

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

9 червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Веб орієнтована інформаційна система кінотеатру»

здобувача групи ІН – 92 Левицького Вячеслава Олеговича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



(підпис)

Вячеслав Левицький

Керівник,

старший викладач кафедри,

кандидат технічних наук

Артем Коробов

(підпис)

Суми – 2023

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-92 Левицького Вячеслава Олеговича

1. Тема роботи: «Інформаційна технологія прогнозування курсу валют»

затверджую наказом по СумДУ від «01» червня 2023 р. № 0475-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року

3. Вхідні дані до кваліфікаційної роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд технологій, що використовуються для прогнозування курсу валют. 3) Розробка інтелектуальної системи з прогнозування курсу валют. 4) Аналіз результатів.

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

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання

_____ (підпис)

Керівник

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються для прогнозування курсу валют</i>		
3	<i>Розробка інтелектуальної системи з прогнозування курсу валют</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти

_____ (підпис)

Керівник

_____ (підпис)

АНОТАЦІЯ

Записка: 45 стр., 21 рис., 3 додатки, 23 використаних джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки вона зосереджена на розробці інформаційних технологій, веб-додатків для надання послуг бронювання квитків та інших функцій кінотеатру.

Мета роботи — розробка веб-додатку для бронювання квитків кінотеатру, та перегляду інформації про сеанси й фільмів.

Результати — У ході роботи був розроблений веб-додаток для кінотеатру, який автоматизує процеси покупки квитків, вибору місця в залі, перегляду сеансів та інформації про фільми.

**WEB-ДОДАТОК, JAVA, БРОНЮВАННЯ КВИТКІВ, SPRING,
АВТОМАТИЗАЦІЯ**

СУМИ 2023

ЗМІСТ

ВСТУП	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Аналіз аналогічних проєктів	7
1.2 Постановка задачі	8
2. ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ	12
2.1 Середовище розробки IntelliJ IDEA	13
2.2 Мова програмування Java	15
2.3 База даних PostgreSQL	17
2.4 Spring Framework	19
2.5 Клієнтський стек	20
3. РОЗРОБКА WEB-ДОДАТКУ	22
3.1 Програмна реалізація	22
3.2 Використання веб-додатку	29
3.3 Архітектура веб-додатку	36
ВИСНОВКИ	38
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	39
Додаток А	41
Додаток Б	43
Додаток В	45

ВСТУП

Актуальність. Актуальність розробки веб-додатку для кінотеатру є дуже актуальною в сучасному світі розвитку цифрових технологій і онлайн-сервісів, коли всі оффлайн служби прагнуть перейти в онлайн та розробити додатки, щоб уникнути скупчення людей на фоні війни та карантину.

Предмет дослідження. Методологія розробки зручних веб-додатків для поліпшення оренди автомобілів.

Гіпотеза. Автоматизацію покупки квитків на сеанси можна досягнути шляхом використання інформаційної технології, що реалізує веб-додаток.

Структура. Ця робота включає такі розділи: вступ, аналітичний огляд, вибір методу для розв'язання поставленої задачі, опис програмного забезпечення інформаційної системи, висновки, список використаних джерел та додатки.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз аналогічних проєктів

Сьогодні існує декілька аналогів такого типу додатків в інтернет-просторі. Основні з них є додатки кінотеатрів «Мультиплекс» та «Планета кіно».

Перший аналог, який розглянуто – це planetakino.ua. Основна сторінка, що цікавить цільового користувача – розклад. Сторінка розкладу на сайті planetakino.ua містить інформацію про розклад сеансів фільмів у всіх кінотеатрах мережі "Планета Кіно". На цій сторінці користувач може обрати дату та час сеансу фільму.

Кожен фільм в списку має свою назву, опис, тривалість та жанр. Крім того, для кожного сеансу зазначається мова дубляжу та 3D-формат. Для зручності в пошуку під розкладом можна вибрати фільтри за жанрами фільмів, мовою дубляжу, наявністю субтитрів та 3D-форматом.

Кожен фільм на сторінці розкладу має свою окрему сторінку, на якій можна знайти додаткову інформацію про нього, таку як трейлер, опис сюжету і рейтинг IMDB.

