

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

**РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ РЕЛЯЦІЙНОЇ
ТА НЕРЕЛЯЦІЙНОЇ БАЗ ДАНИХ**

**Текстова частина до курсової роботи
за спеціальністю „Комп’ютерні науки”**

Керівник курсової роботи
к.ф.-м.н, ст. викладач Ющенко Ю.О.

“ ____ ” _____ 2021 р.

Виконав студент
Мацюк В.О.
“ ____ ” _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,

доцент, к.ф.-м.н.

_____ С. С. Гороховський

(підпис)

“ _____ ” _____ 202_ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту _____ Мацюк Вікторії
_____ 4 _____ курсу факультету інформатики

ТЕМА: _____ Реалізація інформаційної системи на основі реляційної та нереляційної баз даних

Вихідні дані: Інформаційна система, розроблена на мові PHP, що взаємодіє з системою керування базами даних MySQL та MongoDB завдяки локальному веб-серверу Open Server Panel.

Індивідуальне завдання

Вступ

1. Аналіз предметної області. Постановка задач
2. Дослідження інформаційних систем та баз даних
3. Опис реалізації програмного продукту

Висновки

Список використаних джерел

Додатки (за необхідністю)

Дата видачі „_____” _____ 2021 р. Керівник _____
к.ф.-м.н, ст. викладач _____ Ющенко Ю.О.

Завдання отримала _____

Тема: Реалізація інформаційної системи на основі реляційної та нереляційної баз даних

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи	01.10.2020	
2.	Пошук тематичної наукової літератури	15.11.2020	
3.	Огляд аналогічних інформаційних систем	01.01.2021	
3.	Розробка алгоритму та структури програми	15.01.2021	
4.	Створення бази даних	02.02.2021	
5.	Розробка програмного продукту	15.03.2021	
6.	Тестування розробленого програмного продукту	20.03.2021	
7.	Написання пояснювальної роботи	01.04.2021	
8.	Подання першої версії пояснювальної роботи науковому керівнику	04.04.2021	
9.	Корегування роботи згідно зауважень наукового керівника	06.04.2021	
10.	Створення презентації для захисту роботи	08.04.2021	
11.	Захист курсової роботи	19.04.2021	

Студент _____ Мацюк В.О. _____

Керівник _____ Ющенко Ю.О. _____

“ _____ ” _____ 2021 р.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1: Аналіз предметної області. Постановка задач	7
1.1 Аналіз стану питання та обґрунтування теми	7
1.2 Огляд існуючих аналогів	8
1.3 Опис задач для розробки	12
РОЗДІЛ 2: Дослідження інформаційних систем та баз даних	13
2.1 Інформаційні системи	13
2.1.1 Системи обробки транзакцій (TPS).....	15
2.1.2 Інформаційні системи керування (MIS)	17
2.1.3 Маркетингові інформаційні системи	18
2.2 Класифікація ІС за ступенем автоматизації.....	20
2.3 База даних.....	21
2.4 Реляційні та нереляційні БД.....	22
РОЗДІЛ 3: Опис реалізації програмного продукту	26
3.1 Аналіз технічного завдання.....	26
3.2 Обґрунтування обраних засобів розробки	28
3.3 Опис моделі даних.....	32
3.4 Опис різниці у розробці	35
3.5 Тестування програми і результати її виконання	37
Висновок.....	42
Список використаної літератури	44

ВСТУП

У процесі еволюції головним засобом пристосування людського населення до змін навколишнього середовища було пізнання дійсності та закріплення результатів в інформації, що є основою для реагування. Завдяки необхідності використання колективних зусиль у цьому реагуванні, забезпечення спадкоємності досвіду відбувався значний розвиток мови, письма та інших соціалізованих форм фіксації інформації.

Сьогодні більшість задач базується на тих даних, які століттями накопичувались та оброблювались: у доіндустріальному періоді людськими силами, пізніше – автоматизованими обчислювальними системами. З упровадженням електронних інформаційних технологій значна частина сфер життя перекочувала з паперових до цифрових носіїв: економічна, медична, адміністративна, навчальна, тощо. До таких рішень вдалися також бізнесові галузі, і притому не тільки з метою полегшення ведення справ, а й збільшення привабливості серед клієнтів.

Серед тих напрямлень, де інформаційні системи спрощують життя і власникам, і відвідувачам – є мережі кінотеатрів. Більшість пам'ятає, як ще донедавна список фільмів, розклад кіносеансів, наявність вільних квитків можна було дізнатись тільки відвідавши один з кінотеатрів. Станом на 2019 рік [1] у Києві діє 56 кінотеатрів, а загалом в Україні їх близько 200, для порівняння, у 2016 році їх налічувалось 189. Кіноіндустрія росте, вкладаються мільйонні інвестиції [2] у розвиток українського кінематографу, що напевне спричинить ріст і мережевих, і незалежних залів.

На жаль, досі багато кінозалів не мають своїх веб-сайтів для інформаційного забезпечення та онлайн купівлі квитків на сеанси, що не є привабливим для сучасного вимогливого глядача. Отож, на висококонкурентному ринку бізнесу потрібно якісно підходити до систем керування інформацією та забезпечувати нею не тільки працівників, а й клієнтів.

Тому дослідження та автоматизація цієї галузі є не тільки корисним для розробника, але й фінансово вигідним для обох сторін.

Метою цієї роботи є проектування і створення веб-сайту для продажу квитків до кінотеатру на фоні дослідження інформаційних систем та баз даних. Теоретична частина полягатиме в аналізі інформаційних систем, вживаних сховищ інформації, а практична – в закріпленні отриманих знань, а саме в розробці сучасної системи, яка полегшить роботу кінотеатрів та приверне нових клієнтів.

Завданням є реалізація програмного продукту з розподіленням ролей: адміністратор та покупець. Дослідження предметної області, типів інформаційних систем та сучасних баз даних.

Об'єктом дослідження є існуючі веб-сайти кінотеатрів, інформаційні системи та бази даних.

Розроблений продукт є підґрунтям для реалізації високонавантажених складних систем, які забезпечують різним групам користувачів окремий функціонал у системі. Продукт базується на клієнт-серверній архітектурі та розроблений за допомогою веб-мов програмування.

Структура роботи: робота складається з трьох розділів.

У першому розділі наведено інформацію про сучасний стан справ обраної предметної області, а також обґрунтування актуальності та ефективності теми. Зроблено огляд існуючих аналогів на основі яких поставлено задачі для дослідження та розробки.

Другий розділ присвячено теоретичній частині. Досліджено інформаційні системи, їх компоненти та типи. Проаналізовано реляційні та нереляційні бази даних, їх переваги та відмінності.

У третьому розділі детально описано технічне завдання для розробки, обґрунтовано вибір інструментів розробки. Наданий опис різниці роботи у розробці з MongoDB та MySQL фрагментами коду. У підрозділі 3.5 проведено тестування програмного продукту.

РОЗДІЛ 1: Аналіз предметної області. Постановка задач

1.1 Аналіз стану питання та обґрунтування теми

Інформація оточує людство у всіх сферах життя і ефективний підхід до її обробки та візуалізації є необхідним умінням кожного сучасного розробника. У галузі бізнесу автоматизовані інформаційні системи є не тільки інструментом спрощення його ведення, а ще й підняття ефективності [3]. Впровадження АІС буде привабливим не лише власнику бізнесу та рядовим працівникам в забезпеченні їх роботи, а й директору продажів, маркетологам та самим клієнтам.

Для власника бізнесу це дасть інструментарій контролю за веденням звітностей або ж швидкий доступ до необхідних йому цифр. Директор з продажів та маркетологи завдяки автоматизації процесів зможуть наочно бачити результати проведених рекламних акцій в кількісному виді або ж систематизувати цільові групи клієнтів для подальшої якісної роботи з ними. Звісно, найбільша цільова група автоматизації це рядові працівники та клієнти. Спрощення ведення роботи – є головним плюсом для співробітників бізнесу. Щодо клієнтів – це постійний доступ до інформації, що в умовах високої конкурентності є основним критерієм вибору послуг серед клієнтів.

В розрізі обраної предметної області, а саме кінотеатрального бізнесу, веб-сайт з продажу квитків є ключовим аспектом в роботі з клієнтами. По-перше, серед високої конкуренції, яку складають мережеві кінотеатри з якісними веб-сторінками, для комунальних та незалежних підприємств важливо також залучати свій бізнес до онлайн простору. По-друге, в умовах пандемії надзвичайно важливо зменшити кількість нецільових відвідувань (таких, як покупка квитків у касі) до кінотеатру.

Існує практика серед кінотеатрів без своїх інтернет-сторінок співпрацювати з веб-сайтами, які продають квитки до багатьох кінотеатрів. Проте, це економічно не вигідно, адже такі веб-сайти беруть величезну комісію за розміщення, тому бізнесу варто не економити гроші на розробку власної платформи, адже в перспективі це збереже величезні суми.

1.2 Огляд існуючих аналогів

У мережі Інтернет є достатня кількість веб-сайтів з обраної раніше тематики, щоб визначити плюси та мінуси існуючих систем. Було обрано три кардинально різні системи для усестороннього аналізу та демонстрації різниці.

Перший веб-сайт, який був проаналізований – це сторінка кінотеатру «Планета» [4], який розташований у Хмельницькому. Це комунальний кінотеатр, один з перших у цьому місті. Його особливістю [5] є розташування у будівлі колишнього пожежного депо, що є пам'яткою архітектури місцевого значення.

