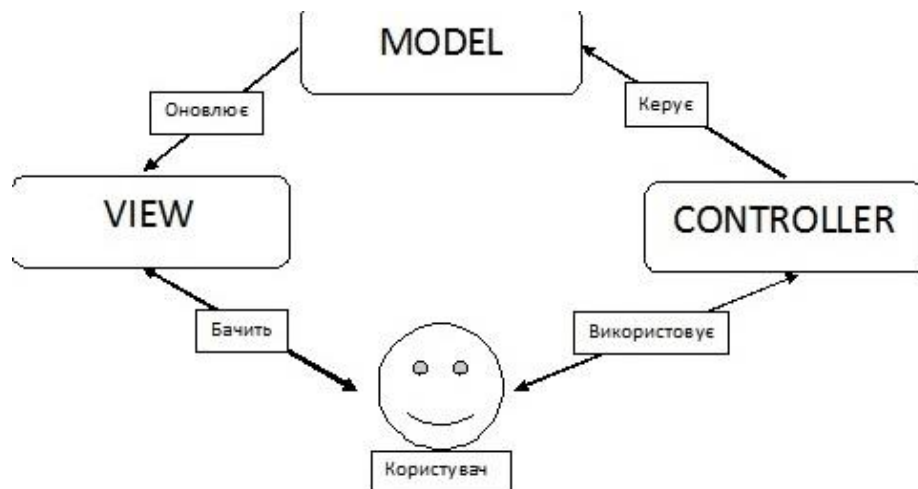


Рисунок 1.3 - Схема архітектури MVC

RESTful API є стандартним підходом для розробки веб-служб, який спрощує взаємодію з зовнішніми додатками або сервісами. Ця архітектура забезпечує незалежність між клієнтом та сервером, дозволяючи їм розвиватися незалежно. Вона також підтримує різні формати даних, такі як JSON або XML, і має вбудовану підтримку кешування для зниження навантаження на сервер. Однак, робота з RESTful API вимагає додаткових зусиль для визначення і налаштування маршрутів та управління HTTP-запитами.

Крім того, також було розглянуто інші архітектурні підходи, такі як розширена шарова архітектура за допомогою сервісів, що дозволяють створювати окремі класи для виконання конкретних завдань. Цей підхід полегшує керування кодом та забезпечує його модульність і перевикористовуваність.

Урахування особливостей кожної архітектури та їх плюсів та мінусів дозволить ефективно побудувати сайт пошуку роботи, забезпечити його масштабованість, модульність та підтримку у майбутньому.

MVC має багато переваг, які роблять його відмінним вибором для реалізації веб-сайту пошуку роботи в рамках кваліфікаційної роботи:

1. Підтримка повторного використання коду: Оскільки MVC розділяє логіку додатку на компоненти, це дозволяє легко перевикористовувати код. Модель може бути використана безпосередньо в інших проектах, представлення може бути повторно використане для відображення даних в різних контекстах, а контролер може мати загальні функціональність, яка може бути використана в інших частинах проекту.
2. Покращена тестованість: Оскільки логіка додатку розділена на компоненти, кожен з них може бути протестований окремо. Це дозволяє проводити модульне тестування моделі, представлення та контролера без необхідності тестувати всю систему в цілому. Це полегшує виявлення помилок та забезпечує вищу якість програмного забезпечення.
3. Підтримка паралельної розробки: Завдяки розділенню відповідальностей, розробка може відбуватись паралельно. Команда може працювати над моделлю, представленням та контролером незалежно одна від одної, що полегшує процес розробки та зменшує ймовірність конфліктів у коді.
4. Розширюваність: MVC надає гнучкість та легкість розширення функціональності. Оскільки кожен компонент має чітко визначені обов'язки, додавання нових функцій або модифікація існуючих стає простішим і не впливає на решту системи.

Таким чином, використання патерна проектування MVC у кваліфікаційній роботі обґрунтовано його перевагами, такими як розділення відповідальностей, підтримка повторного використання коду, полегшення тестування, підтримка паралельної розробки та розширюваність. Це допоможе створити структуровану, гнучку та легко розширювану систему пошуку роботи.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4. Інструментальні засоби реалізації програмного продукту

Існують різні мови програмування для веб-розробки, які використовуються для роботи з інтернет-технологіями. Деякі з них спеціально призначені для створення веб-додатків, тоді як інші можуть використовуватися для роботи з будь-якими типами ресурсів. Деякі мови, такі як PHP, набули популярності та широкого визнання протягом тривалого часу.

Мови програмування для веб-розробки можна поділити на дві основні групи: клієнтські та серверні. Клієнтські мови, такі як HTML, CSS та JavaScript, використовуються для розробки веб-інтерфейсу, який відображається у веб-браузері користувача. HTML використовується для створення структури та розмітки веб-сторінок, CSS - для оформлення та стилізації цих сторінок, а JavaScript - для динамічного та інтерактивного взаємодії з користувачем.

Серверні мови програмування, які включають PHP, Python, Ruby та інші, використовуються для обробки запитів від клієнта, взаємодії з базами даних та формування відповідей, які надсилаються на клієнтський браузер. Ці мови дозволяють розробникам створювати потужні веб-додатки зі складною логікою та взаємодією з даними.

Кожна з цих мов має свої особливості, переваги та недоліки. Вибір мови програмування залежить від конкретних вимог проекту, рівня досвіду розробника та специфіки задачі. Правильний вибір мови програмування допомагає забезпечити ефективну та масштабовану розробку веб-додатків.

1.4.1. PHP

PHP (Hypertext Preprocessor) є однією з найпопулярніших мов програмування для веб-розробки. Вона була спеціально створена для розробки динамічних веб-сторінок і взаємодії з базами даних. Ось деякі ключові особливості мови PHP для веб-розробки:

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Синтаксис: Синтаксис PHP схожий на синтаксис C, що робить його зрозумілим для багатьох програмістів. PHP-код вбудовується безпосередньо в HTML-сторінки, що дозволяє комбінувати статичний контент з динамічним.
2. Розширюваність: PHP має велику кількість розширень і бібліотек, які дозволяють розробникам розширити функціональність своїх додатків. Це означає, що можна використовувати готові рішення для роботи з електронною поштою, обробки зображень, генерації PDF-файлів та багатьох інших завдань.
3. Підтримка фреймворків: PHP має багато популярних фреймворків для веб-розробки, таких як Laravel, Symfony, CodeIgniter, Yii та інші. Ці фреймворки надають структуру, готові компоненти та набір інструментів для швидкої та ефективної розробки веб-додатків.
4. Платформна незалежність: PHP є платформно-незалежною мовою, що означає, що вона може працювати на різних веб-серверах, таких як Apache, Nginx, IIS та інші. Ви також можете розробляти на PHP в будь-якій операційній системі, такій як Windows, Linux або macOS.

Окрім плюсів, слід також перерахувати і мінуси:

1. Безпека: PHP в минулому мала проблеми з безпекою, існували вразливості та можливості для атак. Хоча з часом були внесені поліпшення і виправлення, все ж розробники повинні бути обережними та дотримуватися найкращих практик забезпечення безпеки.
2. Мовні особливості: PHP має деякі мовні особливості та конструкції, які можуть бути складними для розуміння та використання. Наприклад, деякі непослідовності у синтаксисі або варіативність назв

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

функцій та параметрів можуть вимагати більшої уваги до деталей під час програмування.

1.4.2. Ruby

Ruby є мовою програмування, яка здобула популярність серед веб-розробників завдяки своїй простоті, елегантності та зосередженості на продуктивності розробки. Особливо популярним фреймворком для веб-розробки на Ruby є Ruby on Rails (часто називають просто Rails).

Ось деякі ключові особливості мови програмування Ruby для веб-розробки:

1. Синтаксис: Ruby має приємний та зрозумілий синтаксис, що нагадує природну мову. Він підтримує використання динамічних методів та операторів, що спрощує розробку та робить код більш зрозумілим.
2. Об'єктно-орієнтовані можливості: Ruby є повністю об'єктно-орієнтованою мовою, де все є об'єктом. Вона підтримує класи, спадкування, поліморфізм та багато інших ООП-концепцій, що сприяють розробці простих і логічно структурованих програм.
3. Ruby on Rails: Ruby on Rails є потужним фреймворком для швидкої розробки веб-додатків. Він надає готові рішення та конвенції, що допомагають розробникам ефективно працювати та зосереджуватися на бізнес-логіці, а не на низькорівневих деталях. Rails має вбудовану підтримку для маршрутизації, взаємодії з базами даних, кешування, автентифікації та багатьох інших функцій.
4. Багата екосистема: Ruby має широку спільноту розробників, що веде до наявності великої кількості готових бібліотек та пакетів, які значно спрощують розробку веб-додатків. Це дозволяє розробникам швидко та ефективно будувати функціональність, використовуючи вже існуючі рішення.

5. Масштабованість: Ruby та Rails підтримують горизонтальну масштабованість, що означає, що вони можуть легко обробляти зростаючі навантаження шляхом розподілу завдань на багато серверів.

Крім перерахованого вище переліку плюсів, варто також враховувати і вірогідні мінуси, деякі з яких значно впливають на загальну оцінку:

1. Швидкодія: у порівнянні з деякими іншими мовами програмування Ruby може бути менш ефективною з точки зору швидкодії та продуктивності. Це особливо помітно при обробці великого обсягу даних або вимогах до високої швидкості виконання.
2. Менша популярність та ресурси: у порівнянні з PHP або Python, Ruby має меншу спільноту розробників та менше кількості готових ресурсів, таких як фреймворки, бібліотеки або документація. Це може ускладнити пошук підтримки або вирішення проблем під час розробки.
3. Ресурсоємність: Ruby відома своєю високою вживаністю ресурсів, таких як пам'ять та процесорний час. Це може призводити до збільшеного споживання пам'яті та більшої потреби у потужних серверах для підтримки веб-додатків, особливо якщо вони мають велику кількість одночасних користувачів.

1.4.3. Python

Python - це високорівнева, універсальна мова програмування, яка здобула значну популярність у сфері веб-розробки. Хоча Python і використовується в різних галузях, включаючи наукові дослідження та штучний інтелект, він також має багато фреймворків та інструментів, спеціально призначених для веб-розробки. Ось деякі ключові особливості мови Python для веб-розробки:

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

1. Зручний синтаксис: синтаксис Python дуже зрозумілий та читабельний. Він нагадує англійську мову та має мінімальну кількість символів. Це робить код більш зрозумілим та підтримує швидшу розробку.
2. Швидкість розробки: завдяки зручному синтаксису, багатій екосистемі та фреймворкам, розробка веб-додатків на Python може бути швидкою та ефективною. Фреймворки, такі як Django, пропонують готові рішення для автентифікації, маршрутизації, роботи з базами даних та багато іншого.
3. Розширюваність та інтеграція: Python може легко інтегруватися з іншими мовами програмування, такими як C, C++, або Java. Це дозволяє використовувати швидкі низькорівневі бібліотеки для оптимізації певних частин веб-додатків.

Нарівні з плюсами, у мови програмування Python також є і мінуси, зокрема:

1. Доступ до бази даних: програмувати на цій мові легко, але коли ми взаємодіємо з базою даних, її не вистачає. Рівень доступу до бази даних у Python примітивний та недостатньо розвинений у порівнянні з іншими популярними технологіями.
2. Помилки виконання: це мова з динамічною типізацією, тому тип даних змінної може змінюватись у будь-який час. Змінна, що містить ціле число, у майбутньому може містити рядок, що може призвести до помилок виконання.
3. Неефективна для пам'яті: ця мова програмування використовує великий обсяг пам'яті, це може бути недоліком при створенні програм, коли віддають перевагу оптимізації пам'яті.

Було обрано мову програмування PHP та фреймворк Symfony з кількох причин.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

По-перше, PHP є однією з найпопулярніших мов програмування для веб-розробки. Вона має велику спільноту розробників, широкий вибір ресурсів та документацію, що робить її легкою для вивчення та використання.

По-друге, PHP дозволяє швидко розробляти веб-додатки завдяки своїй простоті та гнучкості. Вона має вбудовану підтримку взаємодії з базами даних, роботи з формами та обробки HTTP-запитів, що спрощує процес розробки.

По-третє, фреймворк Symfony є потужним інструментом для розробки веб-додатків на основі PHP. Він надає готові компоненти та структуру для швидкої розробки високоякісних додатків. Symfony пропонує зручний механізм маршрутизації, шаблонів, ORM для роботи з базами даних, аутентифікацію та авторизацію, що спрощує розробку складних функціональностей.

Вибір мови програмування PHP та фреймворку Symfony обґрунтовується їхньою популярністю, простотою використання, широкими можливостями та наявністю активної спільноти розробників. Ці фактори сприяють швидкому розробленню веб-додатків високої якості та забезпечують підтримку та розширення в майбутньому.

Висновки до розділу 1

В першому розділі було поставлено технічне завдання, проаналізовані існуючі аналоги, такі як: kinoukraina.zt.ua, frankocinema.zt.ua. На основі аналізу було зроблено висновки, щодо функціоналу веб-сайту.