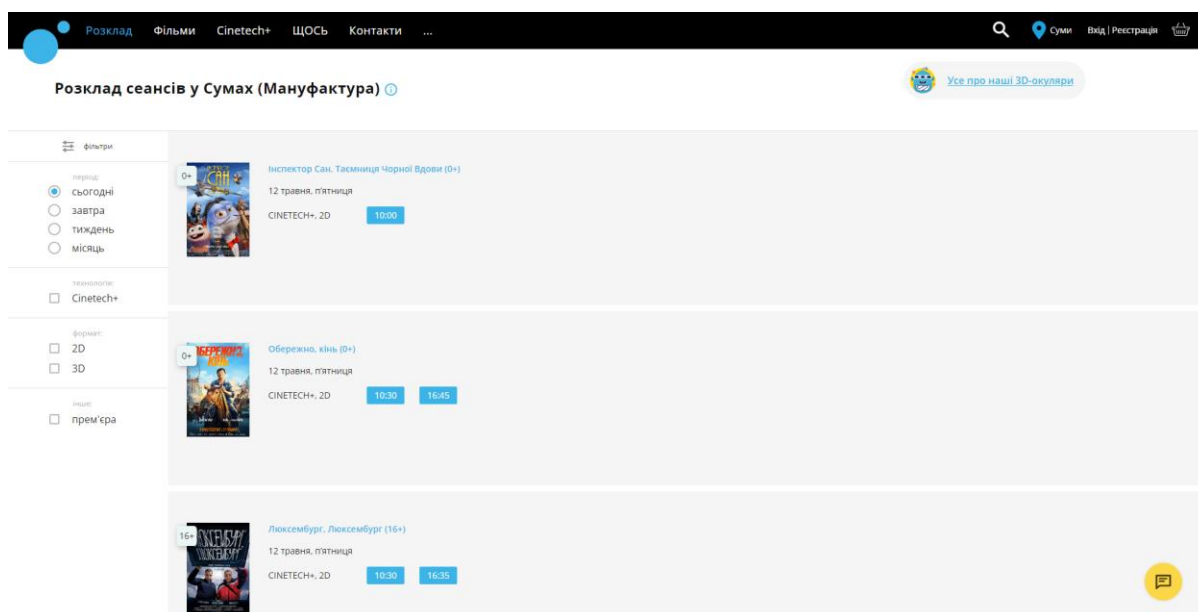


Рисунок 1.1 – Вигляд сторінки розкладу web-додатку planetakino.ua

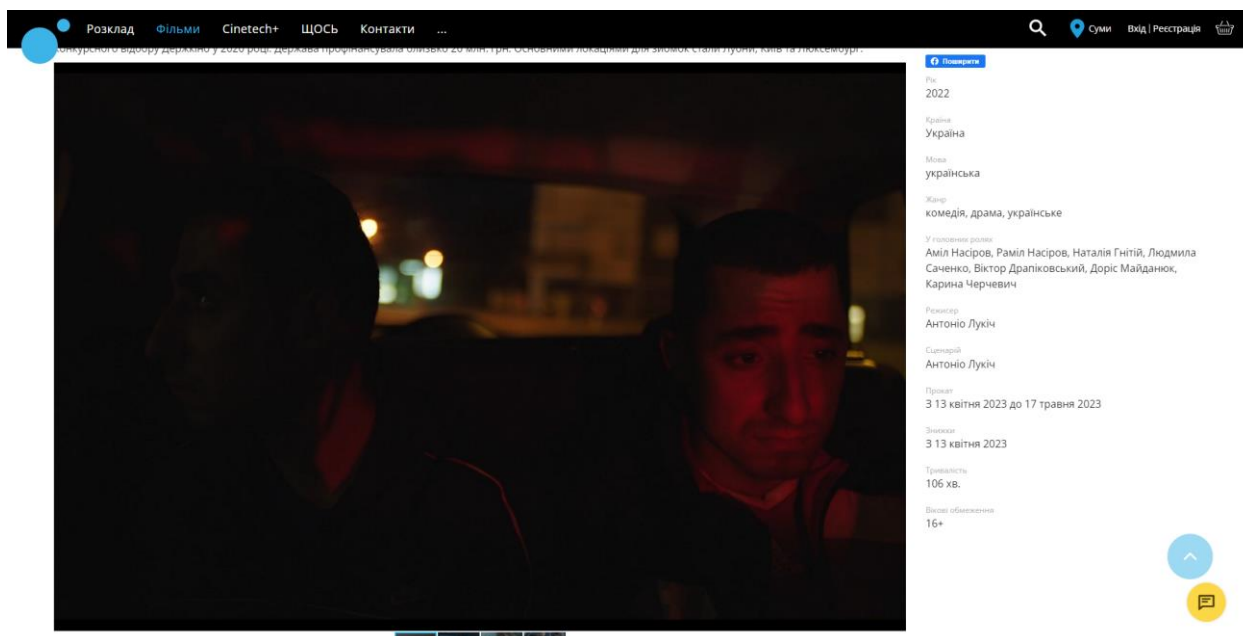


Рисунок 1.2 – Вигляд сторінки фільму web-додатку planetakino.ua

Наступний аналог – multiplex.ua. Дизайн цікавий, тобто, додаток побудований не по принципу «скроль вниз», а «скроль горизонтально». Щоб побачити фільми, що в прокаті, потрібно натискати кнопку далі, таким образом утворюється стрічка фільмів, й можна сказати, що додаток побудований горизонтально. При наведенні на фільм відображають доступні сеанси, їх час та ціна квитку. Крім того, на головній сторінці можуть бути розміщені посилання на програму лояльності, де ви можете дізнатися про переваги та бонуси для постійних клієнтів Multiplex.

1.2 Постановка задачі

У сучасному кінотеатрі глядачеві надається великий вибір цікавих фільмів та мультфільмів з високоякісним зображенням і звуком, а також комфортні сидіння. Проте, перший крок для перегляду фільму - це придбання квитка, що часто вимагає витрат часу на чергування в касі.

Зручніше скористатися мобільним додатком, де можна отримати повну інформацію про фільм та переглянути вільні місця у залі з можливістю забронювати місце або придбати квиток.

Метою дипломної роботи є розробка програмного продукту з використанням сучасних комп'ютерних технологій, який дозволить переглядати інформацію про фільми та вільні місця у кінотеатрі та забронювати місця або придбати квиток без необхідності стояти в черзі до каси.

Додаток, що розроблюється повинен мати можливості:

- створити особистий кабінет
- надавати користувачу інформацію про доступні фільми
- показувати опис фільму, постер, рейтинг
- показувати доступні місця та вибір одного, чи декількох місць
- надавати можливість купити квиток
- прийняти інформацію від користувача (контактні дані, картку)
- надати інформацію про успішну покупку
- показати доступні сеанси
- фільтрувати та сортувати сеанси

Потенційними користувачами створюваного продукту є звичайні користувачі, які мають персональний комп'ютер чи смартфон, з інстальованим браузером.

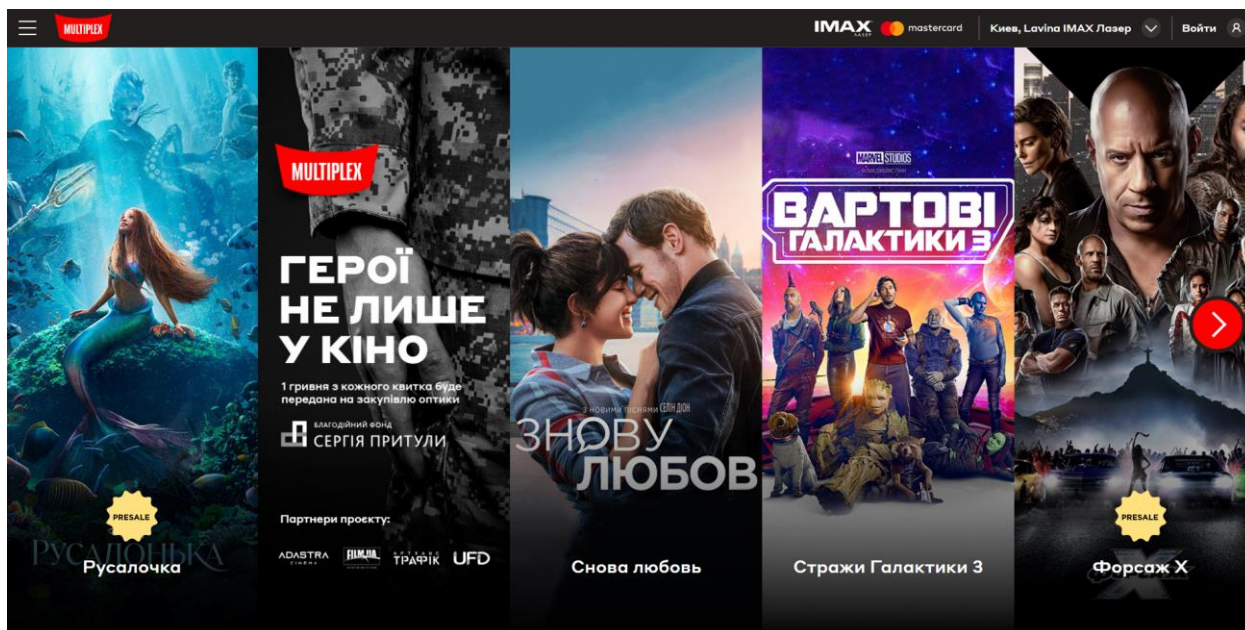


Рисунок 1.3 – Вигляд домашньої сторінки multiplex.ua

На сторінці фільму основна інформація: вікові обмеження, рік випуску, режисер, період прокату, рейтинг, довжина стрічки, студія та короткий опис. У правій частині екрану відображаються сеанси по цьому фільму. Загалом, сторінка фільму добре структурована та інформативна.



Рисунок 1.4 – Вигляд сторінки фільму web-додатку multiplex.ua

Третій та останній аналог - kino.sumu.ua. По посиланню знаходиться web-додаток кінотеатру «Дружба». На головній сторінці додатку ви знайдете зручний та інформативний дизайн, спрямований на надання вам швидкого доступу до інформації про кінопрокат у місті Суми. На відміну від попередніх аналогів, додаток вміщає одразу коротку інформацію про фільм, але щоб купити квитки потрібно перейти на іншу вкладку.