Повертаючись до веб-сайту, користувач одразу потрапляє на головну сторінку з розкладом сеансів на тиждень:

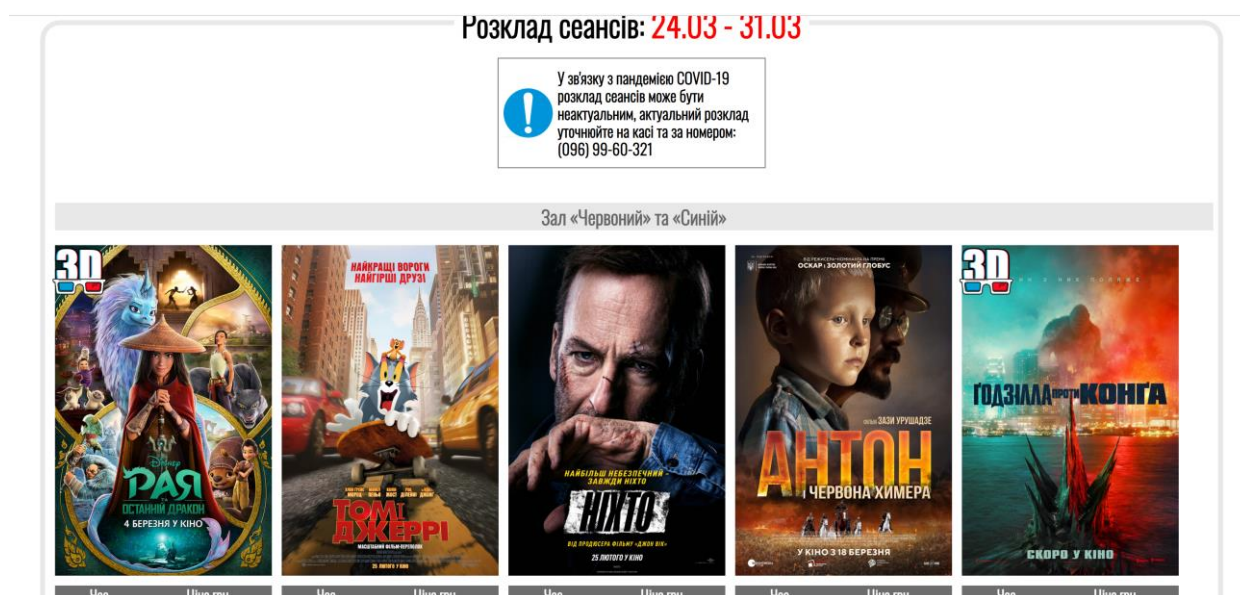


Рисунок 1.1 – Веб-сайт кінотеатру «Планета»

Окрім афіші активних сеансів на головній сторінці є розділ «Акції» та «Контакти», а також посилання на профілі кінотеатру в соціальних мережах, в тому числі неактивне посилання на мережу «Вконтакті», заборонену в Україні [6].

На сторінці кінофільму структуровано подано деталі, такі як рік виходу, країна, режисер, тощо. Також надається короткий опис сюжету та посилання на трейлер. Можливості придбати чи хоча б забронювати квитки на бажаний сеанс немає.

Отже, проаналізувавши даний веб-сайт, можна чітко виділити його плюси та мінуси. Серед плюсів: зручний та інтуїтивно зрозумілий інтерфейс, чітко надана інформація про кінофільм, є можливість дізнатись про актуальні акційні пропозиції та долучитись до соціальних мереж кінотеатру. Найвагоміший мінус даного веб-сайту – це те, що це більше інформаційна сторінка, яка не надає клієнту зробити головне – придбати квитки на сеанс. Окрім цього, інформація надається загалом про сеанси на тиждень без можливості фільтрування. Зазвичай, на вихідні кількість сеансів збільшується, тому є великі сумніви щодо актуальності інформації на сайті.

Наступним було проаналізовано нетиповий для обраної області веб-сайт, проте цікавий в плані проектування та роботи – сторінка кінодрому на кіностудії О.Довженка [7]. Це автомобільний кінотеатр, де щодня на спеціально обладнаній для перегляду стоянці проводять кілька сеансів в різний час.

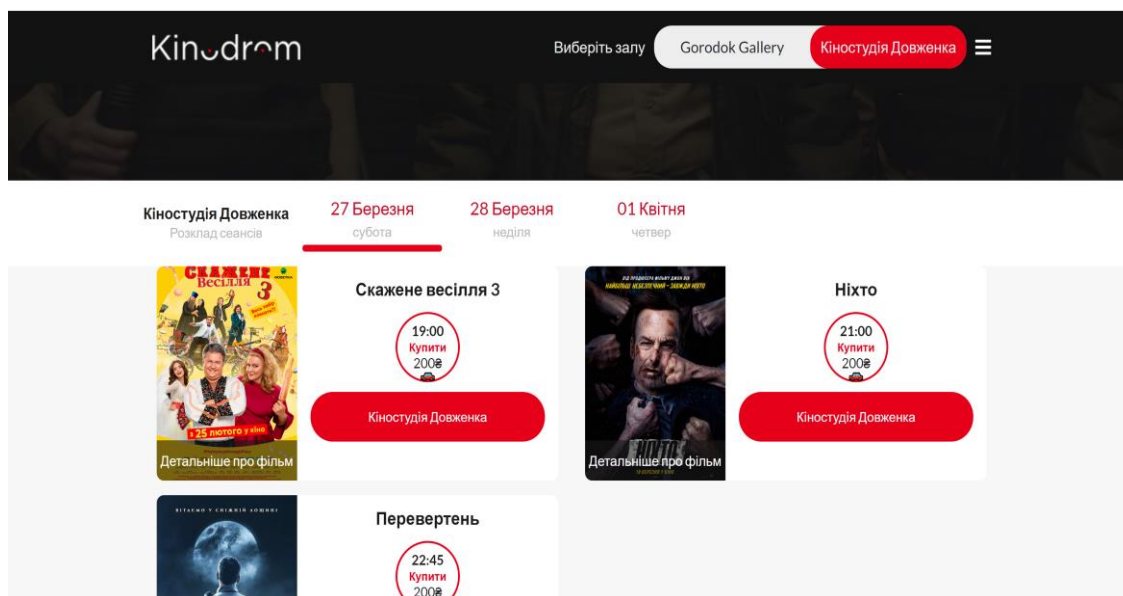


Рисунок 1.2 – Веб-сайт кінодрому на кіностудії О.Довженка

Відвідуючи сайт, неозброєним оком видно, що адміністрація кінодрому дійсно з прискіпливістю віднеслась до його створення. Сучасний та цікавий інтерфейс, який інтуїтивно зрозумілий користувачу. На головній сторінці є фільтрований розклад сеансів на найближчі три дні, посилання на розділ «Контакти» та «Бар», який на території комплексу працює завдяки онлайн

замовленням із сайту. Також, на сторінці можна одразу придбати квиток на сеанс.

Перейшовши на сторінку кінофільму можна, як і в проаналізованому раніше кінотеатрі, побачити деталі. Окрім цього, показано два найближчих сеанси на які можна одразу придбати квитки. При покупці відображається спливаюче вікно з розташуванням автомобілів на території. Місця-автомобілі поділені по виду: седан, хечбек, кросовер, джип або мікроавтобус для якісного перегляду кінофільмів. Квиток можна придбати/забронювати вказавши ім'я, телефон, номер автівки та електронну адресу, на яку надійде підтвердження броні.

Отже, проаналізувавши даний веб-сайт можна виділити такі плюси: інтерфейс, нестандартний підхід до вибору місць, можливість одразу придбати квиток. Серед мінусів: придбати квиток можливо тільки після надходження листа на пошту, що займає час. А також можливість придбати тільки один квиток на машину за один сеанс роботи, що є незручним для покупки квитків для компанії.

Останнім проаналізованим кінотеатром – є мережа «Планета Кіно» [8], яка налічує 9 кінотеатрів у містах Київ, Одеса, Львів, Харків, Дніпро та Суми. Це друга за кількістю кінотеатрів мережа в Україні, тому її було обрано для аналізу та порівняння.

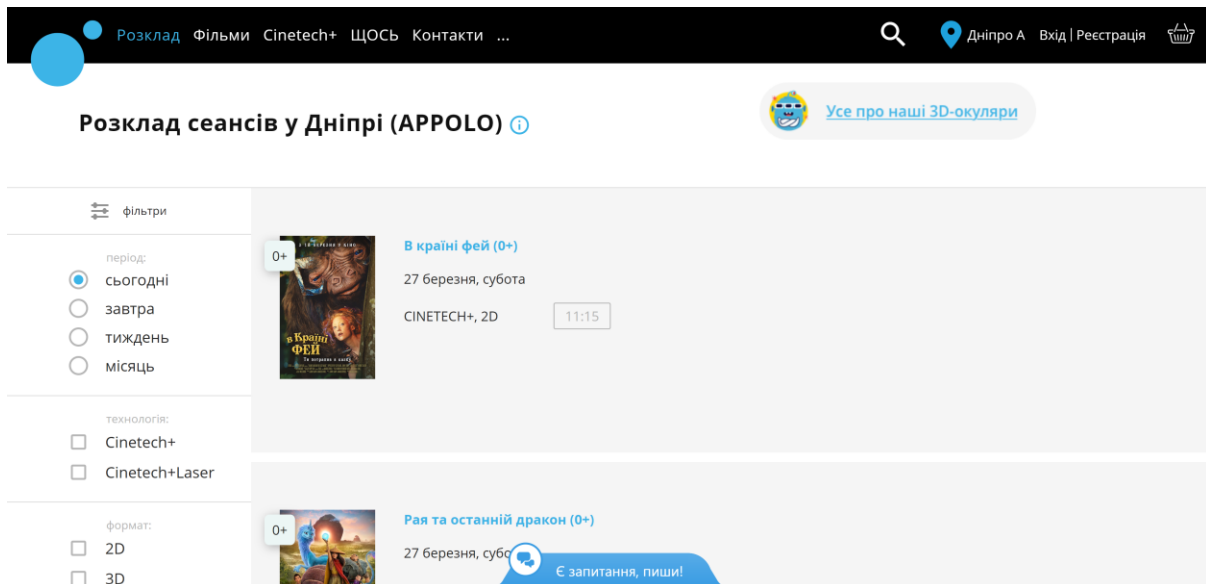


Рисунок 1.3 – Веб-сайт мережевого кінотеатру «Планета Кіно»

Обравши кінотеатр, користувач потрапляє на сторінку з розкладом сеансів, які можна обрати за періодом: на сьогодні, на завтра, через тиждень або через місяць. Окрім цього, завдяки наявності різних технологій та форматів у залах, користувач може обрати бажане за допомогою фільтру. На панелі зверху можемо перейти на сторінки «Фільми» (чого не було в проаналізованих раніше кінотеатрах), технологію «Cinetech+», «ЩОСЬ» - маркетингова сторінка з інформацією щодо мережі, «Контакти», тощо.

