

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки» _____,

освітньо-професійної програми «Інформаційні технології проектування» _____

на тему: «Web-система підтримки діяльності розважального комплексу «Трибуна»

Здобувачки групи ІТ.мз-21с Шевелєва Ірина Ігорівна
(шифр групи) (прізвище, ім'я, по батькові)

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

(підпис)

Ірина ШЕВЕЛЄВА
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник старший викладач кафедри ІТ, к.т.н., Ольга БОЙКО _____
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2023

Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Шевелева Ірина Ігорівна

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи Web-система підтримки діяльності розважального комплексу "Трибуна"

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом кваліфікаційної роботи «11» грудня 2023 р.

3 Вхідні дані до кваліфікаційної роботи дані про розважальний комплекс «Трибуна», дані про послуги та заходи у розважальному комплексі «Трибуна», дані про користувачів

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, моделювання та проектування, розробка Web-системи підтримки діяльності розважального комплексу "Трибуна"

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації) тема, актуальність, постановка задачі, огляд програмних продуктів-аналогів, таблиця порівняння аналогів, функціональні вимоги до Web-системи, інструменти реалізації, структурно-функціональне моделювання, моделювання бази даних, моделювання варіантів використання, практична реалізація, тестування Web-системи, демонстрація результатів розробки, висновки

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та постановка задачі дослідження		
2	Планування робіт		
3	Вибір методів дослідження		
4	Моделювання/проектування		
5	Практична реалізація результатів дослідження		
6	Написання розділів пояснювальної записки та надання керівнику на перевірку		
7	Підготовка заключної версії кваліфікаційної роботи, перевірка на плагіат		
8	Подача до ЕК комплекту документів до захисту		

Магістрант _____

Ірина ШЕВЕЛЄВА

Керівник роботи _____

к.т.н., Ольга БОЙКО

АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Web-система підтримки діяльності розважального комплексу "Трибуна"».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 30 найменувань, додатків. Загальний обсяг роботи – 68 сторінок, у тому числі 47 сторінок основного тексту, 3 сторінки списку використаних джерел, 17 сторінок додатків.

Актуальність роботи полягає в розробці та впровадженню Web-система підтримки діяльності розважального комплексу "Трибуна".

Мета роботи: створення Web-системи, базованою на фреймворку Laravel, що автоматизує покупки квитків на заходи та бронювання спорядження для відвідувачів, а також забезпечує зручний інтерфейс для управління і підтримки діяльності розважального комплексу.

Обсяг анотації: 1 сторінка.

Ключові слова: Web-система, Розважальний комплекс, CSS, PHP, MySQL, PHPMyAdmin, Laravel

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Аналіз аналогічних розробок.....	10
1.2 Висновок з аналізу	15
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	18
2.1 Мета та задачі	18
2.2 Вибір засобів реалізації	19
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-СИСТЕМИ.....	21
3.1. Розробка структурно-функціональної моделі системи	21
3.2. Моделювання варіантів використання системи.....	24
3.3. Проектування моделі бази даних	24
3.3. Створення дизайн-макету web-системи.....	32
4 РОЗРОБКА WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ	34
4.1. Розробка web-системи	34
4.3. Тестування web-системи	43
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТОК А	52
ДОДАТОК Б.....	64

ВСТУП

У сучасному світі, що розвивається з нестримною швидкістю, розважальні комплекси стають неодмінною складовою соціокультурного простору. Це місце, де люди можуть відпочити і розважитись, як великою компанією чи сім'єю, так і на одинці. Зазвичай розважальні комплекси включають в себе багато різних видів діяльності, наприклад активний вид відпочинку, кафе, атракціони, сцену та дитячу зону. У такому середовищі стає важко як відвідувачам, які не знайомі із закладом, орієнтуватися на місці, так і працівникам, контролювати бізнес-процеси та задовольняти потреби користувача у інформації. З плином часу і стрімким технологічним прогресом, виникає необхідність вдосконалення процесів управління та ефективного функціонування розважальних закладів. Саме у цьому контексті стає надзвичайно актуальною тема дослідження, яка присвячена розробці та впровадженню в життя web-системи підтримки діяльності розважального комплексу "Трибуна".

Однією з ключових переваг web-систем для розважальних комплексів є їх здатність забезпечити інтерактивність та персоналізацію вражень. За допомогою web-платформ можна створити індивідуальні програми розваг, враховуючи інтереси кожного відвідувача. Це дозволяє розважальним комплексам ефективно привертати нових клієнтів та забезпечувати повторні візити, враховуючи потреби різних аудиторій.