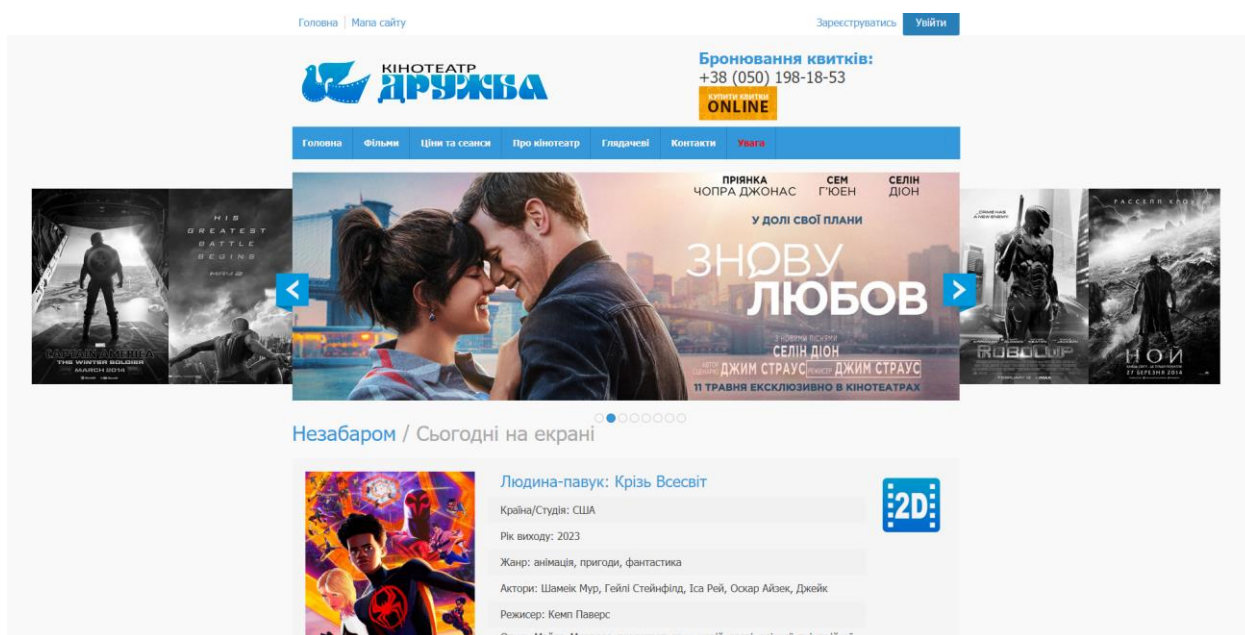


Рисунок 1.5 – Вигляд головної сторінки web-додатку kino.sumu.ua

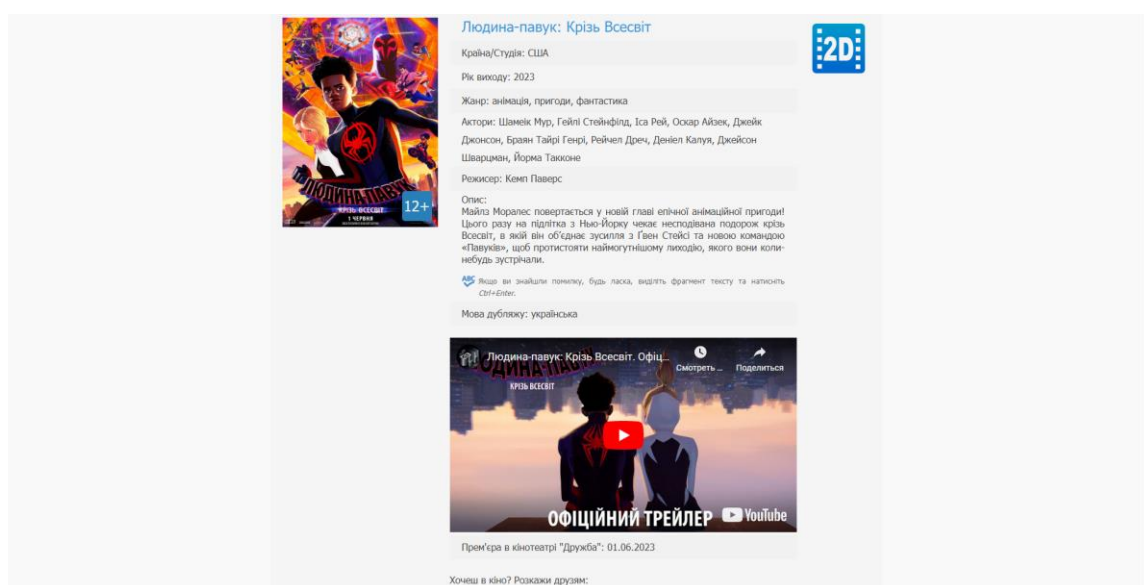


Рисунок 1.6 – Вигляд сторінки фільму web-додатку kino.sumu.ua

2. ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

Для ефективної розробки програмного продукту програміст повинен обрати такі технології та інструменти, що спростять його роботу та нададуть необхідні інструменти для успішного виконання завдань. Вибір правильних технологій та інструментів має безпосередній вплив на швидкість розробки, якість та ефективність роботи продукту.

Для розробки системи кінотеатру було обрано такі засоби:

— середовище розробки програм — JetBrains IntelliJ IDEA

— мова програмування — Java

— база даних — PostgreSQL

— серверна частина — Spring Framework

— клієнтська частина — HTML + Bootstrap + Thymeleaf

Використання такого стеку технологій для написання веб-додатку має кілька переваг порівняно з іншими альтернативами:

- **IntelliJ IDEA:** JetBrains IntelliJ IDEA є потужним інтегрованим середовищем розробки (IDE) для мови програмування Java. Воно надає багато корисних функцій, таких як автодоповнення коду, система контролю версій та інструменти для аналізу коду. IntelliJ IDEA допомагає розробникам писати якісний і ефективний код, що збільшує продуктивність розробки.
- **Мова програмування Java:** Java є однією з найпопулярніших мов програмування для веб-розробки. Вона має широку спільноту розробників, велику кількість ресурсів та багато фреймворків і бібліотек, що спрощують розробку. Java також відома своєю надійністю, безпекою та високою продуктивністю.

- **PostgreSQL:** PostgreSQL є потужною та надійною відкритою реляційною базою даних. Вона має багатий набір функцій, підтримку транзакцій та високу масштабованість. Використання PostgreSQL дозволяє зберігати дані надійно та ефективно керувати ними.
- **Spring Framework:** Spring Framework є одним з найпопулярніших фреймворків для розробки серверної частини веб-додатків на Java. Він надає широкий набір інструментів і бібліотек для розробки веб-додатків, таких як інверсія керування, керування транзакціями, безпека та багато іншого. Spring Framework спрощує розробку та підтримку веб-додатків, забезпечуючи високу продуктивність та масштабованість.
- **HTML + Bootstrap + Thymeleaf:** Використання комбінації HTML, Bootstrap та Thymeleaf для клієнтської частини дозволяє створювати привабливий та динамічний інтерфейс користувача. HTML є стандартною мовою розмітки веб-сторінок, а Bootstrap надає набір готових компонентів та стилів, що спрощує розробку респонсивного дизайну. Thymeleaf є шаблонним двигуном, який інтегрується з Java і дозволяє створювати динамічний контент на стороні сервера.

Загалом, використання цього стеку технологій забезпечує розробникам потужні та надійні інструменти для створення веб-додатків, які можуть бути ефективними, масштабованими та зручними для підтримки.

2.1 Середовище розробки IntelliJ IDEA

IntelliJ IDEA - це інтегроване середовище розробки для різних мов програмування, таких як Java, Kotlin, Groovy, Scala, Python, і багатьох інших. Воно розроблено компанією JetBrains і має велику кількість корисних функцій, що допомагають розробникам бути більш продуктивними.

Серед основних функцій IntelliJ IDEA можна виділити: автодоповнення коду, вбудовану систему контролю версій, візуальний редактор інтерфейсу користувача, розширену систему пошуку та заміни тексту в проекті,

вбудовані інструменти для тестування коду, підтримку популярних фреймворків, таких як Spring і Hibernate, та багато іншого.

IntelliJ IDEA також має широку спільноту користувачів та розробників, які створюють різні плагіни та розширення для покращення функціональності середовища розробки. Додатково, IntelliJ IDEA є платним продуктом, але також має безкоштовну Community Edition з обмеженими функціями.

Узагалі, IntelliJ IDEA є потужним інструментом для розробки програмного забезпечення, що може бути корисним для різних типів проектів та розробників.

Переваги IntelliJ IDEA:

- Функціональність
- Великий вибір інструментів для роботи з кодом
- Ергономічність
- Інструменти для спільної та віддаленої роботи

Недоліки IntelliJ IDEA:

- Вибагливість до системних ресурсів
- Урізана функціональність безкоштовної версії
- Відсутність перекладу інтерфейсу