На сторінці обраного кінофільму користувач бачить деталі, короткий опис та має можливість придбати квиток з корзини. Квитки може придбати як зареєстрований користувач, так і незареєстрований. До сайту прив'язані платіжні системи «LiqPay», «ApplePay» та «24Pay». Також «Планета Кіно» має своїх ботів у популярних месенджерах, та мобільну версію.

Мережа ввела заохочувальну систему для користувачів у вигляді бонусів. Кожен може зареєструватись та отримувати бонусні кошти з покупки. Також в особистому кабінеті можна переглянути придбані квитки та прив'язати кредитні карти для швидкого розрахунку.

Отже, веб-сайт мережі «Планета Кіно» зручний, інформативний та надає великий вибір функціоналу користувачу. Проте як для незалежного чи

комунального кінотеатру функціонал може бути занадто широким та непотрібним, а чим він вагомніше, тим більші кошти має віддати бізнес. Тому проаналізувавши всі плюси та недоліки можна чітко описати задачі, які буде виконувати розроблена система.

1.3 Опис задач для розробки

Завдяки аналізу аналогічних систем було виділені наступні задачі:

1. Розробити сучасний веб-сайт для бронювання квитків в кінотеатр із зручним та інтуїтивно зрозумілим інтерфейсом для користувачів.
2. Забезпечити адміністратора веб-сайту зручним функціоналом для підтримання актуальності інформації щодо фільмів та сеансів.
3. Реалізувати можливість обрання та бронювання місць в зал на певний сеанс та відправлення повідомлення на пошту.
4. Забезпечити якісний функціонал у каталозі фільмів: можливість сортування сеансів та фільтрації.
5. Реалізувати описані умови у реляційній та нереляційній базах даних, таким чином отримавши два проекта.

РОЗДІЛ 2: Дослідження інформаційних систем та баз даних

2.1 Інформаційні системи

Інформаційна система – це сукупність взаємопов’язаних компонентів, які працюють разом для збору, обробки, зберігання та розподілу даних з метою підтримки прийняття рішень, координації, контролю, аналізу проблем та їх візуалізації. Завдяки постійному розвитку обчислювальних технологій та способів їх використання, сучасні інформаційні системи - це в більшості комп’ютерні інформаційні системи, які обробляють мільярди даних для виведення бажаної інформації за зовсім короткий проміжок часу.

Складовими інформаційних систем є наступні компоненти [9][10]:



Рисунок 2.1 – Складові інформаційних систем

- Комп’ютерна техніка – фізичне обладнання, що використовується для введення, виведення та обробки необхідних даних. В залежності від типу користувачів – організація чи персональне використання – діапазон та

кількість пристроїв може варіюватись (збільшуватись або зменшуватись, відповідно).

- Програмне забезпечення – ділиться на два основні класи: системне програмне та прикладне програмне забезпечення.

Системне програмне забезпечення – це операційна система, яка безпосередньо забезпечує працездатність комп'ютера, керує апаратним забезпеченням, файлами даних та іншими системними ресурсами. Прикладне програмне забезпечення - це програми, призначені для реалізації конкретних завдань користувачів.

Важливо, що інформаційні системи, які не використовують комп'ютери, мають компоненти програмного ресурсу. Це справедливо навіть для інформаційних систем давніх часів, або для ручних та машинних інформаційних систем, що використовуються і донині. Усі вони потребують програмного забезпечення у вигляді інструкцій та процедур обробки інформації задля правильного збору, обробки та поширення інформації серед своїх користувачів.

- Телекомунікаційні мережі - це комплекс технічних засобів телекомунікацій та споруд, призначених для маршрутизації, комутації, передавання та/або приймання інформації або повідомлень завдяки електромагнітним системам між кінцевим обладнанням. З їх допомогою користувачі отримують доступ до інформаційних ресурсів, таких як великі бази даних, та до інших осіб, наприклад, колег, клієнтів, друзів, тощо. Інтернет можна вважати мережею мереж. Проте інформаційна система може існувати без можливості зв'язку - перші персональні комп'ютери були автономними машинами, які не виходили в Інтернет. Однак у сучасному гіперпов'язаному світі це надзвичайно рідкісний комп'ютер, який не підключається до іншого пристрою чи мережі.
- Людські ресурси - без людей на місці вся ідея інформаційної системи не реалізується. Вони є користувачами та поділяються на три типи з урахуванням їх функціоналу в системі: внутрішні (корпоративні

співробітники, адміністратори), зовнішні (наприклад, постачальники, які використовують систему, як один з інструментів планування) та кінцеві користувачі (споживачі та корпоративні співробітники).

- Бази даних - більшість інформаційних систем, в першу чергу, є засобами обробки та візуалізації даних, що зберігаються у сховищах або, так званих, базах даних. Детальніше їх буде розглянуто у підрозділі 2.3.

Сучасні інформаційні системи проектуються з урахуванням комплексних вимог щодо їх роботи. Перш за все, це стосується доступу до будь-якої необхідної інформації в заданий час, іншими словами, забезпечення високої продуктивності, надійності, а також коректності даних. Система повинна бути ефективною, приносити користь та зиск при мінімальних витратах на її розробку та подальшу підтримку. Враховується також забезпечення простотою та швидким освоєнням спілкування з БД, легкість та швидкість встановлення на пристрої. І найголовніше, при розробці ІС потрібно враховувати живий розвиток інформаційних технологій, тому система повинна функціонувати достатньо довгий період часу, бути адаптивною під пристрої та операційні системи, а також легко переноситись з однієї платформи на іншу.

Інформаційні системи можна класифікувати за багатьма ознаками. У даній роботі буде розглянуто два напрямки: класифікація за типами рішень та ступенем автоматизації. Основними класами за напрямом виконання рішень є:

- Системи обробки транзакцій (TPS)
- Інформаційні системи керування (MIS)
- Маркетингові інформаційні системи

Детальніше наведені класи буде розглянуто в наступних підрозділах.

2.1.1 Системи обробки транзакцій (TPS)

Системи обробки транзакцій (TPS) [11] були одні із перших комп'ютеризованих систем. Вони використовуються для обробки господарських операцій в організаціях, і таким чином, підтримують діяльність підприємства.

Іншими словами, TPS обслуговують операційний рівень організацій – виконують та реєструють щодня рутинні операції, необхідні для управління бізнесом, такі як продаж, покупку, зняття коштів, повернення коштів. Вони виступають такими собі зв'язувальними елементами з зовнішніми суб'єктами (наприклад, постачальниками чи дистриб'юторами). TPS реєструє саму транзакцію, а також усі її наслідки в базі даних та видає потрібні документи. У свою чергу, TPS за оперативністю обробки даних поділяються на пакетну обробку (Batch processing) та обробку в реальному часі (Real-time processing).

При пакетній обробці відбувається уточнення кластерів транзакції з використанням комп'ютерної системи в один і той же момент часу. Цей метод призначений для ефективного розбиття громіздких програм, проте завдяки цьому має певний недолік – затримку результату транзакцій. Зібрані дані зберігаються у пакетному вигляді з наданням можливості обробки у будь-який час.

Обробка у реальному часі забезпечує миттєву обробку даних з ціллю швидкої перевірки транзакції. Цей метод є ефективним в роботі з одиничними транзакціями та гарантує миттєву відповідь на умову результату транзакції.

Властивості TPS:

- Швидкість: час від введення транзакції до отримання результату становить декілька секунд або менше.
 - Стандартизація: кожна транзакція повинна мати однаковий механізм обробки.
 - Контрольована обробка: якщо ролі та обов'язки в системі розподілені, транзакції повинні відбуватись з врахуванням цих умов.
 - Надійність: мінімізація поломок – кожна несправність заважає транзакції.
- У разі виникнення несправностей відновлення має бути швидким та точним.

Основні складові системи обробки транзакцій:

- Вхідні дані: отримані внаслідок транзакції документи, які слугують вхідними даними для системи TPS.
- Обробка: процес розбиття або подальша обробка інформації із вхідних даних.
- Сховище: зберігання інформації у пам'яті TPS.
- Вихідні дані: будь-які згенеровані документи є виходом.

Висновок:

Системи обробки транзакцій TPS - це інформаційні системи, які слугують задля накопичення, зберігання та обробки повсякденних операцій. Як приклад, це і системи обробки замовлень, і записи співробітників або ж системи бронювання місць у готелях. В свою чергу, TPS за оперативністю обробки даних поділяються на пакетну обробку (генерація рахунку, перевірка чеків) та обробку в реальному часі (термінали торгівельних точок). Жодна організація не обходиться без даного типу систем, адже мінімальні рутинні операції – це певний фундамент їх роботи.