Додатково, web-системи дозволяють розважальним комплексам ефективно взаємодіяти з клієнтами через різноманітні канали, такі як онлайн-бронювання, програми лояльності та спеціальні пропозиції. Це сприяє збільшенню залученості та вірогідності повторних відвідувань, роблячи розважальні комплекси більш конкурентоспроможними в сучасному ринковому середовищі.

Крім того, web-системи можуть значно полегшити управління ресурсами та організацію подій у розважальних комплексах. Вони дозволяють автоматизувати процеси бронювання, контролю доступу, а також ведення обліку та аналітики. Це

робить роботу персоналу ефективнішою та сприяє оптимальному використанню ресурсів

Об'єктом дослідження виступає розважальний комплекс "Трибуна" в цілому, з його структурними підрозділами та клієнтською базою. Дослідження спрямоване на аналіз розробки та моделювання web-системи, яка забезпечить ефективну взаємодію всіх суб'єктів управління.

Предметом є процес розробки web-системи підтримки діяльності розважального комплексу «Трибуна», яка спрямована на оптимізацію та автоматизацію ключових процесів управління та обслуговування розважального комплексу. Враховуючи високу конкуренцію у галузі розваг, важливо створити інтегроване рішення, що враховує специфіку діяльності "Трибуна" та надає засоби для ефективного моніторингу та взаємодії всіх компонентів комплексу.

Основні задачі дипломної роботи:

- Проведення аналізу актуальних аспектів обраної предметної області з метою визначення сучасних тенденцій, викликів та можливостей, що можуть бути використані у контексті досліджуваного дипломного проекту.
- Проектування та моделювання об'єкту дослідження з метою створення концептуальної та технічної основи для подальшої імплементації.
- Розробка дизайн-макету системи для візуалізації структури web-системи та її представлення.
- Розробка та тестування web-системи підтримки діяльності для розважального комплексу.

Таким чином, дана кваліфікаційна робота визначає та обґрунтовує актуальність впровадження web-системи підтримки діяльності розважального комплексу «Трибуна».

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Актуальність web-систем для розважальних комплексів визначається не лише можливістю індивідуалізації та взаємодії з клієнтами, але й суттєвим вдосконаленням управлінських та організаційних процесів у цьому секторі. У світлі швидкої цифрової трансформації відвідувачі розважальних комплексів сподіваються на доступ до інноваційних технологій для забезпечення більш зручного та цікавого дозвілля.

Web-системи дозволяють розважальним комплексам активно пристосовуватися до змін у попиті та трендах споживачів. Можливість вести онлайн-продаж квитків, резервування столиків та послуг, а також створення індивідуальних пакетів для відвідувачів робить web-системи невід'ємною частиною стратегії маркетингу та обслуговування клієнтів.

Окрім того, з урахуванням активного використання мобільних пристроїв, важливим є також розвиток мобільних додатків та web-версій, які надають відвідувачам миттєвий доступ до інформації про події, акції та можливості бронювання, забезпечуючи високий рівень зручності та мобільності.

З врахуванням динамічного характеру ринку розваг, web-системи дозволяють розважальним комплексам ефективно взаємодіяти з аудиторією через соціальні мережі, інтернет-рекламу та інші цифрові канали. Це допомагає не лише залучати нових клієнтів, але й підтримувати сталість інтересу та взаємодії з постійними відвідувачами.

Загалом, web-системи для розважальних комплексів визначаються своєрідним об'єднанням технологічних можливостей та прагматичного використання для створення неповторних та захоплюючих вражень для клієнтів, а також оптимізації управлінських процесів у сфері розваг. У цьому контексті вони стають ключовим інструментом у конкурентній боротьбі та забезпечують стійке позиціонування розважальних комплексів в сучасному світі розваг.

Сформуємо критерії для порівняння web-систем.

1. Дизайн:

– дизайн web-системи повинен бути привабливим, сучасним і відповідати концепції розважального комплексу. Зручність і легкість сприйняття інформації також грають важливу роль.

– важливо, щоб дизайн був адаптивним до різних типів пристроїв (планшети, мобільні телефони, комп'ютери), забезпечуючи ефективний користувацький досвід.

2. Структурна навігація:

– система повинна мати логічну структуру навігації, щоб користувачі могли легко знаходити потрібну інформацію та функції.