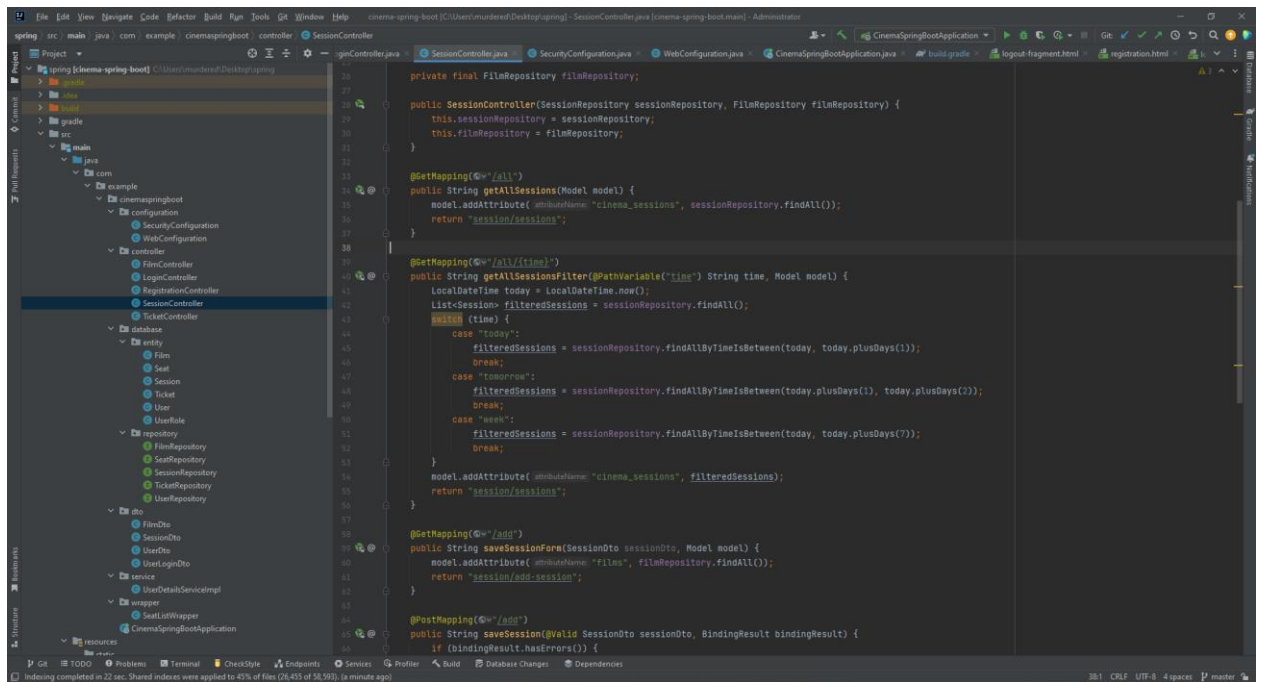


Рисунок 2.1 – Вигляд середовища розробки IntelliJ IDEA

2.2 Мова програмування Java

Java - це високорівнева мова програмування, що дозволяє розробляти серверні додатки та веб-сайти. Її особливість полягає у великій кількості бібліотек, які дозволяють легко і швидко розробляти різноманітні застосунки. Також, Java є об'єктно-орієнтованою мовою, що дає змогу програмістам писати код, який легко зрозуміти та модифікувати.

Java має багато інструментів для розробки серверних додатків, таких як Apache Tomcat та JBoss, які дозволяють створювати високопродуктивні веб-сервери та сервлети.

Іншою важливою особливістю мови Java є її платформонезалежність. Код, написаний на Java, може працювати на будь-якій операційній системі, що має встановлену відповідну віртуальну машину Java (JVM). Це зменшує ризики, пов'язані з підтримкою різних операційних систем, та дозволяє писати більш універсальний код.

Java також має широку підтримку відкритих стандартів, таких як XML, SOAP, JSON та REST, що дозволяє легко обмінюватись даними між різними системами та платформами.

Загалом, Java є дуже популярною мовою для серверного програмування, завдяки своїм особливостям, які дозволяють швидко та ефективно розробляти високоякісні серверні додатки та веб-сайти.

Переваги Java:

- **Кросплатформенність:** Java є платформонезалежною мовою програмування, що означає, що веб-додатки, написані на Java, можуть працювати на різних операційних системах, таких як Windows, macOS та Linux.
- **Велика екосистема:** Java має велику спільноту розробників та багато сторонніх бібліотек і фреймворків, що дозволяє значно прискорити процес розробки. Наприклад, фреймворки, такі як Spring і JavaServer Faces (JSF), надають потужні інструменти для створення веб-додатків на Java.
- **Висока продуктивність:** Java є компільованою мовою, що дозволяє виконувати код швидше, ніж інтерпретовані мови програмування. Вона також має ефективне керування пам'яттю, що допомагає уникнути проблем зі зборкою сміття.
- **Безпека:** Java має вбудовану систему безпеки, яка дозволяє обмежити доступ до ресурсів та запобігти вразливостям, таким як витік пам'яті чи виконання зловмисного коду.

Недоліки Java:

- **Потребує більше ресурсів:** Java-додатки можуть бути більшими за обсягом та вимагати більше ресурсів в порівнянні з іншими мовами програмування. Це може мати вплив на продуктивність

веб-додатків, особливо при великому обсязі даних або високій навантаженості.

- Відсутність підтримки на незалежних пристроях: Веб-додатки, написані на Java, можуть не працювати на деяких мобільних пристроях, які не мають підтримки для виконання Java-коду. Для мобільних додатків можуть використовуватись альтернативні технології, такі як Kotlin або React Native.
- Час розгортання: Написання веб-додатку на Java може вимагати більше часу для розгортання порівняно з іншими мовами, оскільки потрібно налаштувати середовище виконання Java на сервері.
- Написання більшого обсягу коду: В порівнянні з деякими іншими мовами програмування, такими як Python або Ruby, Java вимагає більшого обсягу коду для реалізації тієї ж функціональності. Це може призвести до більшої складності коду та збільшення часу розробки.

Отже, зважаючи на переваги Java, та наявність Spring Framework для розробки Web-додатків, було обрано саме її для розробки web-додатку, що, в перспективі, зможе розширюватись.

2.3 База даних PostgreSQL

PostgreSQL - це об'єктно-реляційна система управління базами даних, яка забезпечує ефективне зберігання та операції зі структурованими даними. Вона є однією з найпопулярніших систем управління базами даних у світі.

PostgreSQL є відкритим програмним забезпеченням, що дає можливість користувачам вільно розповсюджувати, змінювати та використовувати її. Це забезпечує багато переваг, таких як доступність, гнучкість та розширюваність.

Однією з особливостей PostgreSQL є її можливість обробки складних запитів та забезпечення безпеки даних. Вона підтримує різні типи даних, включаючи числа, текст, дати, геодані та бінарні файли.

PostgreSQL має розширену систему управління правами доступу, що дозволяє обмежувати доступ до даних для різних користувачів та груп користувачів. Вона також підтримує різні методи шифрування даних, що дозволяє забезпечувати захист від несанкціонованого доступу до даних.

PostgreSQL підтримує мови програмування, такі як SQL, Python, Java та C++, що дозволяє розширювати функціональність системи та підтримувати більш складні операції з даними.

Переваги PostgreSQL над іншими:

- Вільний доступ
- Можна встановити на будь-яку платформу
- Підтримує різні формати даних
- Дозволяє працювати з великими розмірами даних
- Підтримує всі функції, які є у сучасних базах

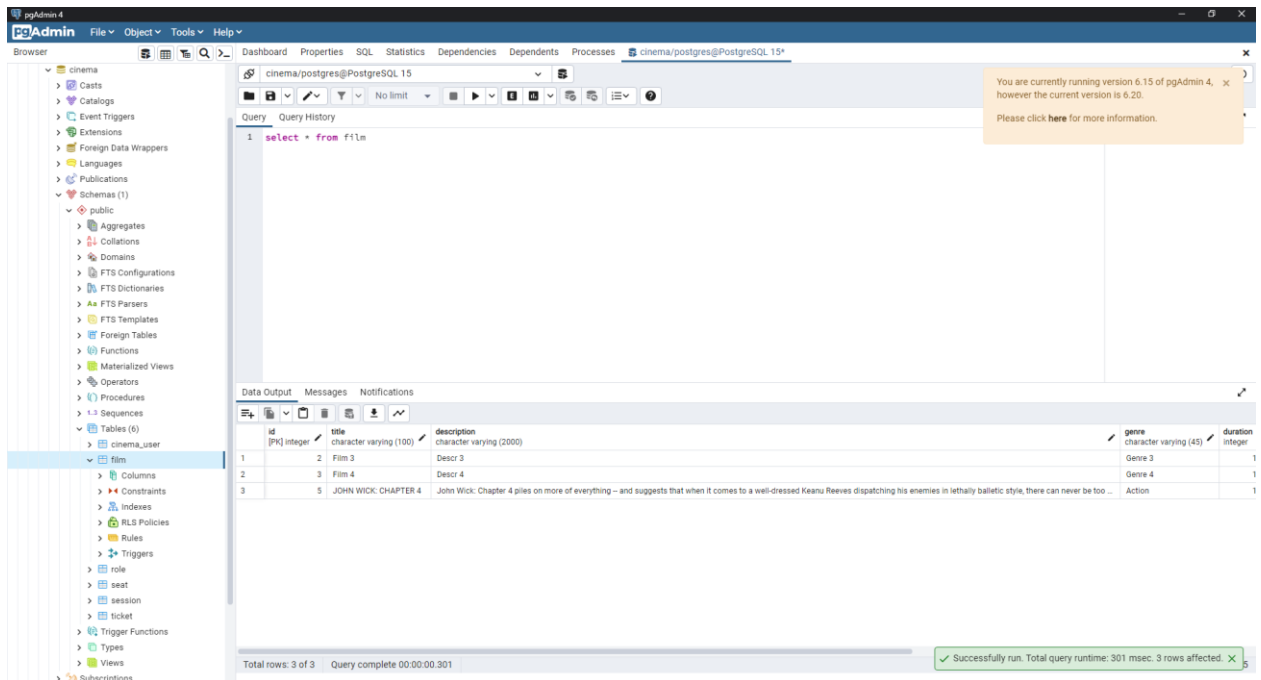


Рисунок 2.2 – Вигляд платформи адміністрування pgAdmin

2.4 Spring Framework

Spring Framework є одним з найпопулярніших та найважливіших фреймворків для розробки програмного забезпечення на мові Java. Він надає розробникам широкий набір інструментів та бібліотек для створення високоякісних, масштабованих та безпечних додатків.