Для реалізації було вирішено використати такі інструментальні засоби: Symfony, базу даних MySQL, Bootstrap. Наведено їх короткий опис.

Було розглянуто існуючі архітектури та обрано патерн проектування MVC.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ ПОШУКУ РОБОТИ

2.1. Визначення варіантів використання системи

Діаграма використання є одним з ключових елементів моделювання вимог у розробці програмного забезпечення. Вона використовується для візуалізації та аналізу взаємодії між акторами (користувачами, системами або зовнішніми компонентами) та системою, яка розробляється.

Основна мета діаграми використання полягає в тому, щоб зрозуміти, як користувачі та інші актори взаємодіють з системою, які дії вони виконують та які результати отримують. Вона дозволяє розробникам програмного забезпечення та стейкхолдерам отримати уявлення про функціональність системи, її можливості та вимоги.

Діаграма використання допомагає виявити основні функціональність системи, ідентифікувати акторів та їх ролі, визначити вимоги до системи та зрозуміти потреби користувачів. Вона також може використовуватися для виявлення помилок або недоліків у проектуванні перед початком розробки.

Діаграма використання наведена на рисунку 2.1.

Вимоги користувачів

Усі користувачі мають доступ до наступних функцій:

1. Перегляд списку фільмів;
2. Перегляд детальної інформації про фільм;
3. Можливість замовити та оплатити квиток на фільм;

Авторизовані користувачі з правами «Адміністратор» мають додатковий доступ до наступних функцій:

1. Вхід у адмінпанель;
2. Управління вмістом БД

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Не функціональні вимоги

1. Сприйняття

- Час, потрібний для навчання інструментами роботи з інформаційною системою для звичайних користувачів – 2-3 години, а для досвідчених – 1 година;
- Час відповіді системи для звичайних запитів не повинен перевищувати 5 секунд, а для більш складних запитів – 10 сек;
- Інтерфейс представлення ІС повинен бути інтуїтивно зручним для користувача та не вимагати від нього додаткової підготовки.

2. Надійність

- Доступність – час, потрібний для обслуговування системи не потрібен перевищувати 10% від загального часу роботи.
- Середній час безперервної роботи – 20 робочих днів.
- Максимальна норма помилок в роботі системи – 1 помилка на 1000 запитів користувачів.

3. Продуктивність

Система повинна підтримувати мінімум 100 одночасно працюючих користувачів, пов'язаних з спільною базою даних.

Аналіз користувацьких та не функціональних вимог дозволив виділити сутності, що забезпечать реалізацію програмного комплексу системи.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

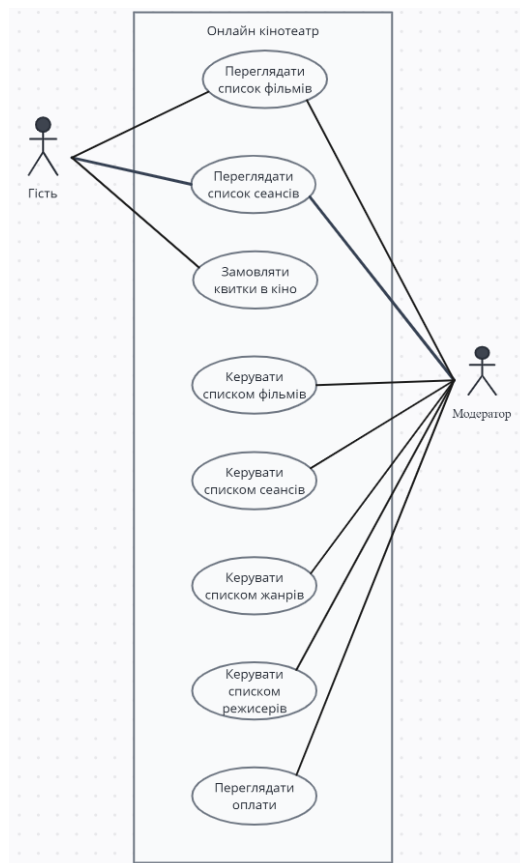


Рисунок 2.1 - Діаграма використання

Кожен користувач (навіть не авторизований) веб сайту може переглядати сторінку з фільмами. Користувач може побачити список сеансів фільму. Користувач може оформити замовлення квитка на фільм.

Користувач з правами «Модератор» має безпосередній доступ до бази даних. Модератор переглядає новостворені резюме та вакансії і перевіряє чи були дотримані правила заповнення. Якщо ці правила були порушено, то адміністратор видаляє резюме чи вакансію а користувач отримує сповіщення з причиною видалення.

2.2. Розробка бази даних системи

У Symfony бази даних створюються за допомогою міграцій. Міграції є потужним інструментом для керування структурою бази даних. Вони дозволяють розробникам описувати та зберігати структуру бази даних у вигляді

коду PHP, що дозволяє легко створювати, змінювати та синхронізувати бази даних між різними оточеннями.

У міграції Symfony можна описувати структуру бази даних, включаючи таблиці, поля, індекси, зовнішні ключі та інші складові бази даних. Міграція описує потрібні зміни структури бази даних і дозволяє створювати, змінювати або видаляти таблиці та їхні складові за допомогою коду.

Спроектowana та розроблена база даних містить 8 таблиць. Враховуючи всі умови та вимоги до розроблюваного сервісу складено наступну базу даних (рис 2.2).

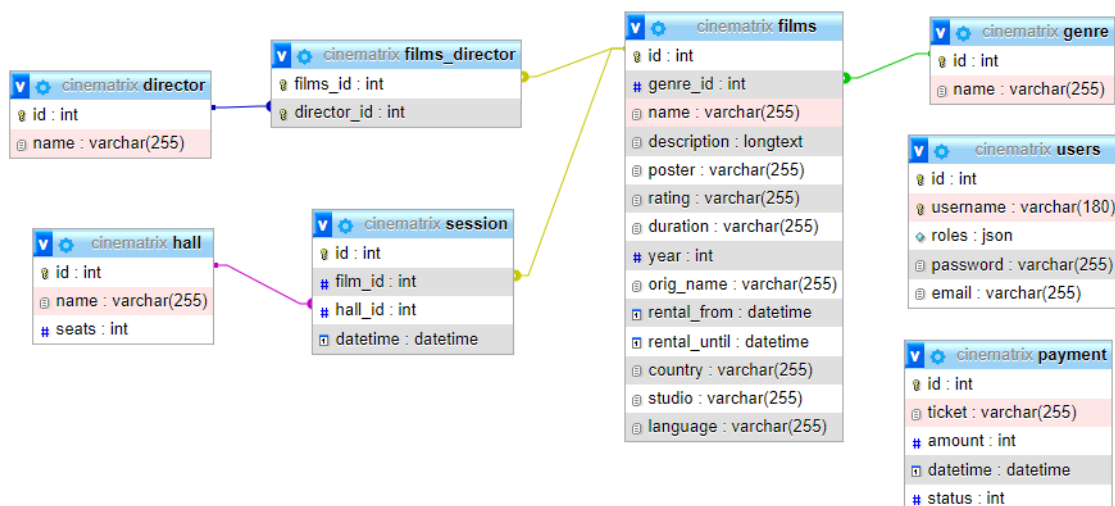


Рисунок 2.2 - Діаграма бази даних

На рисунку 2.3 зображено приклад міграції в php форматі. Symfony використовує Doctrine Migrations для керування міграціями бази даних. Цей компонент дозволяє створювати та застосовувати міграції, які описують зміни в структурі бази даних. Кожна міграція представляє собою клас, який містить два методи: up() та down(). Метод up() містить код, який виконує зміни в базі даних для застосування міграції, тоді як метод down() містить код для відкату цих змін.

Команда `doctrine:migrations:migrate` використовується для застосування нових міграцій до бази даних. Під час виконання цієї команди всі міграції, які ще не були застосовані, будуть виконані в порядку їх створення. Symfony також надає можливість відкату міграцій за допомогою команди `doctrine:migrations:execute`, яка виконує метод `down()` кожної міграції в зворотньому порядку. Загалом, механізм міграцій в Symfony забезпечує зручний та безпечний спосіб керування структурою бази даних у веб-додатках.

```

↑ Vlad Koida
public function up(Schema $schema): void
{
    // this up() migration is auto-generated, please modify it to your needs
    $this->addSql('CREATE TABLE director (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci`');
    $this->addSql('CREATE TABLE films (id INT AUTO_INCREMENT NOT NULL, genre_id INT NOT NULL, name VARCHAR(255) NOT NULL, description VARCHAR(255) NOT NULL, poster VARCHAR(255) NOT NULL, INDEX IDX_51FD28FD939610EE (films_id), INDEX IDX_51FD28FD899FB366 (genre_id)');
    $this->addSql('CREATE TABLE films_director (films_id INT NOT NULL, director_id INT NOT NULL, INDEX IDX_51FD28FD939610EE (films_id), INDEX IDX_51FD28FD899FB366 (director_id)');
    $this->addSql('CREATE TABLE hall (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, seats INT NOT NULL, accessibility INT NOT NULL, PRIMARY KEY(id))');
    $this->addSql('CREATE TABLE payment (id INT AUTO_INCREMENT NOT NULL, ticket_id INT NOT NULL, amount INT NOT NULL, datetime DATETIME NOT NULL, status INT NOT NULL, INDEX IDX_51FD28FD899FB366 (ticket_id), INDEX IDX_51FD28FD939610EE (id)');
    $this->addSql('CREATE TABLE place (id INT AUTO_INCREMENT NOT NULL, hall_id INT NOT NULL, number INT NOT NULL, type INT NOT NULL, price INT NOT NULL, INDEX IDX_51FD28FD939610EE (hall_id), INDEX IDX_51FD28FD899FB366 (number)');
    $this->addSql('CREATE TABLE session (id INT AUTO_INCREMENT NOT NULL, film_id INT NOT NULL, hall_id INT NOT NULL, datetime DATETIME NOT NULL, duration VARCHAR(255) NOT NULL, PRIMARY KEY(id))');
    $this->addSql('CREATE TABLE user (id INT AUTO_INCREMENT NOT NULL, username VARCHAR(255) NOT NULL, password VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL, PRIMARY KEY(id))');
    $this->addSql('CREATE TABLE messenger_messages (id BIGINT AUTO_INCREMENT NOT NULL, body LONGTEXT NOT NULL, headers LONGTEXT NOT NULL, queue_name VARCHAR(190) NOT NULL, PRIMARY KEY(id))');
    $this->addSql('ALTER TABLE films ADD CONSTRAINT FK_CEECCA514296031F FOREIGN KEY (genre_id) REFERENCES genre (id)');
    $this->addSql('ALTER TABLE films_director ADD CONSTRAINT FK_51FD28FD939610EE FOREIGN KEY (films_id) REFERENCES films (id) ON DELETE CASCADE');
    $this->addSql('ALTER TABLE films_director ADD CONSTRAINT FK_51FD28FD899FB366 FOREIGN KEY (director_id) REFERENCES director (id) ON DELETE CASCADE');
    $this->addSql('ALTER TABLE payment ADD CONSTRAINT FK_6D28840070004702 FOREIGN KEY (ticket_id) REFERENCES ticket (id)');
    $this->addSql('ALTER TABLE place ADD CONSTRAINT FK_741D53CD52AFCF06 FOREIGN KEY (hall_id) REFERENCES hall (id)');
    $this->addSql('ALTER TABLE session ADD CONSTRAINT FK_D044D5D4567F5183 FOREIGN KEY (film_id) REFERENCES films (id)');
    $this->addSql('ALTER TABLE session ADD CONSTRAINT FK_D044D5D452AFCF06 FOREIGN KEY (hall_id) REFERENCES hall (id)');
    $this->addSql('ALTER TABLE ticket ADD CONSTRAINT FK_97A0ADA3613FECDF FOREIGN KEY (session_id) REFERENCES session (id)');
    $this->addSql('ALTER TABLE ticket ADD CONSTRAINT FK_97A0ADA3A76ED395 FOREIGN KEY (user_id) REFERENCES user (id)');
    $this->addSql('ALTER TABLE ticket ADD CONSTRAINT FK_97A0ADA3DA6A219 FOREIGN KEY (place_id) REFERENCES place (id)');
    $this->addSql('CREATE INDEX IDX_97A0ADA3613FECDF ON ticket (session_id)');
    $this->addSql('CREATE INDEX IDX_97A0ADA3A76ED395 ON ticket (user_id)');
    $this->addSql('CREATE INDEX IDX_97A0ADA3DA6A219 ON ticket (place_id)');
}

```

Рисунок 2.3 - Приклад міграції

Ця міграція створює базову структуру бази даних для додатку. Вона включає такі таблиці:

1. **Таблиця director:** Зберігає дані про режисерів фільмів.
2. **Таблиця films:** Містить інформацію про фільми, таку як назва, опис, постер, рейтинг тощо.
3. **Таблиця films_director:** Таблиця зв'язку між фільмами та їх режисерами.
4. **Таблиця hall:** Відображає інформацію про кінозали.
5. **Таблиця payment:** Записує дані про оплати за квитки.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

6. **Таблиця session:** Зберігає дані про сеанси показу фільмів.
7. **Таблиця user:** Містить інформацію про користувачів системи.
8. **Індекси та зовнішні ключі:** Встановлюють зв'язки між таблицями та полегшують швидкий доступ до даних.