2.1.2 Інформаційні системи керування (MIS)

Інформаційна система керування (MIS) [11] - це інформаційна система, яка надає точну, своєчасну та структуровану інформацію, дякуючи чому користувачі мають змогу приймати рішення, вирішувати проблеми з врахуванням даних, контролювати діяльність та відслідковувати прогрес. MIS за допомогою рутинних алгоритмів здійснює аналіз вхідних даних, узагальнює їх, і у результаті надає користувачам конкретні рушії для моніторингу та прогнозування. Зазвичай дані системи стосуються внутрішніх джерел інформації, і в ідеальному варіанті витягують дані транзакцій із базових TPS, компілюють їх та видають інформаційні продукти у формі звітів, зображень або прогнозів. Тип і форма цих вихідних продуктів залежать від інформаційних потреб конкретного користувача.

Властивості MIS [12]:

- Системність: ІС побудована на цілісному підході до опановування інформації та з урахуванням концепції, заради якої вона створена.
- Орієнтованість на управління: при розробці потрібно починати з окреслювання управлінських потреб та бізнес-цілей.
- Направлена на потреби: як правило, MIS розроблюються під конкретні підрозділи в організаціях з урахуванням саме їх функціоналу.

Основні складові MIS:

- Дані (дані з проведених транзакцій, цифри і т.д.)
- Користувачі
- Бізнес-процедури - це сукупність правил або керівних принципів у організації для використання комп'ютерної інформаційної системи. Процедури можуть варіюватися в залежності від відділів або організації.
- Програмне та апаратне забезпечення

Висновок

Інформаційна система керування – об'єднання процедур, даних, інформаційних технологій та працівників, яке сприяє прийняттю ефективних рішень на менеджерському рівні. В багатьох сферах системи керування працюють разом з вище описаними системами обробки транзакцій. З даним типом систем, окрім бізнесу, людство стикається щодня, наприклад, шкільна інформаційна система керування, з записами про школярів, вчителів, табелі, відвідування занять тощо, або інформаційна система бібліотеки, де ведуться звіти про кожного відвідувача, заведення їх формулярів.

2.1.3 Маркетингові інформаційні системи

Маркетингова інформаційна система [13]– це система, яка забезпечує систематичний збір, аналіз, інтерпретацію, зберігання та розповсюдження ринкової інформації, як із внутрішніх, так і із зовнішніх джерел, серед

маркетологів на постійній основі. По суті, це підвид інформаційної системи керування, яка орієнтується на підтримку прийняття маркетингових рішень, як-от ціноутворення, розповсюдження, тощо.

Основні властивості маркетингової інформаційної системи:

- Робота у реальному часі: система має працювати з актуальними даними для сприяння у прийнятті ефективних рішень.
- Гнучкість: система має бути скоригована на швидкий прийом та легке встановлення оновлень.
- Робота з великим обсягом інформації
- Неперервність: збір інформації відбувається неперервно.

Основні компоненти маркетингової інформаційної системи:

- Система внутрішнього обліку: дані про продажі та/або маркетингову діяльність всередині компанії. Поточні дані про продажі повинні регулярно підтримуватися та бути завжди актуальними. Як правило, в залежності від бізнес-процесів, компанії ведуть декілька баз даних, таких як база клієнтів, товару чи продавця.
- Система маркетингової розвідки: сукупність процедур та джерел даних, що використовуються маркетологами для збору інформації із середовища, яку вони можуть використовувати при прийнятті рішень. Існують розумні системи, які можна інтегрувати до існуючої системи та отримувати дані з відстежування веб-сайтів конкурентів, галузевих торгових публікацій та спостережень на ринку колег.
- Система маркетингових досліджень: систематичний збір, організація, аналіз та інтерпретація даних із маркетингового середовища для пошуку шляхів вирішення проблем.
- Система управління маркетингом: програмне забезпечення, при якому з частою допомогою аналітичних моделей, маркетингові менеджери можуть приймати рішення.

Висновок

Маркетингові інформаційні системи розроблені для підтримки прийняття управлінських рішень в галузі маркетингу. Такі системи полегшують життя різних напрямків бізнесу: ресторанного, готельного, рекламного, магазинах, кінотеатрах, тощо. При відсутності даних систем персонал мав би власноруч аналізувати звіти, робити прорахунки, щоб займало багато часу та не завжди було б коректним, враховуючи людський фактор.

Отже, підсумовуючи огляд певних типів інформаційних систем, можна дійти до висновку, що в світі інформаційних технологій не варто гребувати автоматизацією процесів. Усі наведені типи ІС можна комбінувати та впроваджувати великі комплексні системи. Загалом, чим більше систем у бізнесі, тим ефективнішим він може стати.

2.2 Класифікація ІС за ступенем автоматизації

Окрім вище описаних основних типів інформаційних систем, усіх їх також можна розділити за рівнем автоматизації інформаційних процесів:

- Ручні ІС: збір, обробка, консолідація та візуалізація даних відбувається виключно з допомогою людини. Як приклад, можна привести конспект чи ведення обліку на папірцях.
- Автоматичні ІС: всі операції з даними виконуються без втручання людської сили. Це повністю керовані комп'ютерними технологіями системи.
- Автоматизовані ІС: найпоширеніші ІС, які поєднують в собі два вище зазначені типи. Зазвичай, головну роль відіграє комп'ютер, проте забезпечується також участь людини у процесі обробки інформації. Наприклад, веб-сайти з системами керування вмісту.

Дана практична робота буде побудована на розумінні саме автоматизованої інформаційної системи. Тому пропонується перейти до головного компоненту ІС, де збираються та зберігаються дані – до баз даних.

2.3 База даних

База даних (БД) [14] – набір логічно взаємопов'язаних даних та їх описів, які використовуються спільно та розроблені для задоволення інформаційних потреб користувачів. До появи комп'ютерних технологій базою даних вважалася бібліотека із книгами, чи записна книга з записами про прізвище, ім'я, адрес, контакти. Сьогодні база даних може містити сотні і тисячі взаємопов'язаних даних про ті чи інші об'єкти.

Як було зазначено вище, база даних використовується для задоволення інформаційних потреб багатьох користувачів, тому вона має орієнтуватись на інтегровані запити, а не на одну програму. Це дозволяє в значній мірі зменшити надлишковість інформації.

У основі всього лежать дані, які є динамічною сутністю, а отже спосіб їх зберігання може різнитись. Усе це, що комплексно можемо охарактеризувати як перехід від структури БД до необхідної структури в програмі користувача, відбувається завдяки систем керування базами даних (СКБД). СКБД забезпечує також доступ до даних, ефективне керування ними, а також підтримує їх захист. Серед типів СКБД:

- Ієрархічні бази даних: структурування даних відбувається за допомогою вузлів відносин батьки-діти. При чому кожен дочірній запис має лише одного «батька», а ось «батьки» можуть мати кількох «дітей». Є підвидом мережових БД.
- Мережові бази даних: дані представлені в вигляді колекцій записів, а зв'язки – у вигляді наборів. Мережові вузли можуть мати декількох «батьків».
- Графічні бази даних: для структурування даних використовуються графіки.
- Реляційні бази даних: дані зберігаються у табличній формі стовпців та рядків.

- Нереляційні бази даних (NoSQL): структурування за допомогою колекцій різних категорій. Вище наведені ієрархічні, мережеві та графічні БД є нереляційними.

Оскільки зазвичай виділяють, як основні типи саме реляційні та нереляційні БД, розглянемо їх детальніше.

2.4 Реляційні та нереляційні БД

Реляційна база даних - це база даних, в основі якої реляційна модель даних. Цей термін уперше був використаний у 1970 році Е.Ф. Коддом у його науковій роботі. Вона структурована, тобто дані впорядковані в таблиці з рядками та стовпцями. Дані в цих таблицях мають зв'язки між собою, які встановлюються за допомогою первинного та зовнішнього ключів. Ключ - це унікальний ідентифікатор, який надається рядку даних, що містяться в таблиці.

Запити до даних у системі керування реляційними базами даних виконуються завдяки мові структурованих запитів (SQL), яка є надійною та призначеною для управління даними. Це зручний інструмент, який підтримується реляційними СКБД та спрощує життя розробнику. Створення, витяг, оновлювання та видалення записів з врахуванням зв'язків між даними – все це забезпечено можливостями SQL. Саме тому система реляційних баз даних і наразі є першим вибором для будь-якої програми, що вимагає потужних транзакційних функцій, аналізу даних та складної звітності.

Згадавши про транзакції одразу розглянемо набір властивостей, які забезпечують їх надійну роботу та забезпечені реляційними БД [15][16]:

- Атомарність (Atomicity): розшифровується як «все або нічого». Транзакції часто містять у собі багато операцій, і якщо збоїла одна із них залишилась невиконаною, тоді транзакція вважається невдалою.
- Узгодженість (Consistency): гарантує, що база даних до і після виконання транзакції залишається в узгодженому, несуперечливому стані.

- Ізольованість (Isolation): гарантує, що при одночасній роботі кількох транзакцій вони не впливають на роботу одна одної. Згідно цієї властивості, у випадку, коли дві транзакції намагатимуться змінити одні й ті самі дані, одна з них буде відхилена або призупинена до завершення другої.
- Довговічність (Durability): гарантує, що після успішного виконання транзакції, дані будуть збережені та відновлені, навіть у випадку аварій у системі.