Spring Framework включає в себе модулі, які надають підтримку різних аспектів програмної розробки, таких як інверсія керування (Inversion of Control, IoC), внедрення залежностей (Dependency Injection, DI), обробка подій, взаємодія з базами даних та інші. Spring також має підтримку архітектурного шаблону Model-View-Controller (MVC), що дозволяє розробляти веб-додатки та API.

Крім того, Spring Framework дозволяє розробникам створювати додатки відповідно до принципів SOLID, що забезпечує гнучкість та розширюваність коду. Фреймворк також підтримує безпеку та

аутентифікацію, що робить його придатним для розробки додатків з високими вимогами до безпеки.

Узагальнюючи, Spring Framework є потужним та гнучким фреймворком для розробки програмного забезпечення на мові Java, який надає розробникам багато інструментів та можливостей для створення високоякісних та масштабованих додатків.

2.5 Клієнтський стек

HTML, Bootstrap та Thymeleaf - це технології веб-розробки, які часто використовуються разом для створення сучасних та ефективних веб-додатків.

HTML - це основна мова розмітки веб-сторінок, яка використовується для створення структури та контенту веб-сторінок. Вона дозволяє розміщувати різноманітні елементи, такі як текст, зображення, відео та інші елементи на веб-сторінці.

Bootstrap - це набір готових CSS та JavaScript компонентів, які дозволяють швидко та ефективно створювати веб-сторінки зі зручним та сучасним дизайном. Bootstrap дозволяє створювати різні компоненти, такі як меню навігації, форми, таблиці та інші елементи, які можна легко налаштувати та адаптувати до різних розмірів екранів.

Thymeleaf - це шаблонний движок, який дозволяє легко та ефективно створювати веб-сторінки, що можуть бути динамічно змінені з боку сервера. Thymeleaf дозволяє вбудовувати веб-елементи прямо в HTML сторінки, що забезпечує легке управління веб-елементами з боку сервера.

Застосування цих технологій разом дозволяє створювати веб-додатки, які мають сучасний та привабливий дизайн, а також можуть бути динамічно змінені з боку сервера. Оскільки ці технології є популярними та широко використовуються веб-розробниками, вони також підтримуються різними

інструментами та фреймворками, такими як Spring, що дозволяє швидко та легко створювати веб-додатки на основі цих технологій.

3. РОЗРОБКА WEB-ДОДАТКУ

3.1 Програмна реалізація

Java є однією з найпопулярніших мов програмування для розробки серверних додатків. Вона відома своєю надійністю, масштабованістю та широким спектром інструментів та бібліотек для розробки веб-додатків. У поєднанні з фреймворком Spring Boot, Java стає ще потужнішою для розробки серверних додатків.

Spring Boot є фреймворком для розробки додатків на основі платформи Spring. Він надає високорівневі абстракції та готові рішення для швидкої та ефективної розробки серверних додатків. Spring Boot спрощує конфігурацію та розгортання додатків, дозволяючи розробникам швидко створювати потужні серверні застосунки. Компонентами додатку є:

- Шар конфігурації
- Шар контролерів
- Шар репозиторіїв
- Шар сутностей
- Шар ремапінгу

Шар конфігурації

У Spring Boot шар конфігурації є важливою складовою для налаштування та керування додатком. Він дозволяє визначити, які компоненти мають бути створені, як вони пов'язані між собою та які налаштування застосовуються.

Шар конфігурації в Spring Boot базується на принципах інверсії управління (Inversion of Control, IoC) та внедрення залежностей (Dependency Injection, DI). Замість того, щоб розраховувати на створення та конфігурацію

об'єктів вручну, Spring Boot дозволяє використовувати анотації та конфігураційні файли для автоматичного створення та зв'язку компонентів.

Основні аспекти шару конфігурації в Spring Boot:

Анотації: Spring Boot надає ряд анотацій, таких як `@Configuration`, `@Component`, `@Service`, `@Repository` і т.д., які можна застосовувати до класів для позначення їх як компонентів, що повинні бути створені та керовані фреймворком.

Автоматичне конфігурування: Spring Boot має механізми автоматичного конфігурування, які дозволяють налаштовувати компоненти на основі наявних налаштувань або умов. Наприклад, можна використовувати анотацію `@Value` для внедрення значень зовнішнього файлу конфігурації в компонент.

Конфігураційні файли: Spring Boot підтримує різні типи конфігураційних файлів, таких як `application.properties` або `application.yml`. У цих файлах можна визначати властивості та налаштування для додатку.

Внедрення залежностей: Шар конфігурації дозволяє використовувати ін'єкцію залежностей для автоматичного внедрення об'єктів в компоненти. Це забезпечує локалізацію створення та управління залежностями та сприяє відокремленню компонентів.

Загалом, шар конфігурації в Spring Boot дозволяє зручно налаштовувати та керувати компонентами додатку, спрощує внедрення залежностей та допомагає забезпечити правильну конфігурацію та функціональність додатку.

Шар контролерів

Шар контролерів в Spring Boot є одним з ключових компонентів для обробки вхідних HTTP-запитів та керування потоком даних у веб-додатку.

Він відповідає за прийняття запитів від клієнтів, виконання відповідних дій та повернення результату назад.

У Spring Boot контролери реалізовані за допомогою анотацій, що спрощує створення та конфігурацію веб-маршрутів. Основні анотації, які використовуються в контролерах:

- `@Controller`: Ця анотація позначає клас як контролер та вказує, що результати методів контролера повинні бути прямо повернуті як відповідь HTTP, замість перенаправлення на вид.
- `@RequestMapping`: Ця анотація вказує на адресу URL, за якою будуть доступні методи контролера. Вона може бути застосована на класі контролера для визначення загального шляху та на методах для конкретних шляхів.
- `@GetMapping`, `@PostMapping`, `@PutMapping`, `@DeleteMapping`: Ці анотації вказують на тип HTTP-запиту, який буде оброблятися відповідним методом контролера.
- `@PathVariable`: Ця анотація використовується для зчитування значень змінних з URL-шляху.
- `@RequestBody`: Ця анотація дозволяє отримати тіло HTTP-запиту як об'єкт або типізовану структуру даних.

Контролери можуть мати методи, які обробляють різні типи запитів (GET, POST, PUT, DELETE) та здійснюють взаємодію з різними шари додатку, такими як сервіси або репозиторії, для обробки логіки бізнес-процесів та отримання необхідних даних.

Контролери в Spring Boot сприяють розробці RESTful API, веб-сервісів та веб-додатків, надаючи зручний спосіб організації та управління веб-маршрутами та обробкою запитів клієнтів. Приклади REST-запросів:

```
http://localhost:8080/session/all
```


`http://localhost:8080/ticket/seats/7`

`http://localhost:8080/ticket/buy/5`

`http://localhost:8080/ticket/user-tickets`

Шар репозиторіїв

Шар репозиторіїв в Spring Boot є компонентом, що відповідає за взаємодію з базою даних або іншим джерелом даних. Він надає абстракцію для збереження, читання, оновлення та видалення даних у додатку.

У Spring Boot репозиторії реалізовані з використанням підходу Object-Relational Mapping (ORM) або Object-Document Mapping (ODM) за допомогою фреймворку Spring Data. Spring Data спрощує взаємодію з різними джерелами даних, такими як реляційні бази даних, NoSQL-сховища, Elasticsearch і багато інших.

Основні аспекти шару репозиторіїв в Spring Boot:

- **Інтерфейси репозиторіїв:** Репозиторії визначаються як інтерфейси, що описують методи для виконання операцій збереження, читання, оновлення та видалення даних. Інтерфейси репозиторіїв можуть використовувати стандартні методи Spring Data або визначати власні кастомні методи для виконання специфічних операцій.
- **Автоматична реалізація репозиторіїв:** Spring Data автоматично генерує реалізацію репозиторіїв на підставі інтерфейсів репозиторіїв. Для цього використовуються конкретні реалізації, такі як Hibernate для реляційних баз даних або Spring Data MongoDB для NoSQL-сховищ. Це дозволяє розробникам працювати з репозиторіями без необхідності написання багатої низькорівневої логіки.
- **Кастомні методи репозиторіїв:** Розробники можуть визначати власні кастомні методи в інтерфейсі репозиторію, які відповідають специфічним потребам додатку. Spring Data автоматично створює реалізацію цих методів на основі їх імен та сигнатур.