Детальніше розглянемо кожну таблицю, що створюється завдяки цій міграції.

1. Таблиця director:

- **id (INT AUTO_INCREMENT):** Унікальний ідентифікатор режисера.
- **name (VARCHAR(255)):** Ім'я режисера.

2. Таблиця films:

- **id (INT AUTO_INCREMENT):** Унікальний ідентифікатор фільму.
- **genre_id (INT):** Ідентифікатор жанру фільму (зв'язок з іншою таблицею).
- **name (VARCHAR(255)):** Назва фільму.
- **description (VARCHAR(255)):** Опис фільму.
- **poster (VARCHAR(255)):** Шлях до постера фільму.
- **rating (VARCHAR(255)):** Рейтинг фільму.

3. Таблиця films_director:

- **films_id (INT):** Ідентифікатор фільму (зв'язок з таблицею **films**).
- **director_id (INT):** Ідентифікатор режисера (зв'язок з таблицею **director**).

4. Таблиця hall:

- **id (INT AUTO_INCREMENT):** Унікальний ідентифікатор кінозалу.
- **name (VARCHAR(255)):** Назва кінозалу.
- **seats (INT):** Кількість сидячих місць у кінозалі.
- **accessibility (INT):** Доступність кінозалу (наприклад, для інвалідів).

5. Таблиця payment:

- **id (INT AUTO_INCREMENT):** Унікальний ідентифікатор оплати.
- **ticket_id (INT):** Ідентифікатор квитка (зв'язок з іншою таблицею).
- **amount (INT):** Сума оплати.
- **datetime (DATETIME):** Дата та час оплати.
- **status (INT):** Статус оплати.

6. Таблиця session:

- **id (INT AUTO_INCREMENT):** Унікальний ідентифікатор сеансу.
- **film_id (INT):** Ідентифікатор фільму (зв'язок з таблицею **films**).
- **hall_id (INT):** Ідентифікатор кінозалу (зв'язок з таблицею **hall**).
- **datetime (DATETIME):** Дата та час проведення сеансу.
- **duration (VARCHAR(255)):** Тривалість сеансу.

7. Таблиця user:

- **id (INT AUTO_INCREMENT):** Унікальний ідентифікатор користувача.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

- **username (VARCHAR(255))**: Ім'я користувача.
- **password (VARCHAR(255))**: Пароль користувача.
- **email (VARCHAR(255))**: Email користувача.

Архітектура бази даних відіграє ключову роль у процесі розробки проекту. Від цього етапу залежить легкість доступу до даних, швидкість виконання запитів і оптимальне використання ресурсів обраної бази даних.

2.3. Проектування та реалізація алгоритмів роботи системи

Діаграма послідовності - це інструмент в рамках мови моделювання UML, який використовується для візуалізації послідовності взаємодії між об'єктами або компонентами системи в часовому аспекті. Вона розкриває, як об'єкти взаємодіють один з одним, передаючи повідомлення та виконуючи певні дії, що дозволяє краще розуміти логіку системи та виявляти можливі проблеми у взаємодії між компонентами.

Діаграми послідовності служать для уявлення алгоритмів функціонування системи з точки зору взаємодії між її складовими частинами. Вони фокусуються на порядку передачі повідомлень та виконанні дій між об'єктами системи протягом певного часу. Це допомагає розуміти послідовність виконання операцій, обмін даними та взаємозалежності між компонентами. За допомогою діаграм послідовності можна виявити можливі проблеми або неузгодженості в системі та знайти оптимальні шляхи їх вирішення.

Основна мета системи кінотеатру полягає в тому, щоб показати клієнту різноманіття нових фільмів в прокаті, та дати змогу замовити собі квиток онлайн, тим самим забезпечивши собі дозвілля.

На рисунку 2.4 зображено діаграму послідовності системи кінотеатру.



Рисунок 2.4 - Діаграма послідовності системи кінотеатру

Діаграма послідовності системи кінотеатру ілюструє взаємодію різних компонентів системи під час процесу вибору фільму користувачем та замовленням квитка.

При відвідуванні сайту користувач відразу повинен бачити список фільмів, які доступні в прокаті на даний момент. Він обирає зацікавивший його фільм та переходить на сторінку з інформацією про даний фільм. Далі користувач має змогу замовити квиток на фільм, який він обрав. Після успішної проплати за допомогою платіжної системи сервер генерує квиток та видає його користувачу, при цьому зберігаючи інформацію про квиток в БД.

2.4. Реалізація веб-додатку

Реалізація веб-додатку на Symfony починається з налаштування з'єднання з базою даних в конфігураційному файлі `.env`. Встановлюємо настройки для з'єднання з базою даних: назва БД, користувач, пароль. Налаштування з'єднання з БД в конфігураційному файлі зображено на рисунку 2.5.

```
APP_ENV=dev
APP_SECRET=99ffa6ff53c106b16e20975cd3ce1cc9
DATABASE_URL="mysql://root:root@127.0.0.1:3306/cinematrix?serverVersion=8.0.30&charset=utf8mb4"
MESSENGER_TRANSPORT_DSN=doctrine://default?auto_setup=0
```

Рисунок 2.5 - Конфігураційний файл `.env`

Symfony використовує патерн проектування MVC (Model-View-Controller), це означає, що логіка додатку буде розподілена між моделями, контролерами та шаблонами.

Модель в Symfony представляє бізнес-логіку та дані додатку. Вона взаємодіє з базою даних або будь-якими іншими джерелами даних, забезпечуючи методи для отримання, збереження, оновлення та видалення даних. Моделі зазвичай розташовуються у директорії `app/Models`. Laravel використовує ORM (Object-Relational Mapping) Eloquent, що дозволяє вам легко взаємодіяти з базою даних за допомогою РНР-класів. Моделі також можуть включати валідацію даних та інші дії, пов'язані з бізнес-логікою.

Для кожної таблиці бази даних були створені моделі для взаємодії з ними. Створені моделі зображені на рисунку 2.6

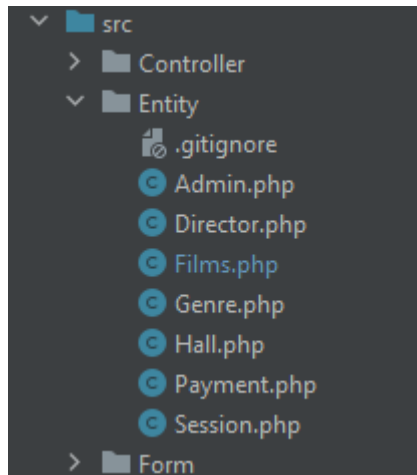


Рисунок 2.6 - Створені моделі

Модель Admin – це модель, що відповідає за дані, що знаходяться в таблиці users.

```
<?php

namespace App\Entity;

use App\Repository\AdminRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;
use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;
use Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface;
use Symfony\Component\Security\Core\User\UserInterface;

#[ORM\Entity(repositoryClass: AdminRepository::class)]
#[ORM\Table(name: '`users`')]
#[UniqueEntity(fields: ['username'], message: 'There is already an account with this username')]
class Admin implements UserInterface, PasswordAuthenticatedUserInterface
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 180, unique: true)]
    private ?string $username = null;

    #[ORM\Column]
    private array $roles = [];

    /**
     * @var string The hashed password
     */
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

```

#[ORM\Column(length: 255)]
private ?string $password = null;

#[ORM\Column(length: 255)]
private ?string $email = null;

public function __construct()
{
    $this->tickets = new ArrayCollection();
}

public function getId(): ?int
{
    return $this->id;
}

public function getUsername(): ?string
{
    return $this->username;
}

public function setUsername(string $username): static
{
    $this->username = $username;

    return $this;
}

/**
 * A visual identifier that represents this user.
 *
 * @see UserInterface
 */
public function getUserIdentifier(): string
{
    return (string) $this->username;
}

/**
 * @see UserInterface
 */
public function getRoles(): array
{
    $roles = $this->roles;
    // guarantee every user at least has ROLE_USER
    $roles[] = 'ROLE_USER';

    return array_unique($roles);
}

public function setRoles(array $roles): static
{
    $this->roles = $roles;
}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        return $this;
    }

    /**
     * @see PasswordAuthenticatedUserInterface
     */
    public function getPassword(): string
    {
        return $this->password;
    }

    public function setPassword(string $password): static
    {
        $this->password = $password;

        return $this;
    }

    public function getEmail(): ?string
    {
        return $this->email;
    }

    public function setEmail(string $email): static
    {
        $this->email = $email;

        return $this;
    }

    /**
     * @see UserInterface
     */
    public function eraseCredentials(): void
    {
        // If you store any temporary, sensitive data on the user, clear it here
        // $this->plainPassword = null;
    }
}

```

Цей клас описує адміністратора системи кінотеатру. Його поля включають ідентифікатор (id), ім'я користувача (username), ролі користувача (roles), пароль (password), і електронну адресу (email). Крім того, адміністратор може мати доступ до списку фільмів через зв'язок OneToMany з класом Movie.

Контролер в Symfony відповідає за обробку запитів користувача та взаємодію з моделями та шаблонами. Він приймає запити, викликає необхідні методи моделей для отримання даних та передає ці дані до відповідних

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

шаблонів для відображення. Контролери зазвичай розташовуються у директорії “src/Controller”. Вони можуть мати різні методи для різних дій, таких як показ списку даних, створення нового запису, відображення конкретного запису тощо.

Контролер аутентифікації відповідає за управління процесом входу та виходу користувачів з системи. У цьому розділі ми представляємо клас AuthController, який містить методи для обробки входу (login) та виходу (logout) користувачів.

Розглянемо код контролера AuthController

```
<?php
namespace App\Controller;

use App\Entity\User;
use App\Form\LoginButtonType;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class SecurityController extends AbstractController
{
    #[Route('/login', name: 'login')]
    public function login(Request $request, EntityManagerInterface $em,
        AuthenticationUtils $authenticationUtils): Response
    {
        $form = $this->createForm(LoginButtonType::class);

        $form->handleRequest($request);
        if ($form->isSubmitted() && $form->isValid()) {
            $data = $form->getData();

            // Проверка пользователя в базе данных
            $user = $em->getRepository(User::class)->findOneBy(['username' =>
                $data['username']]);
            $hashedPassword = md5($data['password']);
            if (!$user || $user->getPassword() !== $hashedPassword) {
                $this->addFlash('error', 'Invalid credentials');
                return $this->redirectToRoute('login');
            }

            if ($user->getAdmin() === 1) {
                // Пользователь является администратором, выполните необходимые
                // действия (например, редирект на админскую панель)
                return $this->redirectToRoute('admin_dashboard');
            }
        }
    }
}
```

```

    }
    $this->addFlash('success', 'Success');
    // Пользователь не является администратором, выполните необходимые
    действия (например, редирект на обычную панель пользователя)
    return $this->redirectToRoute('app_index');
}
$error = $authenticationUtils->getLastAuthenticationError();
$lastUsername = $authenticationUtils->getLastUsername();
$temporaryMessage = 'Ласкаво просимо! Ви успішно увійшли в систему.';
return $this->render('security/login.html.twig', [
    'form' => $form->createView(),
    'temporaryMessage' => $temporaryMessage,
    'last_username' => $lastUsername,
    'error' => $error
]);
}