Проте всі вище описані переваги не стали на заваді відтиснення реляційних баз даних з ринку. Комп'ютерні технології розвивались, і в епоху соціальних мереж виникла потреба у обробці величезних об'ємів даних, з чим структуровані реляційні БД не змогли справитись. Тому у 1998 році завдяки Карло Строцці виникає термін NoSQL - як ім'я файлової бази даних, яка не включала використання мови запитів SQL. Сьогодні нереляційні бази даних – це ті бази даних, які не використовують SQL. Є декілька типів таких БД, які різняться за способом зберігання даних. Розглянемо детальніше документо-орієнтовані бази даних.

Такі БД зберігають дані у вигляді документів. Документи всередині документально-орієнтованої бази даних дещо схожі на записи у реляційних базах даних, але набагато гнучкіші, оскільки менш структуровані. Документи кодуються стандартними форматами, такими як XML, PDF, JSON тощо. Звертання відбувається за допомогою унікального ключа, що містить цей документ. Ці ключі можуть бути простим рядком з ідентифікатором або рядком, на який посилається URI.

Як і в реляційних БД, нереляційні також мають гарантувати послідовність та цілісність, особливо коли мова про критичні дані. Але на відміну від перших, не всі нереляційні підтримують правила ACID. BASE – більш лояльніша модель підходу до системи транзакцій[15][16]:

- Базова доступність (basic availability): гарантування завершення кожної транзакції (успішно або безуспішно).
- Гнучкий стан (soft state): дані можуть залишатись певний час неузгодженими.
- Можлива послідовність (eventual consistency): хоч і певний час дані можуть залишатись у неузгодженому стані, але через деякий час вони все одно стають узгодженими.

Зрозуміло, що дана модель не може бути використана в будь-яких системах, де критично необхідна узгодженість та систематизованість. Тому різні NoSQL бази даних можуть підтримувати ACID чи BASE.

Окрім моделі підтримки транзакцій, різниця між реляційними та нереляційними БД характеризується такими властивостями:

Таблиця 2.1 – Порівняльний аналіз реляційних та нереляційних БД

Критерій	Реляційна БД	Нереляційна БД
Мова запитів	SQL - це єдина мова обробки даних, яку використовують усі реляційні бази даних з незначними варіаціями у реалізації.	Кожна реалізація NoSQL має власну мову обробки даних.
Відкритий чи закритий код	Як з відкритим кодом, так і закриті платформи.	Здебільшого відкритий код.
Хмарні середовища	Не сильно пристосовані до хмарних середовищ.	Гарно інтегруються з хмарними середовищами. Окрім цього, деякі БД забезпечують не тільки

		локальний софт, а й одразу хмарний.
Послідовність	Сильна узгодженість із суворою схемою.	Більша доступність, проте низька узгодженість.
Захищеність	SQL має потужні механізми безпеки, які використовуються для захисту даних, що включають автентифікацію, авторизацію, шифрування, цілісність та аудит.	Обробляється проміжним програмним забезпеченням. Тому може бути вразливою до атак.

Отже, розглянувши характеристики реляційних та нереляційних баз даних, стає очевидним, що у кожної є як свої плюси, так і мінуси. Серед плюсів реляційних – узгодженість та структурованість даних, зручність навігації. Такі бази підходять під розробку систематизованих інформаційних систем, де чітко видно зв'язки та властивості. Наприклад, для зберігання та обробки інформації в магазинах, ведення обліку у організаціях. Найвживанішими СКБД є Oracle, MySQL, PostgreSQL. Як тільки потрібно оброблювати динамічні величезні об'єми даних – краще застосовувати нереляційні БД. Тому більшість соціальних мереж, ігор розроблені саме на даному типі баз даних. Відомими СКБД є MongoDB, CouchDB, Neo4j.

Дослідивши теоретичну частину, переходимо до опису практичної.

РОЗДІЛ 3: Опис реалізації програмного продукту

3.1 Аналіз технічного завдання

Веб-сайт розроблюється з урахуванням наступної групи користувачів: покупець та адміністратор. З урахуванням умов, що описані далі, буде розроблено інтерфейси під покупця та адміністратора (адмінпанель). Покупець має можливості перегляду наявних сеансів, формування корзини та бронювання білетів. Адміністратор має доступ до адмінпанелі, де наявний функціонал по керуванню сеансами та замовленнями.

Технічні вимоги користувачів до функціональності системи:

- **Адміністратор**

1. Створення, редагування та видалення фільмів та сеансів на них у адмінпанелі.
2. Перегляд історії замовлень та даних клієнта.

- **Покупець**

1. Перегляд актуальних фільмів та сеансів у кінотеатрі.
2. Перегляд інформації щодо обраного фільму.
3. Обрання бажаних квитків з врахуванням актуальної інформації щодо уже проданих місць.
4. Можливість бронювання за один раз квитків на один сеанс.
5. Отримання повідомлення на вказаний адрес електронної пошти після успішного бронювання квитків.

Специфікація вимог до даних:

- **Фільм**

Про фільм зберігається його унікальний номер, назва, додаткова інформація (у вигляді жанру та режисера), фото-афіша. Покупець може переглядати всю цю інформацію на слайдері на головній сторінці. У адміністратора є доступ редагувати всі наведені властивості, окрім унікального номеру. Також адміністратор може видаляти фільм з сайту.

- **Сеанс**

Кожен сеанс має унікальний номер, зберігає інформацію про дату та час проведення, ціну квитка на нього. Один сеанс доступний тільки на один певний фільм.

- **Замовлення**

Створюється одним клієнтом, та може містити у собі квитки на один сеанс. Містить будь-яку кількість квитків з обраними місцями. Визначені місця на один сеанс можуть бронюватись тільки один раз. Адміністратор має змогу переглядати номер замовлення, інформацію щодо нього (фільм, дані про сеанс та обрані місця, контактні дані), а також видаляти дане замовлення (у випадку форс-мажору, наприклад локдауну, чи скасування замовлення).

- **Покупець**

Має зберігатись інформація, яку покупець вказує при покупці білетів, а саме його ім'я, телефон, а також електронна пошта, на яку прийде повідомлення у випадку успішної покупки. Здійснювати реєстрацію для придбання квитків не потрібно. Через інтерфейс веб-сайту неможливо повернути квитки. Алгоритм створення замовлення для покупця:

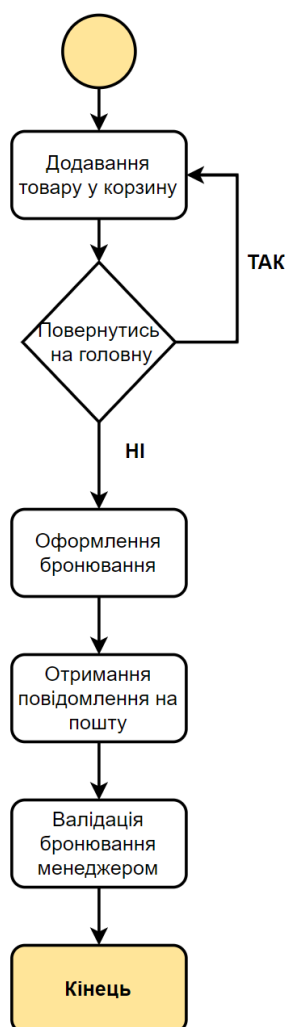


Рисунок 3.1 – Алгоритм створення замовлення

Веб-сайт повинен працювати незалежно від операційної системи користувача або ж обраного браузеру. Проте зауважимо, що задля успішного користування та здійснення замовлення необхідна наявність підключення до Інтернет мережі.

3.2 Обґрунтування обраних засобів розробки

У зв'язку з тим, що робота опирається на дві різних бази даних, проект поділився на два. Було створено два повністю однакових веб-сайти з ідентичним функціоналом, проте перший працював з СКБД MySQL, а інший з MongoDB. Детальніше їх розглянуто наприкінці поточного розділу. Всі інші обрані інструменти ідентичні для кожного з проекту, тому розглянемо їх у розрізі одного проекту.

Отже, проектування структури веб-сайту ґрунтується на принципі клієнт-серверної архітектури. Це принцип на якому побудовано більшість веб-сайтів та мережевих застосунків, і він визначає загальні правила взаємодії в мережі. Основними компонентами є клієнт, сервер та сама мережа. По суті клієнт виступає замовником певних послуг, а сервер дані послуги постачає, і відбувається ця комунікація завдяки мережі.

Головний інструмент цієї взаємодії - HTTP-протокол: клієнт(браузер) надсилає http-запит на сервер, а сервер виконує запит та надсилає відповідь клієнту. Для отримання даних із сервера використовуємо GET-запити. А POST надсилає дані серверу, наприклад, додати чи оновити дані. У коді виглядає наступним чином:

```
$price = $_POST['price'];
```

Розглянувши принцип побудови продукту розглянемо конкретні інструменти, які були використані у його реалізації.

Серверна сторона веб-сайту розроблена завдяки мові PHP – одного з найпопулярніших та найуживаніших інструментів для веб-розробки. Незважаючи на велику кількість аналогів PHP не здає свої позиції і досі. Станом на квітень 2021 року відповідно до Індексу популярності мов програмування (PYPL)[17], PHP займає шосте місце у світі. Два найвідоміші приклади використання PHP – соціальна мережа Facebook та система управління вмістом WordPress. Незважаючи на стрімкий розвиток технологій, ці гіганти й досі незмінні у своєму початковому виборі мови розробки.

Така популярність PHP пояснюється значним переліком переваг. Перше та найголовніше, що за багато років існування цієї мови та широкого використання серед розробників, утворилось багато документації та довідників, що є надважливим в етапах вивчення будь-якої мови програмування. Це забезпечує широкий ринок розробників, що утворює конкуренцію, та забезпечує меншу вартість розробки.