- Підтримка критеріїв і специфікацій: Spring Data надає можливість використовувати критерії та специфікації для складання складних запитів до бази даних. Це дозволяє створювати гнучкі та динамічні запити без написання SQL-запитів вручну.
- Підтримка транзакцій: Репозиторії в Spring Boot можуть бути огорнуті транзакціями, що забезпечує консистентність та цілісність даних при виконанні операцій збереження, оновлення та видалення.

Шар репозиторіїв в Spring Boot допомагає розробникам швидко та зручно взаємодіяти з базою даних або іншими джерелами даних, забезпечує абстракцію від деталей доступу до даних та спрощує розробку стійких та ефективних додатків.

Шар сутностей

Шар сутностей (Entity) в Spring Boot використовується для представлення об'єктів доменної моделі додатку. Сутності відображають структуру та поведінку даних, які зберігаються або передаються в додатку.

У Spring Boot сутності зазвичай представлені як класи, що мають поля, методи та анотації для визначення їх характеристик. Основні аспекти шару сутностей:

- Анотації JPA: Для роботи з реляційними базами даних в Spring Boot використовується Java Persistence API (JPA). Сутності можуть бути позначені анотаціями JPA, такими як `@Entity`, `@Table`, `@Column` та інші, що дозволяють визначати їх структуру та мапування на таблиці бази даних.
- Атрибути сутностей: Сутності мають поля, які відповідають атрибутам даних. Ці поля можуть мати анотації, такі як `@Id`, `@GeneratedValue`, `@Column` та інші, для визначення первинного ключа, автоматичної генерації значень або додаткових характеристик поля.

- **Взаємозв'язки між сутностями:** Сутності можуть мати взаємозв'язки між собою, такі як один-до-одного, один-до-багатьох або багато-до-багатьох. Для визначення цих взаємозв'язків використовуються анотації JPA, такі як `@OneToOne`, `@OneToMany`, `@ManyToOne`, `@ManyToMany`.
- **Методи доступу до даних:** Сутності можуть мати методи доступу (гетери та сетери) для отримання та встановлення значень полів. Це допомагає забезпечити контроль над доступом до даних та виконати додаткову логіку при зміні або отриманні значень.
- **Валідація даних:** Сутності можуть мати анотації для валідації даних, такі як `@NotNull`, `@Size`, `@Pattern` та інші. Це дозволяє перевіряти правильність введених даних та забезпечувати їх цілісність.

Шар сутностей в Spring Boot дозволяє розробникам визначати структуру даних та їх поведінку у додатку, використовуючи анотації та методи доступу. Він спрощує роботу з базою даних та дозволяє виконувати операції збереження, оновлення, видалення та запиту даних.

Блокування функціоналу від час тривоги

Для отримання інформації про повітряні тривоги було використано API, реалізація отримання списку областей де зараз тривога:

```
private List<String> returnAlarm(){
    List<String> namesList = new ArrayList<>();
    try {
        URL url = new
URL("https://alerts.com.ua/api/states");

        HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
        connection.setRequestMethod("GET");
```

```

        int responseCode = connection.getResponseCode();
        if (responseCode == HttpURLConnection.HTTP_OK) {
            BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
            String line;
            StringBuilder response = new StringBuilder();
            while ((line = reader.readLine()) != null) {
                response.append(line);
            }
            reader.close();

            JSONObject jsonResponse = new
JSONObject(response.toString());

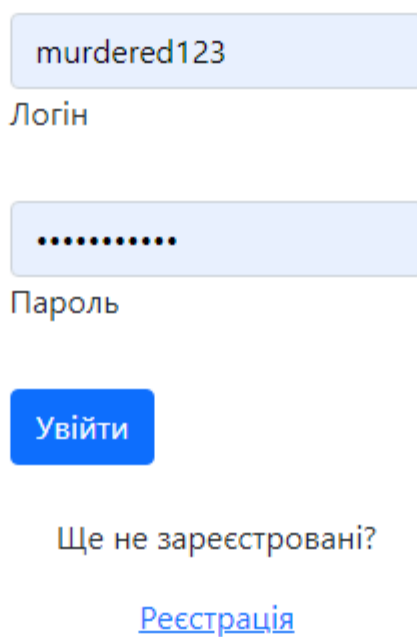
            JSONArray statesArray =
jsonResponse.getJSONArray("states");

            for (int i = 0; i < statesArray.length(); i++) {
                JSONObject state =
statesArray.getJSONObject(i);
                String name = state.getString("name");
                namesList.add(name);
            }
        } else {
            System.out.println("Error: " + responseCode);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return namesList;
}

```

3.2 Використання веб-додатку

При відкритті додатку, перше, що можна побачити на екрані – сторінка авторизації (рис 3.2.1).



The image shows a login form with the following elements:

- A text input field containing the username "murdered123".
- The label "Логін" (Login) positioned below the first field.
- A password input field with masked characters ".....".
- The label "Пароль" (Password) positioned below the second field.
- A blue button labeled "Увійти" (Login).
- The text "Ще не зареєстровані?" (Not yet registered?) below the button.
- A blue underlined link labeled "Реєстрація" (Registration) below the text.

Рис 3.2.1 – Форма авторизації

Якщо користувач не має аккаунту, то він повинен зареєструватись, вказавши незайнятий логін та електронну пошту (рис 3.2.2).

murdered123

Логін

.....

Пароль

Електронна пошта

Зареєструватись

Рис 3.2.2 – Форма реєстрації

Після успішної авторизації користувача буде переадресовано на сторінку розкладу (рис 3.2.3).

Розклад

Фільми [Ваші квитки](#) Мова Вихід

Filter

Today

Tomorrow

Week

Розклад

Назва	Ціна	Купити квитки
Film 4	2023-06-01 15:55	345 Купити
Film 4	2023-06-01 15:55	56 Купити
Film 3	2023-06-01 15:55	120 Купити
Film 4	2023-06-01 15:55	45 Купити

© 2023 Cinema.

Рис 3.2.3 – Сторінка розкладу кінотеатру

Якщо увійти в систему за допомогою аккаунту адміністратора, то буде представлено інший функціонал, а саме видалення й редагування сеансів (рис 3.2.4).

Розклад

Назва	Ціна	Дії
Film 4	2023-06-01 15:58	345 Оновити Видалити
Film 4	2023-06-01 15:58	56 Оновити Видалити
Film 3	2023-06-01 15:58	120 Оновити Видалити
Film 4	2023-06-01 15:58	45 Оновити Видалити

Рис 3.2.4 – Вигляд розкладу з правами адміністратора

Редагуючи сеанс (рис 3.2.5) чи фільм (рис 3.2.6) актуальна інформація заповнюється в форму автоматично:

16.02.2023 20:07
📅

Час

345

Ціна

654

Available places

Фільм Film 4 ▼

Редагувати сеанс

Рис 3.2.5 – Сторінка редагування сеансу

Film 3
Назва
Descr 3
Опис
Genre 3
Жанр
120
Тривалість
4.0
IMDB
https://static.vecteezy.com
Пряме посилання на постер
Редагувати фільм

Рис 3.2.6 – Сторінка редагування фільму

Авторизувавшись як звичайний користувач, буде доступна покупка квитків. Натиснувши на «Купити», користувача буде переадресовано на сторінку вибору місця в залі (рис 3.2.7).

Вибір місця було реалізовано за допомогою елементу «checkbox». Уже обрані місця є неклікабельними.



Рис 3.2.7 – Вибір місця в залі

Куплені квитки можна подивитись в кабінеті користувача (рис 3.2.8).

Надано інформацію про фільм, час сеансу та обрані місця (рис 3.2.9).