#[Route(path: '/logout', name: 'app_logout')]
public function logout(): void
{
    throw new \LogicException('This method can be blank - it will be
intercepted by the logout key on your firewall.');
```

Цей контролер містить два методи:

Опис методу login():

Метод login() відповідає за обробку запиту на вхід користувача на сайт. Він приймає об'єкт Request, EntityManagerInterface для взаємодії з базою даних та AuthenticationUtils для отримання даних про аутентифікацію. Далі:

- Створюється форма входу з використанням класу LoginFormType.
- Обробляється надісланий запит: перевіряється, чи дані коректні, та здійснюється спроба аутентифікації користувача.
- Проводиться перевірка користувача в базі даних, порівнюються хешований пароль користувача та введений пароль.
- Якщо користувач успішно автентифікується, він перенаправляється на відповідну сторінку: адмінську панель для адміністраторів або головну сторінку для звичайних користувачів.
- У випадку невірних даних відображається повідомлення про помилку.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

- Останній введений користувачем логін зберігається для зручності.

Метод logout():

Метод logout() використовується для обробки запиту на вихід користувача з системи. У даному випадку він не має жодної функціональності, оскільки обробка виходу користувача відбувається автоматично на рівні файрволу.

Цей контролер є важливою частиною системи аутентифікації в Symfony-проекті для кінотеатру. Він забезпечує відповідні маршрути для входу та виходу користувачів і відповідає за відображення відповідних сторінок.

Структура папки src/Controller зображена на рисунку 2.7.

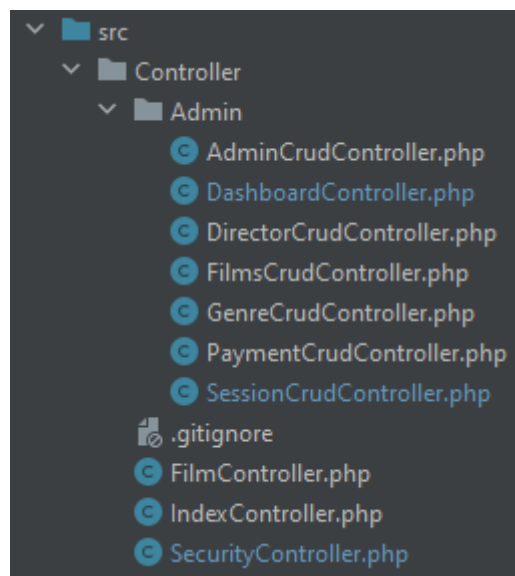


Рисунок 2.7 - Структура папки Controllers

В підпапці Admin знаходяться контролери для роботи адмін панелі.

В таблиці 2.1 наведено пояснення про основні контролери та їх основні задачі.

Таблиця 2.1

Контролери та їх основні задачі

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

№	Назва контролера	Основні задачі
1	FilmController	Відповідає за відображення детальної інформації про фільм.
2	IndexController	Основний контролер, обробляє всі фільми і виводить їх для користувача.
3	SecurityController	Відповідає за авторизацію адміністраторів на сайті.

Під час створення контролерів створювались відразу і маршрути.

Висновки до розділу 2

В даному розділі було спроектовано діаграми використання та діаграми послідовності. Проведено розробку загальної схеми роботи додатку. Розглянуто та спроектовано алгоритми роботи додатку.

Наведені деякі частини коду. Були створені такі можливості: створення та редагування фільмів, вхід адміністрації, замовлення квитків, перегляд списку фільмів.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБ-САЙТОМ ПОШУКУ РОБОТИ

3.1. Порядок встановлення та налаштування системи

Діаграма розгортання в UML використовується для моделювання фізичного розгортання артефактів на вузлах системи. Вузли зображаються у вигляді прямокутних паралелепіпедів, в яких розташовані артефакти, що зображаються у вигляді прямокутників. Вузли можуть мати підвузли, які представляються як вкладені прямокутні паралелепіпеди.

Діаграма розгортання надає можливість візуалізувати фізичне розташування компонентів системи на фізичних вузлах, таких як сервери, комп'ютери або пристрої IoT. Кожен вузол може мати свої власні артефакти, такі як програми, файли, конфігураційні файли тощо, які виконуються або розгортаються на цьому вузлі.

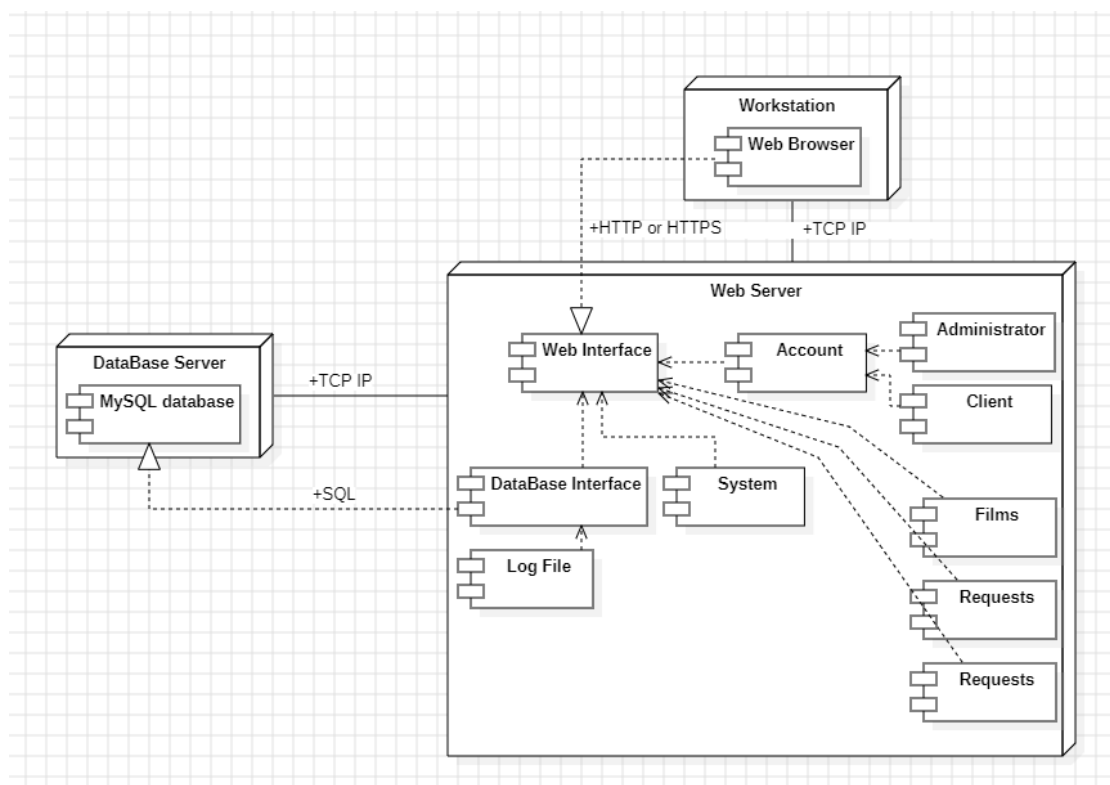


Рисунок 3.1 - Діаграма розгортання системи

Вузли:

- Database Server – сервер бази даних
- Workstation – персональна ЕОМ користувача
- Web Server – веб сервер

Діаграма компонентів дозволяє описати фізичне уявлення системи і визначити її архітектуру, встановлюючи залежності між компонентами програмного забезпечення. Цей графічний інструмент використовує компоненти, інтерфейси та залежності як основні елементи для відображення структури та взаємозв'язків системи.

Важливо відзначити, що діаграма компонентів є одним із ключових інструментів при розробці програмного забезпечення, оскільки вона дозволяє зрозуміти структуру системи та забезпечити її правильну організацію та функціонування.

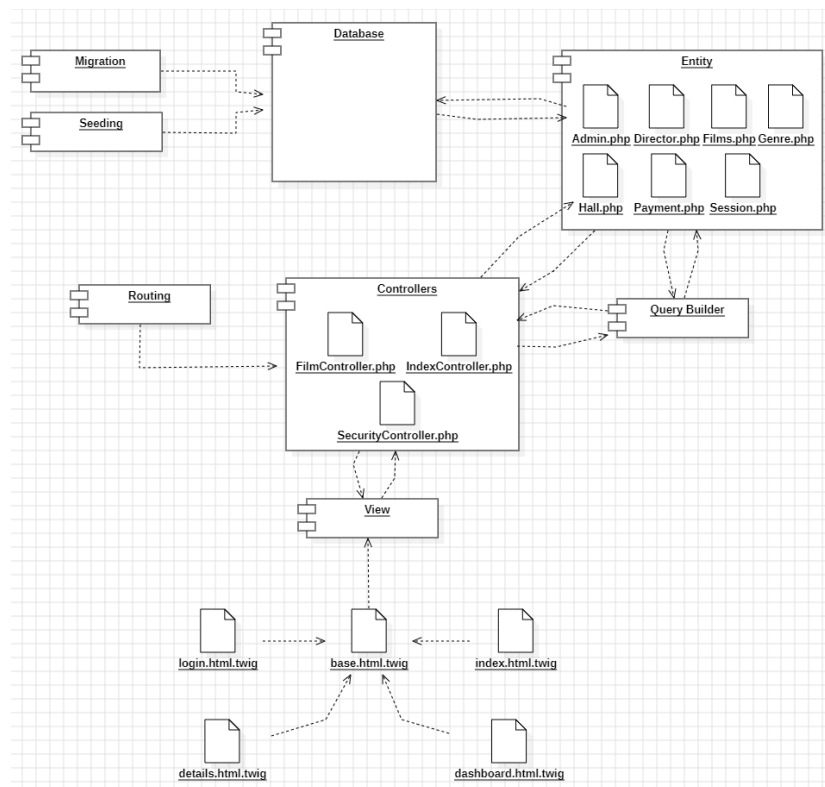


Рисунок 3.2 - Діаграма компонентів

На діаграмі компонентів зображено такі елементи:

1. Routing – маршрутизатор Symfony;
2. Database – база даних;
3. Migrations – спеціальні класи, що створюють структуру таблиць бази даних;
4. Entities – сутності для керування таблицями баз даних.
5. Query Builder – спеціальний клас, який дозволяє будувати запити до бази даних;
6. Controllers - це класи, які відповідають за обробку вхідних запитів, координування взаємодії з моделями та відображенням результатів у відповідні вигляди (views);
7. View - це компонент, який відповідає за візуальне представлення даних інтерфейсу користувача, де відображаються дані з моделей та інші елементи інтерфейсу.

Використання діаграми компонентів дозволяє детально розглянути фізичну організацію системи та встановити взаємозв'язки між різними компонентами програмного забезпечення. Вона надає зрозуміле та графічне представлення архітектури системи, що допомагає розробникам та інженерам зрозуміти взаємодію між компонентами та забезпечити їх ефективну інтеграцію.

3.2. Структура інтерфейсу веб-сайту кінотеатру

Спершу, що потрібно зробити – це перейти на сам сайт. Користувач має можливість переглядати контент сайту без реєстрації, після авторизації користувача «Адміністратор» можливості розширюються до керування фільмами, та всим що їх стосується. Реєстрація відсутня, так як авторизація доступна тільки для Адміністрації кінотеатру.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

Please sign in

Username

Password

Sign in

Рисунок 3.3 – Форма авторизації для адміністратора

Після успішного входу здійснюється перенаправлення на адмін панель сайту. На цій сторінці знаходиться все необхідне, щоб керувати наповненістю сайту.

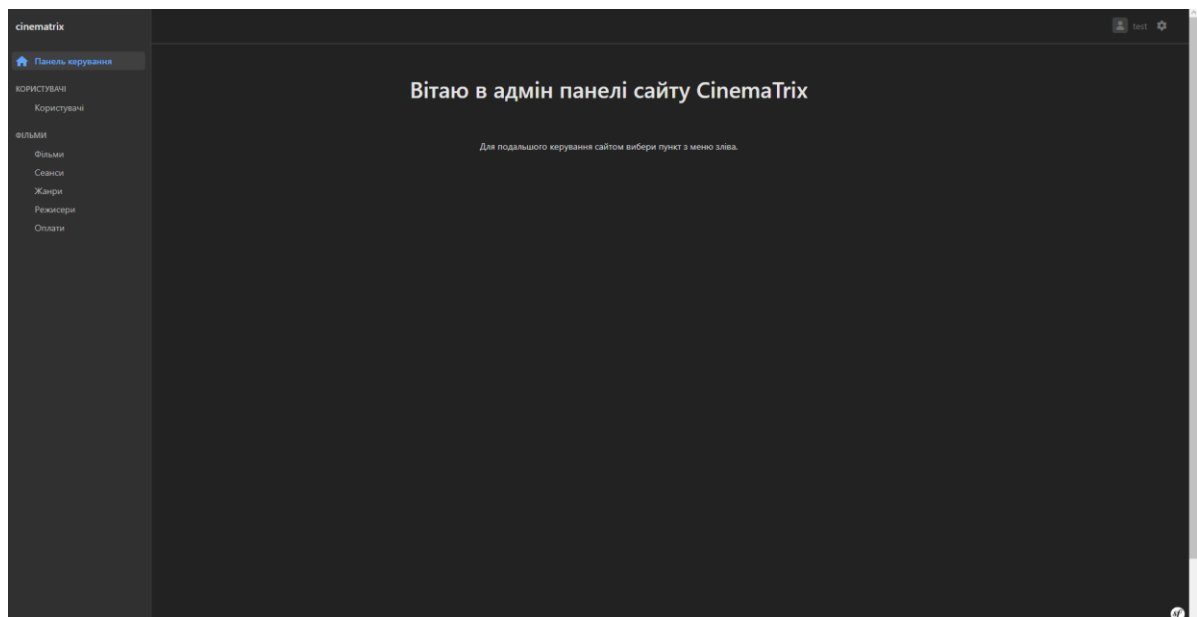


Рисунок 3.4 – Адмін панель сайту

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