Окрім цього значною перевагою PHP є кросплатформеність: мова незалежна від браузерів (сценарії PHP повністю компілюються на стороні сервера перед відправкою клієнту), сумісна з будь-якою операційною системою та не розроблена під певний сервер, що дозволяє саме розробнику вирішувати, що буде ефективніше для нього.

Крім того, PHP достатньо гнучка, що дає змогу ефективно суміщатись з іншими мовами програмування. Є змога імплементації PHP-коду безпосередньо в HTML-код сторінок, які будуть оброблюватись PHP-інтерпретатором. Тому, враховуючи всі описані вище переваги, було обрано мову PHP версії 7.4.

За зовнішній вигляд веб-сайту відповідали HTML, CSS та JavaScript. Разом ці інструменти надають широкий спектр можливостей, аби сторінки були максимально привабливими для користувача, були цікавими та інтерактивними.

HTML [18] – це мова розмітки гіпертексту (від англ. HyperText Markup Language). Саме завдяки HTML можна визначити значення та структуру веб-вмісту: створити та структурувати заголовки, розділи, абзаци, посилання, тощо. При відкритті сторінки в браузері, останній зчитує HTML-код та виявляє з нього спеціальні символи – теги та атрибути, обробляє їх, та створює відповідні елементи. Серед плюсів HTML гарна інтеграція з серверними мовами, такими як обрана PHP або ж Node.js. Дана мова розмітки зчитується усіма браузерами, є безкоштовною та підтримуваною розробниками. До прикладу, оновлення HTML5, що вийшло у 2008 році, надає інструменти для відтворення аудіо- та відеофайлів без додаткових розширень.

CSS або каскадна таблиця стилів [19] - це мова таблиці стилів для HTML-документу. Іншими словами, вбудувавши файл CSS у документ HTML, можна стилізувати дизайн, макет, шрифти, кольори вмісту веб-сайту. CSS також є інструментом стилізації XML-документів будь-якого вигляду або SVG-документів. Щоб підключити до HTML-сторінки стилі CSS потрібно: або використати тег `link`; або використати тег `style`; або ж атрибут цього тегу; якщо CSS-файл знаходиться окремо та має розширення `.css` – імпорт відбувається

через директиву `@import`. Тому завдяки зручній структурі, великій кількості селекторів CSS тривалий час є запорукою гарного вигляду веб-сторінок.

Проте ні HTML, ні CSS не зможуть дати бажаної інтерактивності на веб-сторінці. Меню-бари, слайдери, інтерактивні зображення, спливаючі вікна і багато іншого – створюється завдяки об'єктно-орієнтованій мові програмування JavaScript, яка надає можливості для формування динамічно оновлюваного контенту веб-сторінки без перезавантажування. JavaScript домінує серед сучасних мов сценаріїв на стороні клієнта в Інтернеті[20] – біля 97% веб-сайтів розроблені з його допомогою.

Обґрунтувавши засоби розробки перейдемо до обраних СКБД. Для роботи з реляційною БД було обрано MySQL. Для керування нереляційною БД було обрано MongoDB.

MySQL - це система управління реляційними базами даних з відкритим кодом. Оскільки мова йде про реляційні БД, то очікувано, що MySQL зберігає дані в таблицях, що складаються з рядків і стовпців. Користувачі можуть визначати, обробляти, контролювати та запитувати дані за допомогою SQL. MySQL тривалий час залишається одним з найпопулярніших інструментів з керування реляційними БД завдяки зручності інтерфейсу та в цілому роботи, підтримки багатьох мов програмування, швидкості та надійності. Саме тому було обрано це програмне забезпечення.

MongoDB – одна з популярних документоорієнтованих систем керування нереляційними базами даних. Формат зберігання документів у MongoDB [21] - BSON (двійковий JSON). Колекція складається з документів BSON, які у свою чергу містять упорядкований перелік елементів, що складається з назви поля, типу та значення. MongoDB надає унікальний ідентифікатор для кожного документа, що являє собою 12-байтове шістнадцяткове число. Це все, а також масштабованість, легкість зв'язків і власні запити, схожі на SQL, роблять СКБД MongoDB привабливою серед розробників.

Отже, розглянувши інструменти розробки, оглянемо реляційні та нереляційні моделі даних розроблені під веб-сайт.

3.3 Опис моделі даних

Реляційна модель

База даних складається з 4 реляцій, зв'язаних між собою.

Таблиця «Фільми» складається з чотирьох атрибутів: порядкового номеру, назви фільму, жанру фільму та режисера. Всі вони є обов'язковими.

Таблиця 3.1 – Таблиця «Фільми»

Films				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	INT(11)	NOT NULL	Атрибут простий, обов'язковий «Порядковий номер»
	name	VARCHAR(45)	NOT NULL	Атрибут простий, обов'язковий «Назва фільму»
	genre	VARCHAR(20)	NOT NULL	Атрибут простий, обов'язковий «Жанр фільму»
	director	VARCHAR(25)	NOT NULL	Атрибут простий, обов'язковий «Режисер фільму»

Таблиця «Сеанс» зберігає інформацію про дату сеансу, час та ціну квитка. Ідентифікує кожен сеанс унікальний номер. Таблиця має зв'язок з таблицею «Фільми».

Таблиця 3.2 – Таблиця «Сеанс»

Session				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	id	INT(11)	NOT NULL	Атрибут простий, обов'язковий «Номер сеансу»
	date	DATE	NOT NULL	Атрибут простий, обов'язковий «Дата сеансу»
	time	Можливі значення: «12.00», «13.00», «14.00», «18.00»	NOT NULL	Атрибут простий, обов'язковий «Час сеансу»
	price	FLOAT	NOT NULL	Атрибут простий, обов'язковий «Ціна квитка»
FK	film_id	INT(11)	NOT NULL	Зв'язок з таблицею Films:id Зв'язок обов'язковий з боку Films

Таблиця «Замовлення» містить в собі інформацію про місце із замовлення. В базі буде зберігатись окремо кожне місце з замовлення. Таблиця має зв'язок з таблицями «Сеанс» і «Покупець».

Таблиця 3.3 – Таблиця «Замовлення»

Order				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення

PK	id	INT(11)	NOT NULL	Атрибут простий, обов'язковий «Номер замовлення»
	place	DATE	NOT NULL	Атрибут простий, обов'язковий «Місце»
FK	session_id	INT(11)	NOT NULL	Зв'язок з таблицею Session:id Зв'язок необов'язковий з боку Session
FK	mail	VARCHAR(30)	NOT NULL	Зв'язок з таблицею Client:mail Зв'язок обов'язковий з боку Client

Таблиця «Покупець» ведеться для адміністратора та подальших маркетингових операцій (наприклад, масова розсилка реклами кінотеатру на електронні пошти клієнтів. Унікальним ідентифікатором є сама електронна пошта, також в базі зберігається телефон та ім'я покупця.

Таблиця 3.4 – Таблиця «Покупець»

Client				
Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	Пояснення
PK	mail	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Електронна пошта»
	name	VARCHAR(30)	NOT NULL	Атрибут простий, обов'язковий «Ім'я»
	phone	VARCHAR(12)	NOT NULL	Атрибут простий, обов'язковий «Телефон»

Нереляційна модель

Дані зберігаються у двох колекціях: «films» та «orders». Колекція «films» містить наступні поля: ключ за замовчуванням id, наданий самою mongodb, назва фільму, дата, час сеансів, режисер, ціна квитка. Колекція «orders» містить дані про місце, ім'я, електронну пошту та телефон замовника. Також зберігається посилання на об'єкт з колекції фільму, на який зроблено замовлення.

На етапі проектування було зрозуміло, що під даний проект більше підходить саме реляційна БД, оскільки дані у нас чітко структуровані і на горизонті масштабування не передбачається. Проте на етапі розробки варто пересвідчитись у цих судженнях.

3.4 Опис різниці у розробці

Робота проектів починається із створення сесії та підключення до БД. У проекті з реляційною БД це виглядає наступним чином:

```
//Початок сесії користувача
session_start();
// Задаємо параметри входу
$db_host = 'localhost';
$db_user = 'root';
$db_pass = 'root';
$db_database = 'films';

$link = mysqli_connect($db_host, $db_user, $db_pass);
```

Підключення до MongoDB виглядає так:

```
session_start();
$client = new MongoClient();

//створення колекції підключення->назва бази->назва документа
$collection = $client->kursova->cinema;

//дані з колекції
$res = $collection->find()->toArray();
```

При створенні нового фільму інформація в БД заноситься завдяки POST запиту.
Для MongoDB:

```
$client = new \MongoDB\Client();
//колекція підключення->назва бази->назва документа
$collection = $client->kursova->cinema;

//пуш в колекцію методом InsertOne
    $collection->InsertOne([
        'name'=>$name,
        'genre'=>$genre,
        'date'=>$date,
        'time'=>array(
            $time1,$time2,$time3
        ),
        'director'=>$director,
        'price'=>$price,
        'photo'=>$photo
    ]);
```

На відміну від MongoDB, яка вносить інформацію завдяки методам, MySQL використовує запити SQL:

```
//додавання у колекцію
    $collection = mysqli_query($link,"INSERT INTO `products`(`name`, `price`, `genre`, `date`, `time`, `director`, `image`) VALUES (
        '$name',
        '$price',
        '$genre',
        '$date',
        '$times[0]',
        '$director',
        '$photo'
    )");
```

Видалення та оновлення інформації у MongoDB також завдяки методам, а не запитам як у MYSQL. Порівняємо:

```
$collection->deleteOne(['name'=>$name]);
$collection->updateOne($oldDocs,['$set'=>$newDoc,['w'=>1]);
```