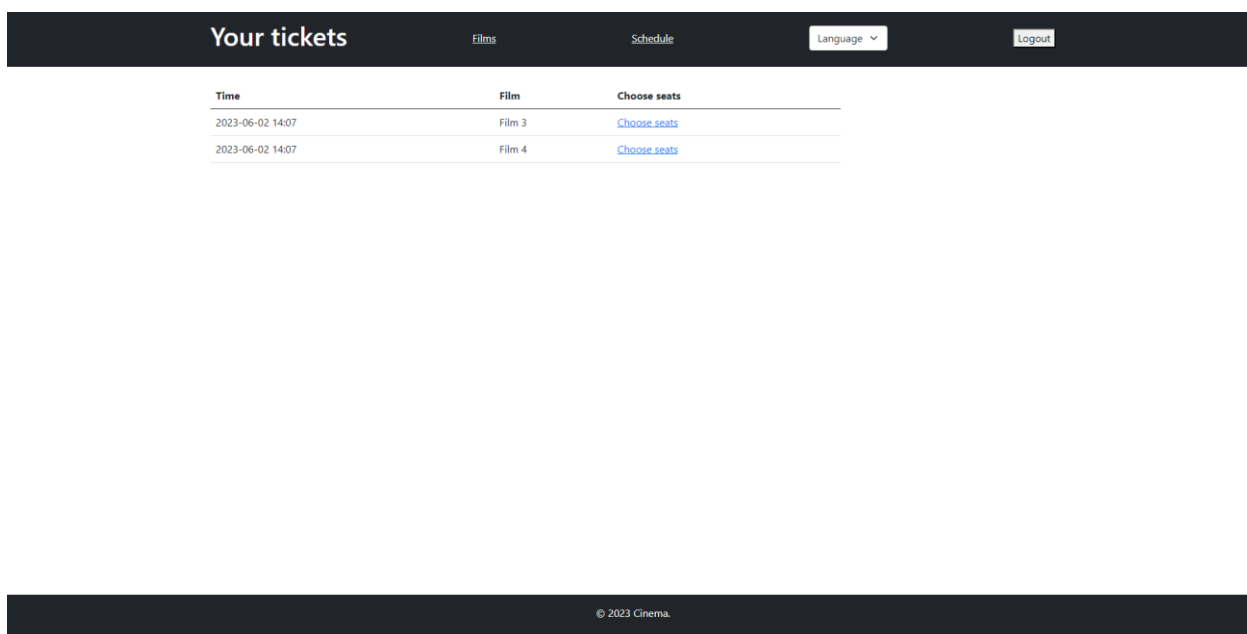


Рис 3.2.8 – Сторінка кабінету користувача



Рис 3.2.9 – Сторінка куплених місць користувача на сеанс

Користувач може переглянути фільми, що були та є в прокаті, а саме: назву, опис, жанр, тривалість та оцінку з IMDb (рис 3.2.10).

Фільми					
		Розклад Ваші квитки		Мова ▾	
				Вихід	
Назва	Опис	Жанр	Тривалість	IMDB	
Film 3	Descr 3	Genre 3	120	4.0	
Film 4	Descr 4	Genre 4	180	5.0	
JOHN WICK: CHAPTER 4	John Wick: Chapter 4 piles on more of everything -- and suggests that when it comes to a well-dressed Keanu Reeves dispatching his enemies in lethally balletic style, there can never be too much.	Action	169	8.0	

© 2023 Cinema.

Рис 3.2.10 – Сторінка фільмів кінотеатру

Назва фільму є клікабельною, натиснувши на неї, відбувається переадресація на сторінку фільму (рис 3.2.11).



JOHN WICK: CHAPTER 4

John Wick: Chapter 4 piles on more of everything -- and suggests that when it comes to a well-dressed Keanu Reeves dispatching his enemies in lethally balletic style, there can never be too much.

169 minutes

IMDB: 8.0

[Розклад](#)

© 2023 Cinema.

Рис 3.2.11 – Сторінка перегляду інформації конкретного фільму

Також, якщо на момент немає тривоги, та успішно придбались квитки, то на пошту аккаунту прийде повідомлення (рис 3.2.12).

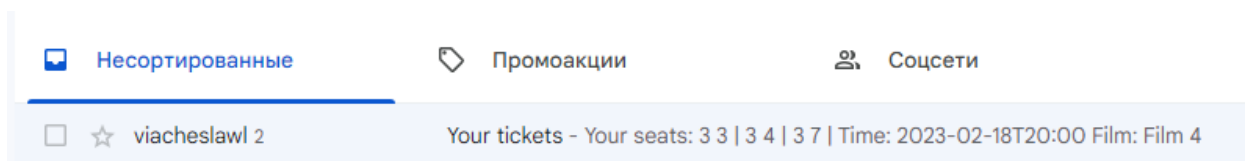


Рис 3.2.12 – Повідомлення про покупку квитків

3.3 Архітектура веб-додатку

Архітектура веб-додатку є надзвичайно важливою, оскільки вона визначає основну структуру, організацію та взаємозв'язки між компонентами додатку. Хороша архітектура дозволяє забезпечити ефективну роботу додатку, зменшити витрати на розробку та підтримку, покращити безпеку і масштабованість. Архітектура пакетів (рис 3.3.1).

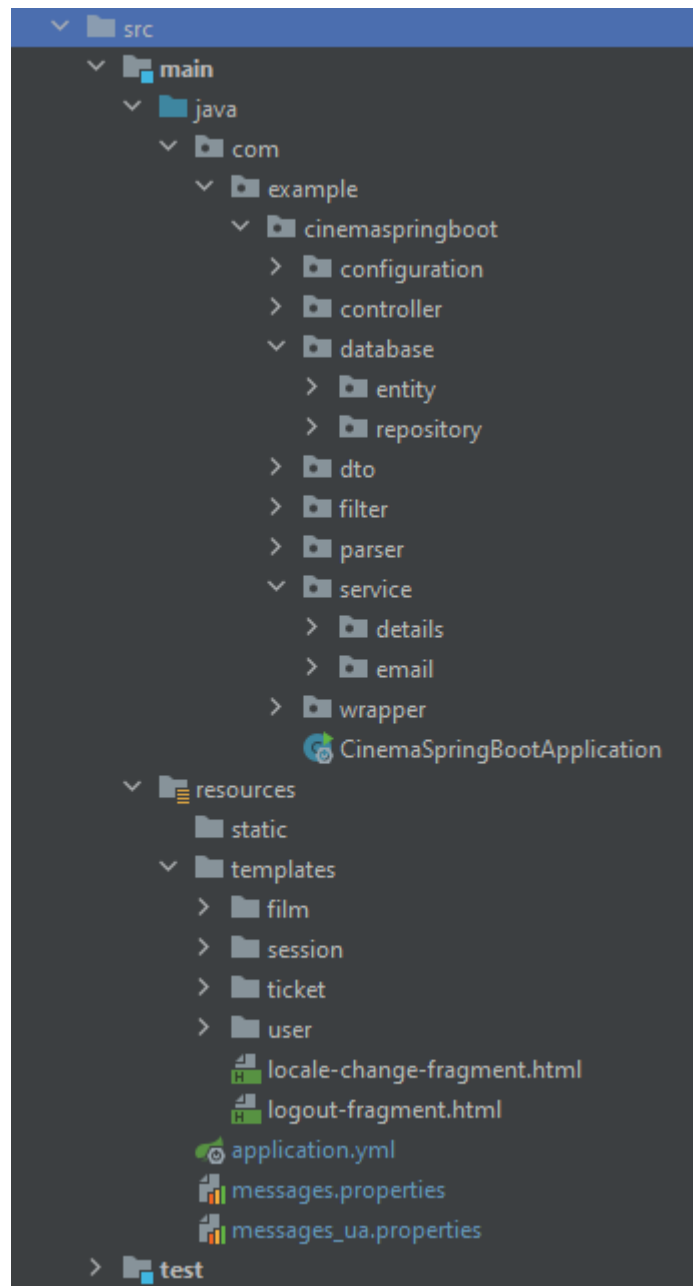


Рис 3.3.1 – Архітектура пакетів

Характеристика пакетів додатку:

- **Configuration:** захист проекту та обмеження доступу по ролям, конфігурація локалізації.
- **Controller:** класи взаємодії з веб-частиною додатку.
- **Database:** класи сутностей бази даних та репозиторії, що з нею взаємодіють.
- **Dto:** класи сутностей з валідацією полів.
- **Filter:** фільтри для локалізації та повітряної тривоги
- **Parser:** взаємодія з API тривоги
- **Service:** деталі користувача для захисту та сервіс відправки повідомлень
- **Wrapper:** клас-обгортка для опрацювання зайнятих місць
- **Templates:** html-сторінки відображення

ВИСНОВКИ

У ході роботи був розроблений веб-додаток для кінотеатру, який автоматизує процеси пов'язані з організацією покупки квитків. В рамках проекту були виконані наступні завдання:

- Проведено аналіз сучасних джерел, присвячених розробці веб-додатків для кінотеатрів, з метою створення постановки задач для подальшої реалізації.
- Розглянуто наявні інструменти для розробки веб-додатків та обрано ті, що найкраще відповідають потребам кінотеатру, забезпечуючи зручність використання та ефективне проектування додатку.
- Проаналізовано основні інструменти, що використовуються для створення веб-додатків, зокрема ті, які дозволяють керувати продажами квитків, відображенням розкладу сеансів та іншими функціями, необхідними для кінотеатру.
- Реалізовано функціонал збереження інформації про фільми, розклад сеансів та доступні квитки для продажу, а також можливість бронювання квитків онлайн.
- Забезпечено надійний захист додатку від втрати інформації та несанкціонованого доступу.
- Розмежовано рівні доступу для різних користувачів, таких як адміністратори, працівники та звичайні відвідувачі.
- Впроваджено базу даних для збереження і управління інформацією про фільми, сеанси та продаж квитків.