У пункті меню «Користувачі» можна побачити всіх існуючих користувачів та додати нових та видалити або відредагувати існуючих.

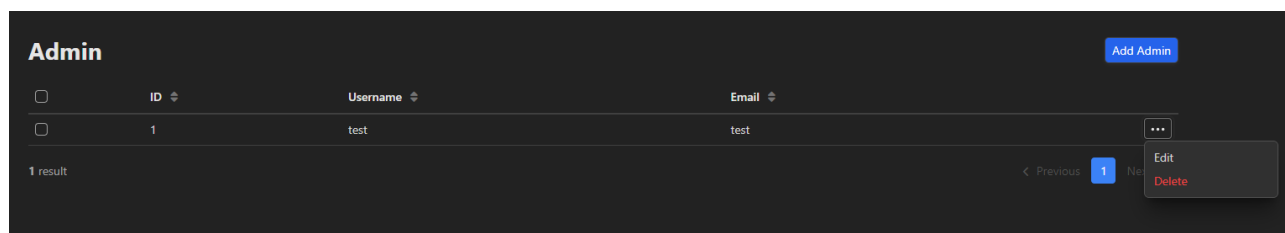


Рисунок 3.5 – Пункт «Користувачі»

Рисунок 3.6 – Додавання нового користувача

Інші пункти меню працюють ідентично пункту «Користувачі», в них всі ті самі можливості взаємодії, тільки підлаштовані для інших таблиць БД.

При переході на головну сторінку сайту можна побачити список фільмів та основне правило кінотеатру.

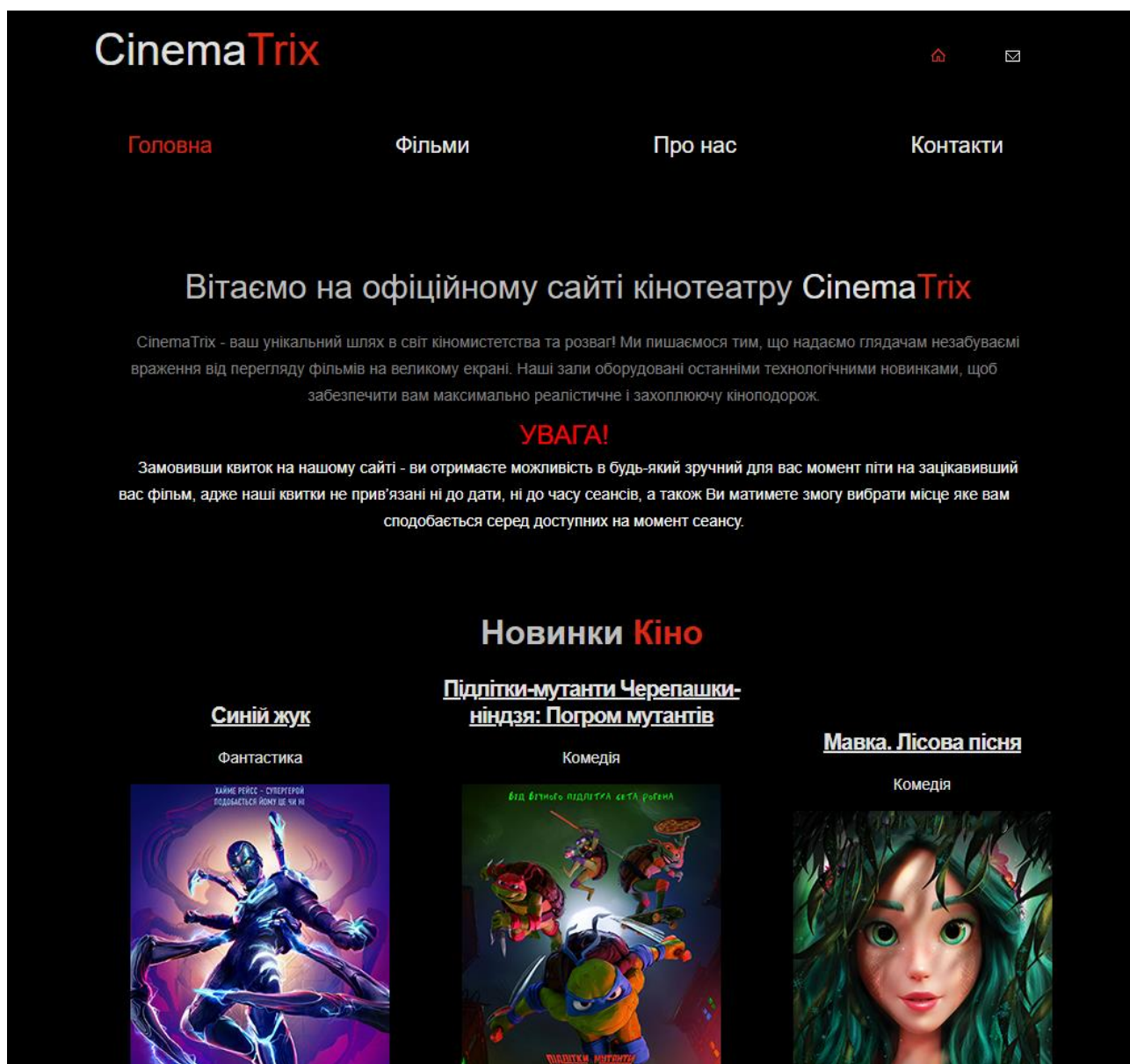


Рисунок 3.7 – Головна сторінка

Під кожним фільмом вказані всі доступні часи сеансів для даного фільму, щоб користувач міг обрати підходящий йому час для відвідування кінотеатру, а також вказано дату закінчення прокату фільму.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Новинки Кіно

Синій жук

Фантастика



Тривалість: 2 год. 7 хв.

Випускник коледжу Хайме Рейс повертається до рідного дому, сповнений надій на майбутнє. Однак дім тепер – зовсім не те місце, яким він його пам'ятає. Поки хлопець шукає своє призначення, доля втручає ...

В прокаті до: 29.02.2024

Доступні сеанси:

| 14:30 | | 20:45 | | 23:25 |

Підлітки-мутанти Черепашки-ніндзя: Погром мутантів

Комедія



Тривалість: 1 год. 33 хв.

Довгий час вони ховалися від навколишнього світу в міській каналізації. Проте нарешті герої вирішують піднятися нагору, щоб жити життям звичайних підлітків серед жителів Нью-Йорку. Та зовсім скоро їм ...

В прокаті до: 25.02.2024

Доступні сеанси:

| 11:00 | | 17:10 |

Мавка. Лісова пісня

Комедія



Тривалість: 1 год. 30 хв.

Безмежні українські ліси з давніх-давен приховують безліч таємниць і загадок. Тут, серед столітніх дерев, живуть незвичайні міфічні істоти, а серед них - Мавка, Душа і новообрана Берегиня Лісу. Її го ...

В прокаті до: 17.02.2024

Доступні сеанси:

| 08:00 | | 10:00 | | 16:00 |

Більше фільмів...

Рисунок 3.8 – Список фільмів на головній сторінці

На головній сторінці зображено всього три фільми, які були добавлені останніми, повний список фільмів знаходиться за посиланням «Більше фільмів...», також на сторінку з фільмами можна потрапити натиснувши пункт меню «Фільми» відповідно. На цій сторінці буде відображено всі доступні в прокаті фільми з відображенням сеансів кожного з них.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Головна

Фільми

Про нас

Контакти

Синій жук

Фантастика

Тривалість: 2 год. 7 хв.

Випускник коледжу Хайме Рейс повертається до рідного дому, сповнений надій на майбутнє. Однак дім тепер – зовсім не те місце, яким він його пам'ятає. Поки хлопець шукає своє призначення, доля втручає...

В прокаті до: 29.02.2024

Доступні сеанси:

| 14:30 | | 20:45 | | 23:25 |

Гран туризмо

Драма

Підлітки-мутанти Черепашки-ніндзя: Погром мутантів

Комедія

Тривалість: 1 год. 33 хв.

Довгий час вони ховалися від навколишнього світу в міській каналізації. Проте нарешті герої вирішують піднятися нагору, щоб жити життям звичайних підлітків серед жителів Нью-Йорку. Та зовсім скоро їм...

В прокаті до: 25.02.2024

Доступні сеанси:

| 11:00 | | 17:10 |

Оппенгеймер

Історичний

Мавка. Лісова пісня

Комедія

Тривалість: 1 год. 30 хв.

Безмежні українські ліси з давніх-давен приховують безліч таємниць і загадок. Тут, серед столітніх дерев, живуть незвичайні міфічні істоти, а серед них - Мавка, Душа і новообрана Берегиня Лісу. Її го...

В прокаті до: 17.02.2024

Доступні сеанси:

| 08:00 | | 10:00 | | 16:00 |

Рисунок 3.9 – Сторінка з всіма фільмами

При натисканні на фільм відбудеться перехід на сторінку з детальним описом фільму. На цій сторінці виведена вся інформація, що може зацікавити користувача. Також тут знаходиться кнопка замовлення квитка.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46



Синій жук

Жанр: Фантастика

Тривалість: 2 год. 7 хв.

Опис: Випускник коледжу Хайме Рейес повертається до рідного дому, сповнений надій на майбутнє. Однак дім тепер – зовсім не те місце, яким він його пам'ятає. Поки хлопець шукає своє призначення, доля втручається. Хайме несподівано стає власником Скарабей – стародавньої реліквії інопланетних біотехнологій. Скарабей обирає його своїм симбіотичним господарем та обдаровує неймовірним обладунком. Тепер Хайме, одягнений в обладунок, має надзвичайні та непередбачувані здібності, що назавжди змінюють його життя. Хайме стає супергероєм – Синім Жуком.

Вікові обмеження: 12+

Режисери:

Рік: 2023

Оригінальна назва: Blue Beetle

Дата прокату: 17.02.2024 - 29.02.2024

Країна: Blue Beetle

Студія: Warner Bros. Pictures, Safran Company Production

Мова: українська мова

Забронювати зараз

Рисунок 3.10 - Сторінка-резюме пошукача

Після натискання кнопки «Забронювати зараз» відкриється вікно замовлення квитка. У вікні замовлення буде назва фільму та попередження що вибір місця в залі є вільним.

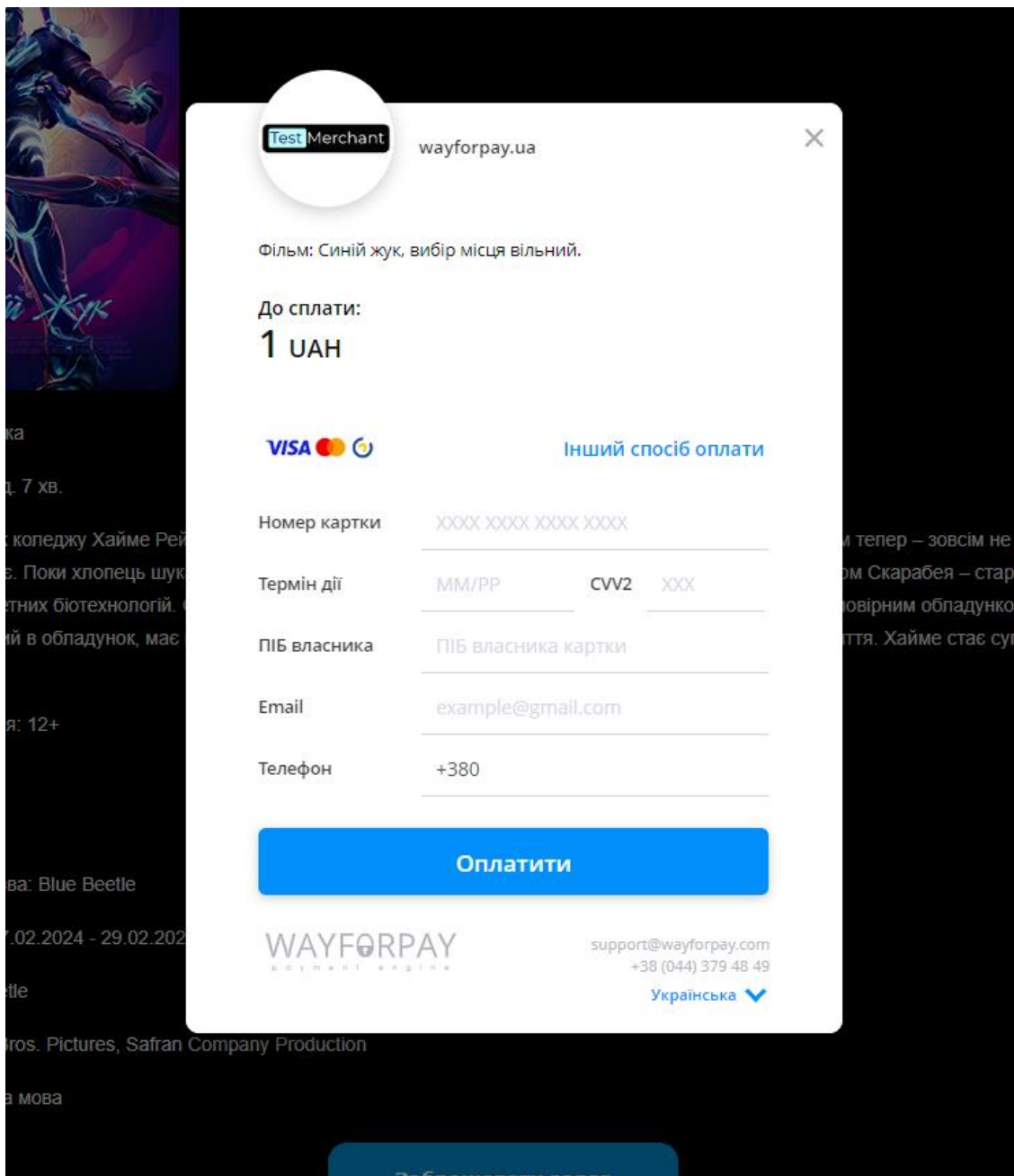


Рисунок 3.11 – Вікно замовлення

Після успішної оплати буде виведено повідомлення про це та відразу почнеться завантаження купленого квитка.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

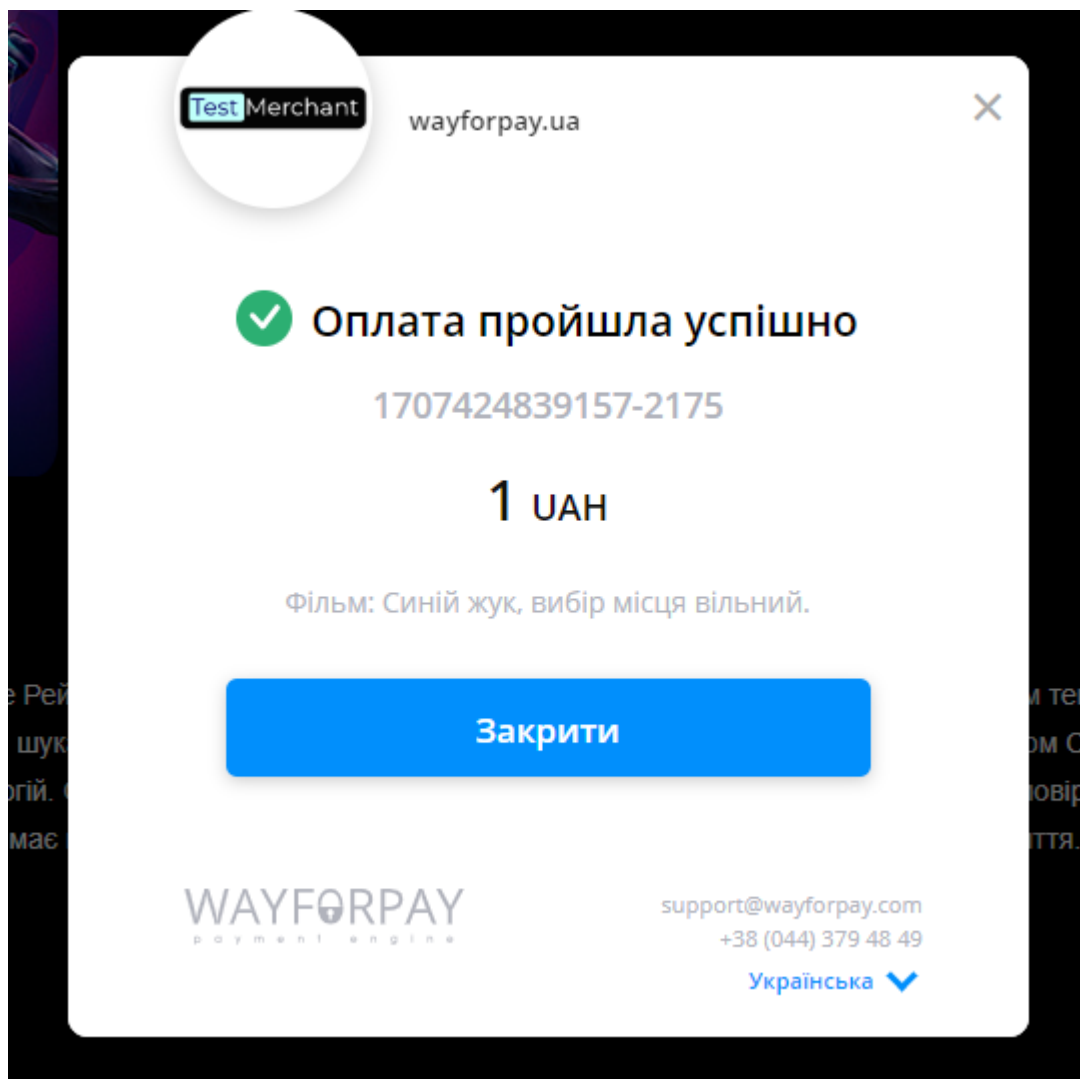


Рисунок 3.12 – Успішна оплата

В квитку зображено тільки найважливішу інформацію для кінотеатру, а саме: назва фільму та номер квитка. При цьому номер квитка генерується автоматично, він є унікальним.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

Ticket

Film: Blue Beetle

Location: the choice of location is free

Ticket number: cnmtrx_65c53cfa932a32.98590924

Рисунок 3.13 – Вигляд завантаженого квитка

При цьому цей номер автоматично внесеться в БД для можливості перевірки його в кінотеатрі.

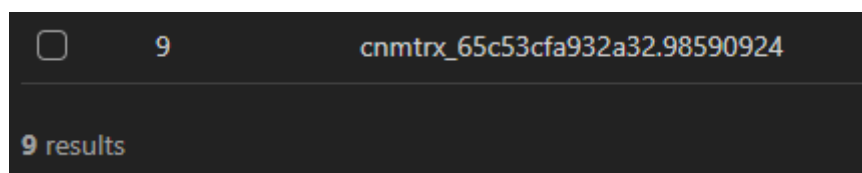


Рисунок 3.14 – Номер квитка в БД

3.3. Тестування роботи веб-сайту

Тестування програмного забезпечення (англ. Software Testing) — це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою оцінки.

Тестування програмного забезпечення - процес перевірки відповідності заявлених до продукту вимог і реально реалізованої функціональності, здійснюваний шляхом спостереження за його роботою в штучно створених ситуаціях і на обмеженому наборі тестів, обраних певним чином.

Може оцінюватись:

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

- відповідність вимогам, якими керувалися проектувальники та розробники
- правильна відповідь для усіх можливих вхідних даних
- виконання функцій за прийнятний час
- практичність
- сумісність з програмним забезпеченням та операційними системами
- відповідність задачам замовника.

Після завершення розроблення веб-застосунку було проведено тестування на відповідність вимогам, поставленим у даній роботі. Тестування функціональних можливостей та роботи елементів веб-сайту проводилося за допомогою ручного тестування. Після даного тестування було виявлено деякі помилки та виправлено.

Для тестування веб-сайту пошуку роботи написаному на Symfony Framework було використано Symfony Profiler.

Symfony Profiler - це інструмент, який надається Symfony, PHP-фреймворком для створення веб-додатків, і призначений для допомоги розробникам у відлагодженні та профілюванні додатків. Symfony Profiler надає докладну інформацію про запити до додатку, включаючи час виконання, використану пам'ять, виконані SQL-запити, а також інші відлагоджувальні та профілювальні дані.

Profiler доступний під час розробки, коли увімкнений режим відлагодження в Symfony, і зазвичай доступний за адресою «/_profiler» у додатку. Він може бути дуже корисним для ідентифікації проблем у продуктивності, виявлення "вузьких місць" у коді та оптимізації запитів до бази даних.

На рисунку 3.15 зображено інтерфейс Symfony Profiler.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

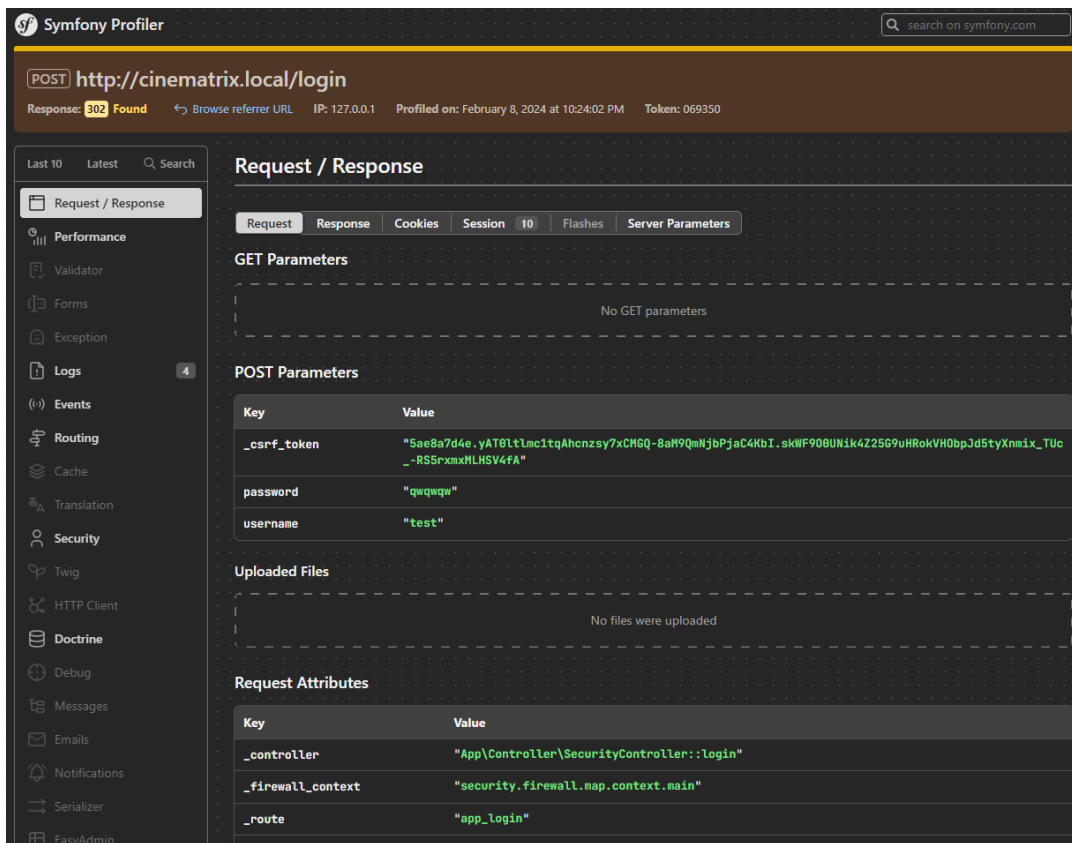


Рисунок 3.16 – Запит POST до сторінки авторизації

Висновки до розділу 3

У цьому розділі була розроблена архітектура веб-сайту кінотеатру. Завдяки проектуванню діаграм компонентів та розгортання було визначено основні об'єкти моделей у системі.

Було описано інтерфейс веб-сайту. На основі опису інтерфейсу зрозумілі можливості користувача, а саме: для глядачів можливість увійти на сайт, переглянути розклад сеансів, забронювати квитки та здійснювати онлайн-оплату. Для адміністраторів кінотеатру - це можливість додавати нові фільми до розкладу, керувати сеансами та переглядати статистику продажів.

Було проведено тестування веб-сайту за допомогою спеціального програмного забезпечення. виправлено незначні помилки. Сайт кінотеатру працює коректно та виконує поставлені завдання.

ВИСНОВКИ

Під час розробки веб-проєкту для кінотеатру було здійснено значну кількість робіт з метою виконання завдань проєкту. Було ретельно проаналізовано існуючі аналоги та визначено можливий функціонал для проєкту. Далі, враховуючи вимоги користувачів та функціональні та нефункціональні вимоги, був проведений детальний аналіз.

За допомогою діаграм варіантів використання та компонентів було чітко відображено, як користувачі взаємодіють з системою та які функції вони використовують. Особливості фізичного представлення системи були описані через діаграму компонентів, що дозволило встановити залежності між програмними компонентами.

Було розроблено загальний і функціональний алгоритми веб-додатку, а також проведено тестування, під час якого були виявлені та виправлені помилки.

Проєкт відкриває широкі можливості для подальшого вдосконалення. Наприклад, можливе оновлення інтерфейсу, інтеграція з сервісами штучного інтелекту для покращеного аналізу даних, впровадження авторизації через соціальні мережі та створення стрічки новин.

Розробка була завершена в повному обсязі та відповідає всім вимогам. Обрана архітектура дозволить легко підтримувати та розвивати продукт у майбутньому без серйозних ускладнень.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «Ознакомлення з патерном MVC» [Електронний ресурс] – Режим доступу до ресурсу: <https://javarush.com/ua/groups/posts/uk.2536.chastina-7-oznayomlennja-z-paternom-mvc-model-view-controller>.
2. «Symfony Documentation» [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/index.html>.
3. «Symfony Best Practices» [Електронний ресурс] – Режим доступу до ресурсу: https://symfony.com/doc/current/best_practices.html.
4. «The Joy of Symfony» [Електронний ресурс] – Режим доступу до ресурсу: <https://leanpub.com/the-joy-of-symfony>.
5. «Symfony 5: The Fast Track» [Електронний ресурс] – Режим доступу до ресурсу: <https://leanpub.com/symfony5-the-fast-track>.
6. «Symfony 5: The Book» [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/book>.
7. «Symfony 5 Controllers» [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/controller.html>.
8. «Mastering Symfony» [Електронний ресурс] – Режим доступу до ресурсу: <https://www.packtpub.com/product/mastering-symfony-5-third-edition/9781800564312>.
9. «Doctrine ORM для Symfony: Початок роботи» [Електронний ресурс] – Режим доступу до ресурсу: <https://symfony.com/doc/current/doctrine.html>.
10. «Розуміння основ роботи API і REST API – короткий вступ» [Електронний ресурс] – Режим доступу до ресурсу: <https://sebweo.com/rozuminnya-osnov-roboti-api-i-rest-api-korotkij-vstup/>.
11. «Для чого потрібне веб-програмування?» [Електронний ресурс] – Режим доступу до ресурсу: <https://indevlab.com/uk/blog-ua/dlya-chogo-potribne-veb-programuvannya/>.

- 12.«Повне розуміння діаграми компонентів UML за допомогою легкого методу» [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mindonmap.com/uk/blog/uml-component-diagram/>.
- 13.«Основні елементи веб-сайту кінотеатру» [Електронний ресурс] – Режим доступу до ресурсу: <https://www.businessnewsdaily.com/10986-essential-features-movie-theater-website.html>.
- 14.«Використання уніфікованої мови моделювання» [Електронний ресурс] – Режим доступу до ресурсу: http://iwanoff.inf.ua/oop_kn/LabTraining05.html.
- 15.«Діаграма послідовності» [Електронний ресурс] – Режим доступу до ресурсу: <https://www.maxzosim.com/sequence-diagrams/>.
- 16.«PHP для початківців» [Електронний ресурс] – Режим доступу до ресурсу: <http://programming.in.ua/web-design/allphp/30-about-php.html>.
- 17.«17 переваг фреймворку Symfony» [Електронний ресурс] – Режим доступу до ресурсу: <https://icstudio.online/post/17-perevag-frejmworku-laravel>.

ДОДАТКИ

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Код IndexController.php