SQL:

```
$collection = mysqli_query($link, "DELETE FROM `products` WHERE `id`='$id'");
$collection = mysqli_query($link,"UPDATE `products` SET
    name='$name',
```

```
price='$price');
```

Отже, головна принципова різниця у коді між проектом з реляційною та нереляційною БД – це робота з даними. Перша задля цього використовує мову SQL, інша підключені бібліотечні методи. Розглянувши головні відмінності, переходимо до тестування програми

3.5 Тестування програми і результати її виконання

Оскільки дизайн та функціонал проектів на реляційній та нереляційній БД ідентичні, тому тестування буде здійснюватися на першому проекті. Головною сторінкою на веб-сайті є сторінка зі списком актуальних фільмів, також бачимо знак кошика в правому куті, при нажаті перемістимось до нього:

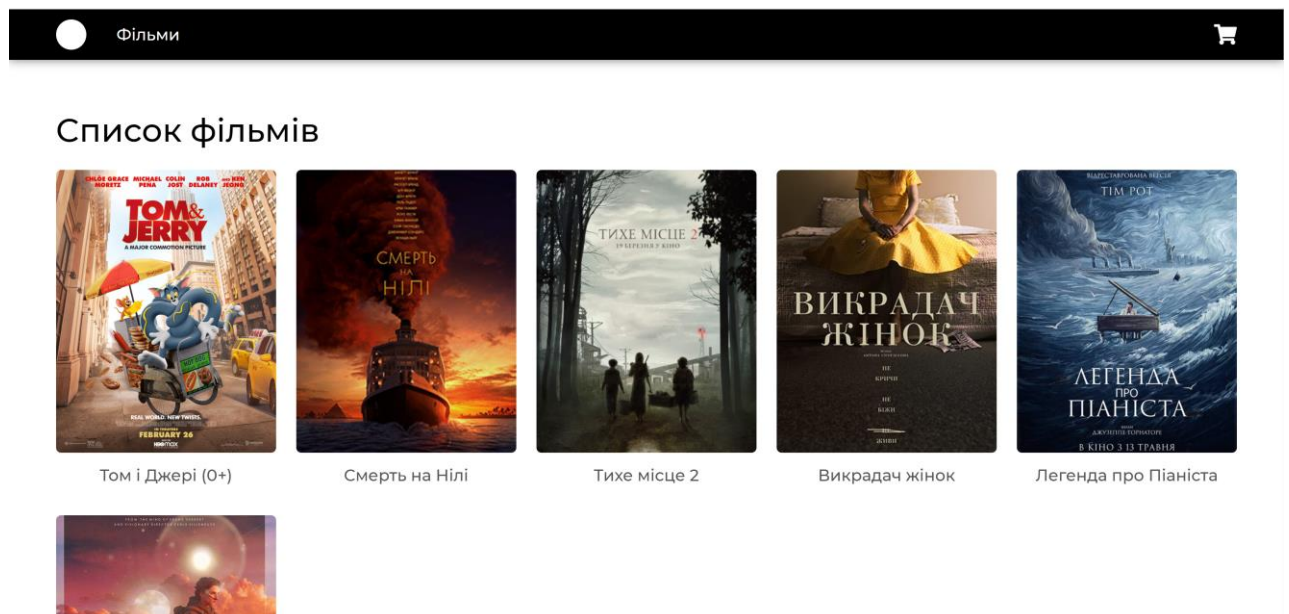


Рисунок 3.2 – Головна сторінка сайту

За допомогою інструментів JavaScript реалізовано слайдер, який при наведенні курсору на афішу, надає інформацію про обраний фільм та сеанси у конкретний день:

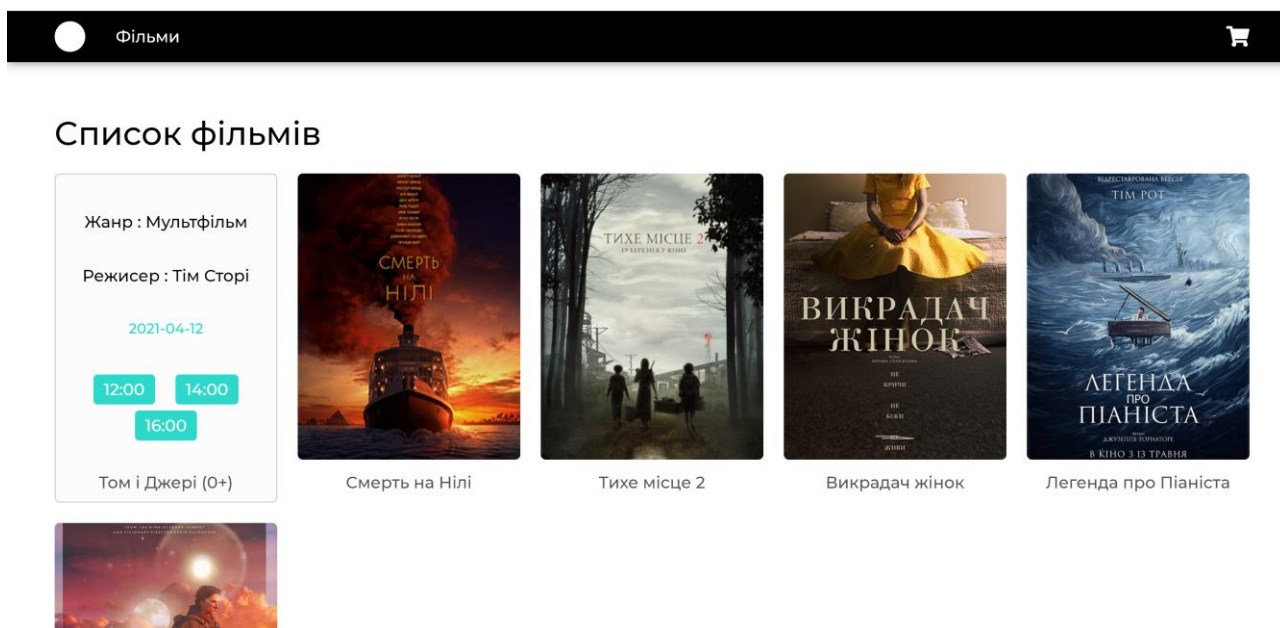


Рисунок 3.3 – Слайдер з інформацією про конкретний фільм

Обравши бажаний фільм та сеанс, користувач переходить на сторінку з обранням місць, де зручно промальовано кожне місце:

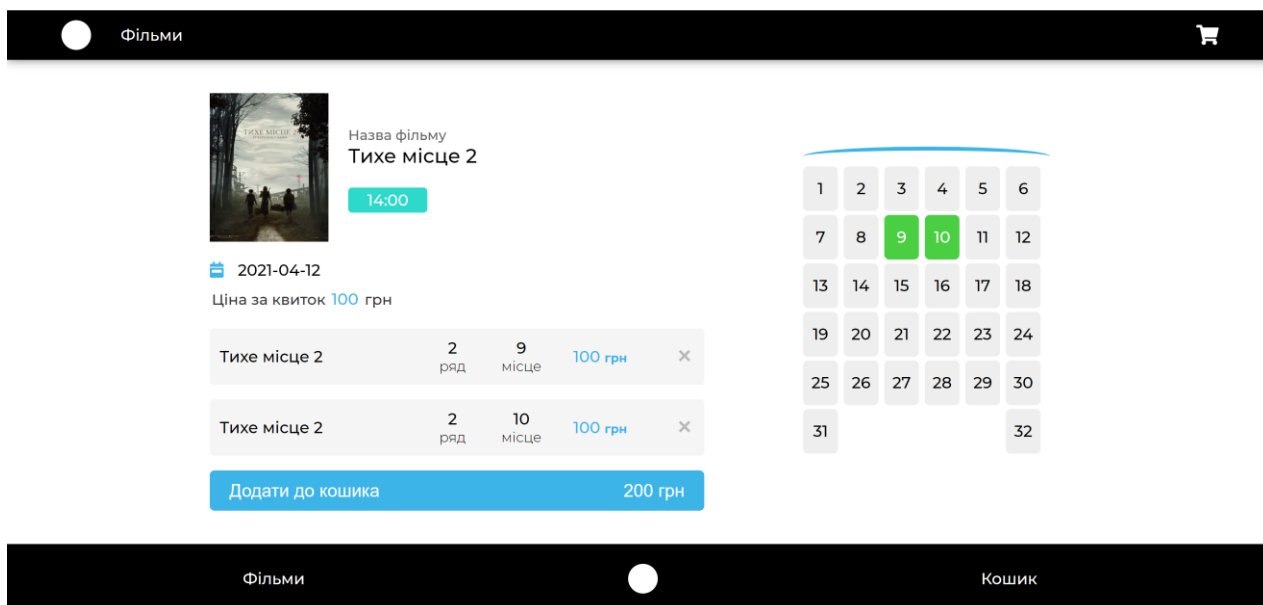


Рисунок 3.4 – Сторінка обраного сеансу

Обравши бажані квитки, користувач може здійснити замовлення, перейшовши до кошику. Можна видалити квиток у разі помилки, або ж перейти до форми оформлення. Вказавши усі дані, користувач отримує повідомлення, що його замовлення успішне:

Рисунок 3.5 – Сторінка кошику

Пройшовши шлях користувача, можемо перейти до тестування адмінпанелі. Одразу потрапляємо на сторінку з переліком активних фільмів, дат та сеансів на них. Є опція додавання нового фільму, редагування інформації про фільм, та його видалення з сайту:

Фільми Замовлення До сайту							
Новий фільм							
Назва	Жанр	Дата	Час сеансу	Режисер	Ціна	Редагування	Видалення
Том і Джері (0+)	Мультфільм	2021-04-12	12:00,14:00,16:00	Тім Сторі	125	Редагувати	Видалити
Смерть на Нілі	Детектив	2021-04-13	13:00,15:00,16:00	Кеннет Брана	130	Редагувати	Видалити
Тихе місце 2	жахи	2021-04-12	12:00,14:00,15:00	Джон Красінскі	100	Редагувати	Видалити
Викрадач жінок	жахи	2021-04-23	12:00,13:00,14:00	Антон Сігурдссон	75	Редагувати	Видалити
Легенда про Піаніста	мелодрама	2021-04-12	12:00,14:00,16:00	Джузеппе Торнаторе	90	Редагувати	Видалити
Дюна	пригоди	2021-04-20	12:00,14:00,16:00	Дені Вільньов	150	Редагувати	Видалити

Рисунок 3.6 – Панель адміністратора

Натиснувши кнопку «Новий фільм» переходимо на сторінку з потрібними полями. Після їх заповнення, натискаємо кнопку «submit»:

Фільми
Замовлення
До сайту

Час сеансів

Вибрати файли

Файл не вибрано

Рисунок 3.7 – Створення нового фільму

Наступний функціонал в адмінпанелі – це перегляд замовлень, адміністратор може переглядати деталі по кліку на посилання «Інформація», та видаляти замовлення зі списку у разі його анулювання:

Фільми
Замовлення
До сайту

СТОРІНКА ЗАМОВЛЕНЬ

Номер замовлення	Інформація	Дія
27	Інформація	Видалення
28	Інформація	Видалення
29	Інформація	Видалення
30	Інформація	Видалення
31	Інформація	Видалення
32	Інформація	Видалення
33	Інформація	Видалення
34	Інформація	Видалення
35	Інформація	Видалення
36	Інформація	Видалення
37	Інформація	Видалення

Рисунок 3.8 – Перегляд списку замовлень

Перейшовши на сторінку з деталями замовлення отримуємо його дані, а також дані клієнта для подальшої комунікації з адміністратором:

Замовлення Фільми До сайту				
Замовлення # 21				
Клієнт: Вікторія Мацюк				
Пошта: vikama@ukr.net				
Телефон: 0952874519				
Назва	Дата	Час	Місце	Ціна
Тихе місце 2	2021-04-21	14:00	15	130
Тихе місце 2	2021-04-21	14:00	21	130

Рисунок 3.9 – Перегляд деталей конкретного замовлення

Отже, тестування функціоналу пройдене успішно, продемонстровано основні можливості як користувача, так і адміністратора. Веб-сайт працює за стандартними вимогами до веб-застосунків.

Висновок

По закінченню роботи всі поставлені задачі були виконані. Перш за все, було обрано предметну область, а саме кінотеатральний бізнес. Проаналізувавши даний ринок, було виявлено, що не дивлячись на високий рівень технологічного прогресу, багато з діючих кінотеатрів досі не мають власних веб-сайтів. Тому була поставлена задача створити продукт, що задовільнить чітко описані умови. Задля визначення цих умов було досліджено уже існуючі продукти, описано їх переваги та недоліки. У проекті вони майже всі враховані та реалізовані.

Окрім цього, було досліджено інформаційні системи, основні компоненти та типи. У результаті отримано висновок, що кожен тип інформаційних систем варто використовувати під певний спектор задач, і об'єднавши їх, отримаємо ідеальну систему. Також розглянуто та використано в проекті основні типи баз даних: реляційна та нереляційна. Були проведені між ними паралелі у роботі, завдяки чому сформовано висновок, що для обрання необхідної бази необхідно чітко розуміти потреби та вимоги програми, враховувати всі проблеми, з якими може зіткнутися програма не на горизонті. У конкретному випадку розробленого веб-сайту зручніше використовувати саме реляційну модель, оскільки маємо чітку структуру даних та їх взаємозв'язків, та розуміння, що динамічне масштабування не є критично важливим.

Розроблений веб-сайт є придатним до використання не тільки у навчальних цілях, але й комерційних. В планах покращення та розширення функціоналу можемо зазначити наступні дії: додавання фільтрів (по датам, по фільмам, і тд), створення та імплементація кабінету користувача задля збільшення привабливості серед клієнтів, підключення платіжних систем для можливості онлайн-розрахунку. Також, варто додати можливість для адміністратора переглядати статистику сеансів, витяг інформації про продані квитки у вигляді Excel таблиць. Проект розроблено з використанням локального веб-серверу Open Server Panel, а отже до планів вдосконалення можемо також додати хостинг веб-сайту на хмарі.

Таким чином основними результатами роботи є аналіз та дослідження інформаційних систем та їх типів, дослідження реляційних та нереляційних баз даних, порівняння їх переваг і функцій, дослідження сучасних мов веб-програмування; та розробка зручного та інтуїтивно зрозумілого веб-сайту з врахуванням усіх набутих знань.

Список використаної літератури

1. Что нам стоит кинозал построить: обзор кинотеатрального рынка Украины [Електронний ресурс]. 17.09.2019 Автор Анастасія Рахманіна, Маріанна Коцан
<https://mbr.com.ua/ru/news/analytics/253-chogo-nam-var-to-kinozal-pobuduvati-oglyad-kinoteatralnogo-ri>
2. Как оживило украинское кино [Електронний ресурс]. 07.06.2018 Автор Діана Куришко
<https://www.bbc.com/ukrainian/features-russian-44394631>
3. Автоматизация продаж в рознице и онлайн – секреты успеха [Електронний ресурс]. 25.12.2019 Автор Envybox блог
<https://envybox.io/blog/avtomatizacija-prodazh/>
4. Веб-сайт кінотеатру «Планета» [Електронний ресурс]. Дата використання 01.04.2021
<http://www.kino-planeta.com/>
5. Планета (кінотеатр, Хмельницький) [Електронний ресурс]. Дата використання 01.04.2021
[https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD%D0%B5%D1%82%D0%B0_\(%D0%BA%D1%96%D0%BD%D0%BE%D1%82%D0%B5%D0%B0%D1%82%D1%80,_%D0%A5%D0%BC%D0%B5%D0%BB%D1%8C%D0%BD%D0%B8%D1%86%D1%8C%D0%BA%D0%B8%D0%B9\)](https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD%D0%B5%D1%82%D0%B0_(%D0%BA%D1%96%D0%BD%D0%BE%D1%82%D0%B5%D0%B0%D1%82%D1%80,_%D0%A5%D0%BC%D0%B5%D0%BB%D1%8C%D0%BD%D0%B8%D1%86%D1%8C%D0%BA%D0%B8%D0%B9))
6. Продовження заборони російських соцмереж в Україні [Електронний ресурс]. 15.05.2020 Автор Радіо Свобода
<https://www.radiosvoboda.org/a/news-sotsmerezhi-rosiyski-zaborona-ukaz/30613348.html>
7. Веб-сайт кінодрому О.Довженка [Електронний ресурс]. Дата використання 01.04.2021
<https://www.kinodrom.kiev.ua/>

8. Веб-сайт мережі кінотеатрів «Планета-кіно» [Електронний ресурс]. Дата використання 01.04.2021
<https://planetakino.ua/>
9. Information Systems and Future [Електронний ресурс]. 03.10.2019 Автор Witanworld
<https://witanworld.com/article/2019/10/03/information-systems-and-future/>
10. Components Of Information System [Електронний ресурс]. Дата використання 04.04.2021
<https://www.geeksforgeeks.org/components-of-information-system/>
11. Relationships between Information Systems and Organizations [Електронний ресурс]. Дата використання 04.04.2021
https://uotechnology.edu.iq/ce/Lectures/SarmadFuad-MIS/MIS_Lecture_7.pdf
12. What is MIS? Introduction & Definition [Електронний ресурс]. Дата використання 04.04.2021
<https://www.guru99.com/mis-definition.html>
13. Marketing Information System [Електронний ресурс]. Дата використання 04.04.2021
<https://businessjargons.com/marketing-information-system.html>
14. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Авторы Томас Конолли, Каролін Бегг, 2003
15. Relational vs. NoSQL Databases: A Survey [Електронний ресурс]. 03.05.2014 Автор Mohamed A. Mohamed, Obay G. Altrafi, Mohammed O. Ismail
https://www.researchgate.net/profile/Mohamed-Mohamed-314/publication/263272704_Relational_Vs_NoSQL_databases_A_survey/links/00b7d53a495312ad22000000/Relational-Vs-NoSQL-databases-A-survey.pdf
16. A Comparative Study of NoSQL and Relational Database [Електронний ресурс]. Дата використання 04.04.2021 Автор Douglas Kunda, Hazael Phiri
<https://ictjournal.iciict.org.zm/index.php/zictjournal/article/view/8/3>

PYPL PopularitY of Programming Language [Електронний ресурс]. Дата використання 04.04.2021

<https://pypl.github.io/PYPL.html>

18. What is HTML? The Basics of Hypertext Markup Language Explained [Електронний ресурс]. 25.11.2019 Автор Domantas G

<https://www.hostinger.com/tutorials/what-is-html>

19. CSS Cheat Sheet – A Complete Guide for Beginners and Professionals [Електронний ресурс]. 09.03.2021 Автор Domantas G

<https://www.hostinger.com/tutorials/css-cheat-sheet>

20. JavaScript [Електронний ресурс]. Дата використання 04.04.2021

<https://en.wikipedia.org/wiki/JavaScript>

21. MongoDB — Краткое руководство [Електронний ресурс]. 14.04.2019 Автор CoderLessons.com

<https://coderlessons.com/tutorials/bazy-dannykh/uchitsia-mongodb/mongodb-kratkoe-rukovodstvo>