Також, розроблений додаток має потенціал для розширення функціоналу. Наприклад, можливе підключення платіжних систем для зручної онлайн-оплати квитків.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] "What is Web Application (Web Apps) and its Benefits."
<https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app> (accessed May 5, 2023).
- [2] "Postgresql:: Documentation." <https://www.postgresql.org/docs/> (accessed May 2, 2023).
- [3] "What is Normalization in DBMS (SQL)? 1NF, 2NF, 3NF Example."
<https://www.guru99.com/database-normalization.html> (accessed April 28, 2023).
- [4] "What is the digital revolution?"
<https://courses.minnalearn.com/en/courses/digital-revolution/the-digital-revolution/what-is-the-digital-revolution/> (accessed April 15, 2023).
- [5] "10 popular database management systems (DBMS) [List]."
<https://www.stackscale.com/blog/popular-database-management-systems/> (accessed May 10, 2023).
- [6] "What is ICT (Information and Communications Technology)?"
<https://www.techtarget.com/searchcio/definition/ICT-information-and-communications-technology-or-technologies> (accessed May 7, 2023).
- [7] "Sentiment Analysis on Social Network Data and Its Marketing Strategies: A Review," *ECS Trans*, vol. 107, no. 1, pp. 7417–7425, April 26, 2022, doi: 10.1149/10701.7417ECST.
- [8] "How to Build an Online Store in 2023: 9 Simple Steps."
<https://www.websitebuilderexpert.com/building-online-stores/> (accessed April 5, 2023).
- [9] "What is the digital revolution?"
<https://courses.minnalearn.com/en/courses/digital-revolution/the-digital-revolution/what-is-the-digital-revolution/> (accessed April 2, 2023).
- [10] "What is ICT (Information and Communications Technology)?"
<https://www.techtarget.com/searchcio/definition/ICT-information-and-communications-technology-or-technologies> (accessed May 9, 2023).
- [11] "Digital and Technological Solutions for Vietnam Tourism Services Development," *Lecture Notes in Networks and Systems*, vol. 522, pp. 515–522, 2023, doi: 10.1007/978-981-19-5292-0_48.
- [12] "The benefit and impact of E-commerce in tourism enterprises," 2021 2nd International Conference on Smart Computing and Electronic Enterprise: Ubiquitous, Adaptive, and Sustainable Computing Solutions for New Normal,

ICSCEE 2021, pp. 193–198, April 8, 2021, doi:
10.1109/ICSCEE50312.2021.9497947.

[13] "Web-based E-Commerce Products Grouping," J Phys Conf Ser, vol. 1898, no. 1, April 30, 2021, doi: 10.1088/1742-6596/1898/1/012018.

[14] "The History and Evolution of eCommerce - Translate By Humans." <https://translatebyhumans.com/blog/history-and-evolution-of-ecommerce/> (accessed April 1, 2023).

[15] "E-commerce evolution: Getting bigger and better every day." <https://www.the-future-of-commerce.com/2021/09/17/e-commerce-evolution/> (accessed April 5, 2023).

[16] "IntelliJ IDEA — IDE для разработки на Java и Kotlin." <https://www.jetbrains.com/idea/> (accessed April 6, 2023).

[17] "What is Agile? | Agile 101 | Agile Alliance." <https://www.agilealliance.org/agile101/> (accessed May 1, 2023).

[18] "Importance of Coping with Cyber Security Challenges in E Commerce Business," pp. 1596–1601, March 30, 2023, doi:
10.1109/IIHC55949.2022.10059851.

[19] "What is an IDE?" <https://www.redhat.com/en/topics/middleware/what-is-ide> (accessed April 4, 2023).

[20] "Strategic behavior of e-commerce businesses in online industry of electronics from a customer perspective," Adm Sci, vol. 10, no. 4, December 15, 2020, doi: 10.3390/ADMSCI10040078.

[21] "Figma: the collaborative interface design tool." <https://www.figma.com/> (accessed April 3, 2023).

[22] "Analytical Method for Social Network User Profile Textual Content Monitoring Based on the Key Performance Indicators of the Web Page and Posts Analysis," CEUR Workshop Proc, vol. 3171, pp. 1380–1402, April 8, 2022.

[23] "What's a design pattern?" <https://refactoring.guru/design-patterns/what-is-pattern> (accessed April 9, 2023).

Додаток А

Залежності проекту (Gradle)

```
plugins {
    id 'java'
    id 'org.springframework.boot' version '3.0.2'
    id 'io.spring.dependency-management' version '1.1.0'
}

group = 'com.example'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '17'

repositories {
    mavenCentral()
}

dependencies {
    compileOnly 'org.projectlombok:lombok:1.18.26'
    annotationProcessor 'org.projectlombok:lombok:1.18.26'

    implementation 'org.springframework.boot:spring-boot-starter-security'
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.springframework.boot:spring-boot-starter-aop'
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.thymeleaf.extras:thymeleaf-extras-springsecurity6:3.1.0.M1'
    implementation 'org.springframework.boot:spring-boot-starter-actuator'
```

```
    implementation 'org.springframework.boot:spring-boot-
starter-validation'

    runtimeOnly 'org.postgresql:postgresql'

    testImplementation 'org.springframework.boot:spring-boot-
starter-test'

    testImplementation 'org.springframework.security:spring-
security-test'
}

tasks.named('test') {
    useJUnitPlatform()
}
```

Додаток Б

Конфігурація Spring Security

```

@Configuration
public class SecurityConfiguration {

    @Autowired
    private UserDetailsServiceImpl userDetailsService;

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http)
    throws Exception {
        http
            .csrf().disable()
            .authorizeHttpRequests()
            .requestMatchers("/registration")
            .permitAll()
            .and()
            .authorizeHttpRequests()
            .requestMatchers("/film/update/**",
"/film/delete/**, " +
                "/session/update/**,
/session/delete/**")
            .hasAuthority("ADMIN")
            .and()
            .authorizeHttpRequests()
            .requestMatchers("/ticket/buy/**")
            .hasAuthority("USER")
            .and()
            .formLogin()
            .loginPage("/login")
            .defaultSuccessUrl("/session/all", true)
            .permitAll()
    }
}

```

```
        .and()
            .logout()
            .logoutUrl("/logout")
            .logoutSuccessUrl("/login")
            .invalidateHttpSession(true)
            .deleteCookies("JSESSIONID")
            .permitAll()
        .and()
            .authorizeHttpRequests()
            .anyRequest()
            .authenticated();

    return http.build();
}

@Bean
protected PasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}

@Bean
protected DaoAuthenticationProvider
authenticationProvider() {
    DaoAuthenticationProvider daoAuthenticationProvider =
    new DaoAuthenticationProvider();

    daoAuthenticationProvider.setPasswordEncoder(passwordEncoder());

    daoAuthenticationProvider.setUserDetailsService(userDetailsService);

    return daoAuthenticationProvider;
}
```

Додаток В

Контроллер реєстрації

```
@Controller
public class RegistrationController {

    private final UserRepository userRepository;

    private final PasswordEncoder passwordEncoder;

    public RegistrationController(UserRepository userRepository,
        PasswordEncoder passwordEncoder) {
        this.userRepository = userRepository;
        this.passwordEncoder = passwordEncoder;
    }

    @GetMapping("/registration")
    public String saveUser(UserDto userDto, Model model) {
        return "user/registration";
    }

    @PostMapping("/registration")
    public String saveUser(@Valid UserDto userDto, BindingResult
        bindingResult) {
        if (bindingResult.hasErrors()) {
            return "user/registration";
        }
        userRepository.save(User.builder()
            .login(userDto.getLogin())
            .email(userDto.getEmail())
            .password(passwordEncoder.encode(userDto.getPassword()))
            .role(new UserRole(1, "User"))
            .build());
        return "redirect:/session/all";
    }
}
```