```

<?php

namespace App\Controller;

use App\Entity\Films;
use DateTime;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class IndexController extends AbstractController
{
    #[Route('/', name: 'app_index')]
    public function index(EntityManagerInterface $entityManager): Response
    {
        /** @var Films[] $films */
        $films = $entityManager->getRepository(Films::class)->findBy([]);
        $currentDateTime = new DateTime();

        return $this->render('index/index.html.twig', [
            'controller_name' => 'IndexController',
            'films' => $films,
            'currentDateTime' => $currentDateTime,
        ]);
    }

    #[Route('/about-us', name: 'about-us')]
    public function about(): Response
    {
        return $this->render('index/about.html.twig', [
            'controller_name' => 'IndexController',
        ]);
    }

    #[Route('/contacts', name: 'contacts')]
    public function contacts(): Response
    {
        return $this->render('index/contacts.html.twig', [
            'controller_name' => 'IndexController',
        ]);
    }
}

```

Код SecurityController.php

```
<?php

namespace App\Controller;

use App\Entity\User;
use App\Form\LoginFormType;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class SecurityController extends AbstractController
{
    #[Route('/login', name: 'app_login')]

    public function login(Request $request, EntityManagerInterface $em,
AuthenticationUtils $authenticationUtils): Response
    {
        $form = $this->createForm(LoginFormType::class);

        $form->handleRequest($request);
        if ($form->isSubmitted() && $form->isValid()) {
            $data = $form->getData();

            // Проверка пользователя в базе данных
            $user = $em->getRepository(User::class)->findOneBy(['username' =>
$data['username']]);
            $hashedPassword = md5($data['password']);
            if (!$user || $user->getPassword() !== $hashedPassword) {
                $this->addFlash('error', 'Invalid credentials');
                return $this->redirectToRoute('app_login');
            }

            if ($user->getAdmin() === 1) {
                // Пользователь является администратором, выполните необходимые
действия (например, редирект на админскую панель)
                return $this->redirectToRoute('admin_dashboard');
            }
            $this->addFlash('success', 'Success');
            // Пользователь не является администратором, выполните необходимые
действия (например, редирект на обычную панель пользователя)
            return $this->redirectToRoute('app_index');
        }
        $error = $authenticationUtils->getLastAuthenticationError();
        $lastUsername = $authenticationUtils->getLastUsername();
        $temporaryMessage = 'Ласкаво просимо! Ви успішно увійшли в систему.';
        return $this->render('security/login.html.twig', [
            'form' => $form->createView(),
            'temporaryMessage' => $temporaryMessage,
            'last_username' => $lastUsername,
            'error' => $error
        ]);
    }

    #[Route(path: '/logout', name: 'app_logout')]
    public function logout(): void
    {
    }
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

```
{  
    throw new \LogicException('This method can be blank - it will be  
intercepted by the logout key on your firewall.');
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Код FilmController.php

```
<?php

namespace App\Controller;

use App\Entity\Films;
use Dompdf\Dompdf;
use DateTime;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Routing\Annotation\Route;
use App\Entity\Payment;
use Symfony\Component\HttpFoundation\Response;

class FilmController extends AbstractController
{
    #[Route("/films", name: "films")]
    public function index(EntityManagerInterface $entityManager)
    {
        $films = $entityManager->getRepository(Films::class)->findBy([]);
        $currentDateTime = new DateTime();

        return $this->render('film/index.html.twig', [
            'controller_name' => 'FilmController',
            'films' => $films,
            'currentDateTime' => $currentDateTime,
        ]);
    }

    public function generateReference()
    {
        $prefix = 'cnmtrx_';
        $uniqid = uniqid($prefix, true);

        return $uniqid;
    }

    #[Route("/films/{id}", name: "film_details")]
    public function showFilmDetails($id, EntityManagerInterface $entityManager)
    {
        // Retrieve film information from the database using Doctrine
        $film = $entityManager->getRepository(Films::class)->find($id);
        $currentDateTime = new DateTime();

        if (!$film) {
            throw $this->createNotFoundException('Фільм не знайдено!');
        }
        $currentDateTimeString = $currentDateTime->format('dmYHi');
        $currentDateTimeStr = $currentDateTime->format('d.m.Y H:i');

        return $this->render('film/details.html.twig', [
            'controller_name' => 'FilmController',
            'film' => $film,
            'filmName' => $film->getName(),
            'currentDateTime' => $currentDateTimeStr,
            'currentDateTimeString' => $currentDateTimeString,
            'generateTicket' => $this->generateReference(),
        ]);
    }
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    #[Route("/generate-ticket/{id}/{reference}", name: "generate_ticket")]
    public function generateTicket($id, $reference, EntityManagerInterface
$entityManager)
    {
        $film = $entityManager->getRepository(Films::class)->find($id);

        if (!$film) {
            throw $this->createNotFoundException('Film not found for the
provided ID!');
        }

        $reference = $this->generateReference(); // Generate the ticket
reference

        $payment = new Payment();
        $payment->setTicket($reference); // Set the ticket reference
        $payment->setAmount("1");
        $payment->setDatetime(new \DateTime());
        $payment->setStatus("1");

        // Persist the Payment entity to the database
        $entityManager->persist($payment);
        $entityManager->flush();

        $html = '<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Ticket</title>
</head>
<body style="text-align: center">
    <h1>Ticket</h1>
    <p>Film: ' . $film->getOrigName() . '</p>
    <p>Location: the choice of location is free</p>
    <p>Ticket number: ' . $reference . '</p>
</body>
</html>';

        $dompdf = new Dompdf();
        $dompdf->set_option('isHtml5ParserEnabled', true);
        $dompdf->set_option('isPhpEnabled', true);
        $dompdf->loadHtml($html, 'UTF-8'); // Set UTF-8 encoding
        $dompdf->setPaper('A4', 'portrait');
        $dompdf->render();

        $pdfContent = $dompdf->output();

        $response = new Response($pdfContent);
        $response->headers->set('Content-Type', 'application/pdf');
        $response->headers->set('Content-Disposition', 'attachment;
filename="ticket.pdf"');

        return $response;
    }
}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Код DashboardController.php

```
<?php

namespace App\Controller\Admin;

use App\Entity\Admin;
use App\Entity\Director;
use App\Entity\Films;
use App\Entity\Genre;
use App\Entity\Payment;
use App\Entity\Session;
use EasyCorp\Bundle\EasyAdminBundle\Config\Assets;
use EasyCorp\Bundle\EasyAdminBundle\Config\Dashboard;
use EasyCorp\Bundle\EasyAdminBundle\Config\MenuItem;
use EasyCorp\Bundle\EasyAdminBundle\Controller\AbstractDashboardController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class DashboardController extends AbstractDashboardController
{
    public function configureAssets(): Assets
    {
        return Assets::new()->addCssFile('css/admin.css');
    }

    public function configureDashboard(): Dashboard
    {
        return Dashboard::new()
            ->setTitle('cinematrix')
            ;
    }

    public function configureMenuItems(): iterable
    {
        yield MenuItem::linkToDashboard('Панель керування', 'fa fa-home');

        yield MenuItem::section('Користувачі');
        yield MenuItem::linkToCrud('Користувачі', 'fa fa-admin', Admin::class);
        yield MenuItem::section('Фільми');
        yield MenuItem::linkToCrud('Фільми', 'fa fa-admin', Films::class);
        yield MenuItem::linkToCrud('Сеанси', 'fa fa-admin', Session::class);
        yield MenuItem::linkToCrud('Жанри', 'fa fa-admin', Genre::class);
        yield MenuItem::linkToCrud('Режисери', 'fa fa-admin', Director::class);
        yield MenuItem::linkToCrud('Оплати', 'fa fa-admin', Payment::class);

    }

    #[Route('/admin', name: 'admin_dashboard')]
    public function index(): Response
    {
        return $this->render('admin/dashboard.html.twig');
    }
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

Код SessionCrudController.php

```
<?php

namespace App\Controller\Admin;

use App\Entity\Hall;
use App\Entity\Session;
use Doctrine\ORM\EntityManagerInterface;
use EasyCorp\Bundle\EasyAdminBundle\Controller\AbstractCrudController;
use EasyCorp\Bundle\EasyAdminBundle\Field\AssociationField;
use EasyCorp\Bundle\EasyAdminBundle\Field\DateTimeField;

class SessionCrudController extends AbstractCrudController
{
    private EntityManagerInterface $entityManager;

    public function __construct(EntityManagerInterface $entityManager)
    {
        $this->entityManager = $entityManager;
    }

    public static function getEntityFqcn(): string
    {
        return Session::class;
    }

    public function configureFields(string $pageName): iterable
    {
        return [
            DateTimeField::new('dateTime'),
            AssociationField::new('film'),
        ];
    }

    public function createEntity(string $entityFqcn)
    {
        $hall = $this->entityManager->getRepository(Hall::class)->find(1);

        $session = new Session();
        $session->setHall($hall);

        return $session;
    }

    /*
    public function configureFields(string $pageName): iterable
    {
        return [
            IdField::new('id'),
            TextField::new('title'),
            TextEditorField::new('description'),
        ];
    }
    */
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

Код admin/dashboard.html.twig

```
{# templates/admin/index.html.twig #}

{% extends '@EasyAdmin/layout.html.twig' %}
{% block stylesheets %}
    {{ parent() }}
    <link href="{{ asset('css/admin.css') }}" rel="stylesheet" />
{% endblock %}
{% block content %}
    <h1 style="text-align: center; margin-top: 50px">Вітаю в адмін панелі сайту
CinemaTrix</h1>
    <br>
    <p style="text-align: center; margin-top: 50px">Для подальшого керування
сайтом вибери пункт з меню зліва.</p>
{% endblock %}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Код film/details.html.twig

```
{# templates/film/details.html.twig #}
{% extends 'base.html.twig' %}

{% block title %}{{ film.name }}{% endblock %}

{% block stylesheets %}
    {{ parent() }}
    <link href="/css/details.css" rel="stylesheet" type="text/css" />
{% endblock %}

{% block content %}
    <div class="head">
        
        <h1>{{ film.name }}</h1>
    </div>
    <p>Жанр: {{ film.genre.name }}</p>
    {% set hours = film.duration // 60 %}
    {% set minutes = film.duration % 60 %}
    <p>Тривалість:
        {% if hours > 0 %}
            {{ hours }} год.
        {% endif %}
        {% if minutes > 0 %}
            {{ minutes }} хв.
        {% endif %}
    </p>
    <p>Опис: {{ film.description }}</p>
    <p>Вікові обмеження: {{ film.rating }}</p>
    <p>Режисери:
        {% for director in film.director %}
            {{ director.name }}
            {% if not loop.last %}, {% endif %}
        {% endfor %}
    </p>
    <p>Рік: {{ film.year }}</p>
    <p>Оригінальна назва: {{ film.getOrigName }}</p>
    <p>Дата прокату: {{ film.getRentalFrom|date("d.m.Y") }} - {{
film.getRentalUntil|date("d.m.Y") }}</p>
    <p>Країна: {{ film.country }}</p>
    <p>Студія: {{ film.studio }}</p>
    <p>Мова: {{ film.language }}</p>

    <script src="https://secure.wayforpay.com/server/pay-widget.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.9-
1/crypto-js.js"></script>
    <script type="text/javascript">
        var paymentSuccess = sessionStorage.getItem('paymentSuccess');
        if (paymentSuccess === 'true') {
            document.getElementById('generateTicketButton').style.display =
'block';
        }
        var wayforpay = new Wayforpay();
        function pay() {
            console.log("{{ filmName }}");
            var processResponse = function (response, params) {
                console.log(response);
                // Handle response here
            }
        }
    </script>
</block>
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

```

window.removeEventListener("message", function() {
    return true;
});

window.addEventListener("message", function (event) {
    if (event.data === "WfpWidgetEventClose") {
        location.reload();
    }
}, false);
function generateOrderID() {
    const timestamp = new Date().getTime();
    const randomNumber = Math.floor(Math.random() * 10000); //
Adjust as needed
    return `${timestamp}-${randomNumber}`;
}
const Reference = generateOrderID();
const product = "ФІЛЬМ: {{ filmName }}, вибір місця вільний.";
const dataExt = {
    merchantAccount: "test_merch_n1",
    merchantDomainName: "cinematrix.local",
    orderReference: Reference,
    orderDate: /*"{{ currentDateTimeString }}"*/ "1415379863",
    amount: "1",
    currency: "UAH",
    productName: product,
    productCount: "1",
    productPrice: "1",
};

const signatureData = [
    dataExt.merchantAccount,
    dataExt.merchantDomainName,
    dataExt.orderReference,
    dataExt.orderDate,
    dataExt.amount,
    dataExt.currency,
    dataExt.productName,
    dataExt.productCount,
    dataExt.productPrice
].join(';');

const merchantSecretKey = "flk3409refn54t54t*FNJRET";
const merchantSignature = CryptoJS.HmacMD5(signatureData,
merchantSecretKey).toString();
console.log(merchantSignature);
const requestData = {
    merchantAccount: dataExt.merchantAccount,
    merchantDomainName: dataExt.merchantDomainName,
    orderReference: dataExt.orderReference,
    orderDate: dataExt.orderDate,
    amount: dataExt.amount,
    currency: dataExt.currency,
    productName: dataExt.productName,
    productCount: dataExt.productCount,
    productPrice: dataExt.productPrice,
    merchantSignature: merchantSignature,
    language: "UA",
};

console.log(requestData);

wayforpay.run(requestData,

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

```

        function (response) {
            // on approved
            processResponse(response, { type: "success" });
            var downloadButton =
document.getElementById('downloadTicketButton');
            downloadButton.style.display = 'block';
            downloadButton.click();
        },
        function (response) {
            // on declined
            processResponse(response, { type: "fail" });
        },
        function (response) {
            // on pending or in processing
            processResponse(response, { type: "pending" });
        }
    );
};
</script>
<div class="butt">
    <button type="button" onclick="pay();"
        style="display:inline-block!important;background:#0488cd;background-
size:cover;width:
256px!important;height:54px!important;border:none!important;border-
radius:14px!important;padding:18px!important;box-shadow:3px 2px 8px
rgba(71,66,66,0.22)!important;text-
align:center!important;outline:none!important;"
        onmouseover="this.style.opacity='0.8';"
onmouseout="this.style.opacity='1';"><span
        style="font-family:Verdana,Arial,sans-serif!important;font-
weight:bold!important;font-size:14px!important;color:#ffffff!important;line-
height:18px!important;vertical-align:middle!important;">Забронювати зараз</span>
    </button>
    <div id="generateTicketButton" class="butt" style="display: none;">
        <a href="{{ path('generate_ticket', { 'id': film.id, 'reference':
generateTicket }) }}" id="downloadTicketButton" class="btn btn-
primary">Завантажити білет</a>
    </div>
</div>

{% endblock %}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

Код film/index.html.twig

```
{% extends 'base.html.twig' %}

{% block title %}Фільмотека{% endblock %}

{% block films %}class="active"{% endblock %}

{% block content %}
    {% for film in films|reverse %}
        {% if film.getRentalUntil|date("d-m-Y") >= currentDateTime|date("d-m-Y") %}
            <div class="movie">
                <h4> <a href="/films/{{ film.id }}">{{ film.name }}</a></h4>
                <span>{{ film.genre.name }}</span>
                <br>
                <br>
                <a href="/films/{{ film.id }}"></a>
                <br>
                <br>
                {% set hours = film.duration // 60 %}
                {% set minutes = film.duration % 60 %}
                <p>Тривалість:
                    {% if hours > 0 %}
                        {{ hours }} год.
                    {% endif %}
                    {% if minutes > 0 %}
                        {{ minutes }} хв.
                    {% endif %}
                </p>
                <p>
                    {{ film.description[:200] }}
                    {% if film.description|length > 200 %}
                        <a href="/films/{{ film.id }}">...</a>
                    {% endif %}
                </p>
                <br>

                {% set sessions_found = false %}
                {% set sessions_time = false %}

                <p>В прокаті до: {{ film.getRentalUntil|date("d.m.Y") }}</p>
                {% set session_times = [] %}
                {% for session in film.sessions %}
                    {% set session_times =
session_times|merge([session.datetime|date("H:i")]) %}
                {% endfor %}
                {% set session_times = session_times|sort %}
                <p>Доступні сеанси:</p>
                {% set unique_times = [] %}
                {% for time in session_times %}
                    {% if time not in unique_times %}
                        {% set unique_times = unique_times|merge([time]) %}
                        <a href="/films/{{ film.id }}"><span class="time-
seanse">{{ time|date("H:i") }}</span></a>
                    {% endif %}
                {% endfor %}
            </div>
        {% endif %}
    {% endfor %}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

```
{% endfor %}  
{% endblock %}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

Код index/index.html.twig

```
{% extends 'base.html.twig' %}

{% block title %}Головна сторінка{% endblock %}
{% block act %}-act{% endblock %}
{% block home %}class="active"{% endblock %}
{% block box %}
    <div class="box">
        <div class="border-right">
            <div class="border-left">
                <div class="inner">
                    <p class="inner-h3-p">Вітаємо на офіційному сайті кінотеатру
<b>Cinema</b><span>Trix</span></p>
                    <p>
                        CinemaTrix - ваш унікальний шлях в світ кіномистецтва
та розваг! Ми пишаємося тим, що надаємо
                        глядачам незабувані враження від перегляду фільмів на
великому екрані. Наші зали обладнані
                        останніми технологічними новинками, щоб забезпечити вам
максимально реалістичне і захоплюючу
                        кіноподорож.
                    </p>
                    <div style="color: red; text-align: center; font-size: 24px;
margin-bottom: 10px">УВАГА!</div>
                    <p style="color: white;">
                        Замовивши квиток на нашому сайті - ви отримаєте
можливість в будь-який зручний для вас момент
                        піти на зацікавивший вас фільм, адже наші квитки не
прив'язані ні до дати, ні до часу сеансів,
                        а також Ви матимете змогу вибрати місце яке вам
сподобається серед доступних на момент сеансу.
                    </p>
                </div>
            </div>
        </div>
    </div>
{% endblock %}

{% block movies %}
    <h3 class="inner-h3-p">Новинки <span>Кіно</span></h3>
    {% set film_found = 0 %}
    {% for film in films|reverse %}
        {% if film.getRentalUntil|date("d-m-Y") >= currentDateTime|date("d-m-Y") %}
    {%
        {% if film_found < 3 %}
            {% set film_found = film_found + 1 %}
            <div class="movie">
                <h4> <a href="/films/{{ film.id }}">{{ film.name }}</a></h4>
                <span>{{ film.genre.name }}</span>
                <br>
                <br>
                <a href="/films/{{ film.id }}"></a>
                <br>
                <br>
                {% set hours = film.duration // 60 %}
                {% set minutes = film.duration % 60 %}
                <p>Тривалість:
                    {% if hours > 0 %}
                        {{ hours }} год.
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

```

        {% endif %}
        {% if minutes > 0 %}
            {{ minutes }} хв.
        {% endif %}
    </p>
    <p>
        {{ film.description[:200] }}
        {% if film.description|length > 200 %}
            <a href="/films/{{ film.id }}">...</a>
        {% endif %}
    </p>
    <br>

    {% set sessions_found = false %}
    {% set sessions_time = false %}

    <p>В прокаті до: {{ film.getRentalUntil|date("d.m.Y") }}</p>
    {% set session_times = [] %}
    {% for session in film.sessions %}
        {% set session_times =
session_times|merge([session.datetime|date("H:i")]) %}
    {% endfor %}
    {% set session_times = session_times|sort %}
    <p>Доступні сеанси:</p>
    {% set unique_times = [] %}
    {% for time in session_times %}
        {% if time not in unique_times %}
            {% set unique_times = unique_times|merge([time]) %}
            <a href="/films/{{ film.id }}"><span class="time-
seanse">{{ time|date("H:i") }}</span></a>
            {% endif %}
        {% endfor %}
    </div>
    {% endif %}
    {% endfor %}
    <h4 style="text-align: center"><a href="/films">Вільше фільмів...</a></h4>
{% endblock %}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

Код security/login.html.twig

```
<title>CinemaTrix | Log In!</title>

{% block stylesheets %}
    <link href="{{ asset('css/login.css') }}" rel="stylesheet" />
{% endblock %}

{% block content %}
<form method="post">
    <div class="login-container">
        <form method="post">
            <h1 class="h3 mb-3 font-weight-normal">Please sign in</h1>
            <div class="form-group">
                <label for="inputUsername">Username</label>
                <input type="text" value="{{ last_username }}" name="username"
id="inputUsername" class="form-control" autocomplete="username" required
autofocus>
            </div>
            <div class="form-group">
                <label for="inputPassword">Password</label>
                <input type="password" name="password" id="inputPassword"
class="form-control" autocomplete="current-password" required>
            </div>
            <input type="hidden" name="_csrf_token"
value="0a70b8ff5ab273727851ae7730c81.nEmoYNPCKNhpgAO9XTSW2y4Z4SZxOksv43UMhlg09h4
.-DHOFLmmp-kccTXuC0D1gld4lGcSQz1CrTpcyDd2j0z2Jf5VoIff9VpxUg">
            <div class="form-group">
                {% if error %}
                    <div class="alert alert-danger">{{
error.messageKey|trans(error.messageData, 'security') }}</div>
                {% endif %}
                <input type="hidden" name="_csrf_token"
                    value="{{ csrf_token('authenticate') }}"
                >

                <button class="btn btn-lg btn-primary" type="submit">
                    Sign in
                </button></div>
            </form>
        </div>
    {% endblock %}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

Код base.html.twig

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>CinemaTrix | {% block title %}{% endblock %}</title>

    {% block meta %}
        <meta name="description" content="CinemaTrix" />
        <meta name="keywords" content="Cinema, Trix, online, cinema" />
        <meta name="author" content="Koida V.M. - student IPZk-20-1" />
    {% endblock %}

    {% block favicon %}
        <link rel="icon" type="image/x-icon" href="/favicon/favicon.ico">
        <link rel="apple-touch-icon" sizes="180x180" href="/favicon/apple-touch-
icon.png">
        <link rel="icon" type="image/png" sizes="32x32" href="/favicon/favicon-
32x32.png">
        <link rel="icon" type="image/png" sizes="16x16" href="/favicon/favicon-
16x16.png">
        <link rel="manifest" href="/favicon/site.webmanifest">
        <link rel="mask-icon" href="/favicon/safari-pinned-tab.svg"
color="#5bbad5">
        <meta name="msapplication-TileColor" content="#da532c">
        <meta name="theme-color" content="#ffffff">
    {% endblock %}

    {% block stylesheets %}
        <link href="/css/style.css" rel="stylesheet" type="text/css" />
        <link href="/css/ie6.css" rel="stylesheet" type="text/css" />
    {% endblock %}

    {% block javascripts %}

    {% endblock %}

</head>
<body id="page1">
    <div class="tail-top">
    <div class="tail-bottom">
        <div id="main">
            {% block header %}
                <div id="header">
                    <div class="row-1">
                        <div class="fleft inner-h3-p"><a
href="/">Cinema<span>Trix</span></a></div>
                        <ul>
                            <li><a href="/"></a></li>
                            <li><a href="/contacts"></a></li>
                        </ul>
                    </div>
                    <div class="row-2">
                        <ul>
                            <li><a href="/" {% block home %}{% endblock
%}>Головна</a></li>
                            <li><a href="/films" {% block films %}{% endblock
%}>Фільми</a></li>
                        </ul>
                    </div>
                </div>
            {% endblock %}
        </div>
    </div>
</div>
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

```

<li><a href="/about-us" {% block about %}{% endblock
%}>Про нас</a></li>
<li><a href="/contacts" {% block contact %}{%
endblock %}>Контакти</a></li>
</ul>
</div>
</div>
{% endblock %}
<div id="content">
{% block content %}

{% block box %}

{% endblock %}

<div class="content">

<div class="movies">
{% block movies %}
{% endblock %}
</div>
</div>
{% endblock %}
</div>
{% block footer %}
<div id="footer">
<div class="left">
<div class="right">
захищено.<br /></div>
</div>
</div>
</div>
{% endblock %}
</div>
</div>
</div>
</body>
</html>

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		