– навігація повинна бути інтуїтивно зрозумілою, з мінімальною кількістю клацань для досягнення конкретних розділів чи функцій.

3. Фотоматеріали:

– важливо, щоб фотоматеріали були якісними та репрезентативними, передаючи атмосферу та архітектурні особливості розважального комплексу.

– система повинна надавати можливість зручного перегляду фотографій та відеоматеріалів.

4. Онлайн оплата:

– система повинна забезпечувати високий рівень безпеки для онлайн транзакцій та зручність введення платіжних даних.

– наявність різних платіжних опцій для задоволення різних вподобань клієнтів.

5. Інформація про акції:

– Актуальність: інформація про акції повинна бути актуальною та швидко оновлюватися для заохочення клієнтів взяти участь.

– Ясність та деталізація: Інформація повинна бути ясною та деталізованою, щоб користувачі могли легко розуміти умови та переваги.

6. Інтерактивна мапа:

– Повнота: інтерактивна мапа повинна надавати повну інформацію про всі розділи та сервіси розважального комплексу.

– Легкість використання: важливо, щоб користувачі могли легко маніпулювати мапою та швидко знаходити необхідні об'єкти.

7. Фільтрування:

– Гнучкість: система повинна надавати гнучкі фільтри для швидкого і точного пошуку інформації чи послуг.

– Можливість збереження налаштувань: користувачі повинні мати можливість зберігати свої фільтри для майбутніх візитів.

8. Можливість залишати відгуки:

– Зручність: простий та зручний інтерфейс для залишення відгуків.

– Відкритість: система повинна бути відкритою до відгуків різних типів, забезпечуючи збалансований погляд на досвід відвідувачів.

Оцінювання web-систем за цими критеріями дозволить розважальним комплексам забезпечити високий стандарт якості та забезпечити задоволення та зручність для своїх клієнтів.

1.1 Аналіз аналогічних розробок

Web-система «Буковель». Web-система являє собою інформаційну сторінку, де користувачі мають змогу переглянути інформацію про туристичне місце (рис.1.1).

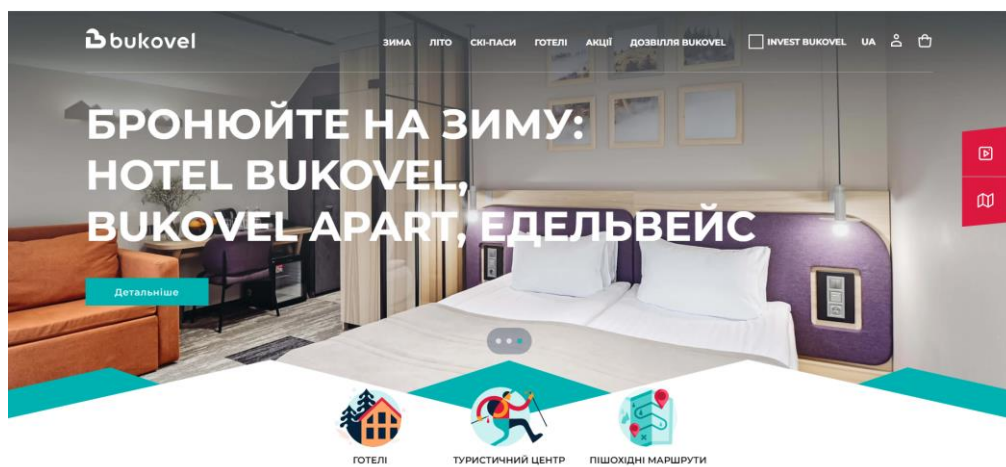


Рисунок 1.1 – Головна сторінка web-системи

Джерело: побудовано автором (знімок з екрану)

Web-система має можливість реєстрації користувачів. Вона призначена для того, щоб користувачі могли переглядати свої покупки та залишати відгуки.

На головній сторінці присутня вкладка «Акції», де користувач має змогу ознайомитись з списком пропозицій для нього та обрати акцію під себе.

На вкладці «Літо» та «Зима» користувачі можуть переглянути список заходів, які їм доступні в цю пору року.

На вкладці «Скі-паси» користувачі мають змогу купити, або орендувати на прокат скі-паси у процесі відпочинку.

На вкладці «Готелі» користувачі можуть замовити номер в готелі (рис.1.2).

На цій вкладці також є присутній фільтр готелів, де можна обрати зручності, харчування, кількість людей, відстань до розважального комплексу.

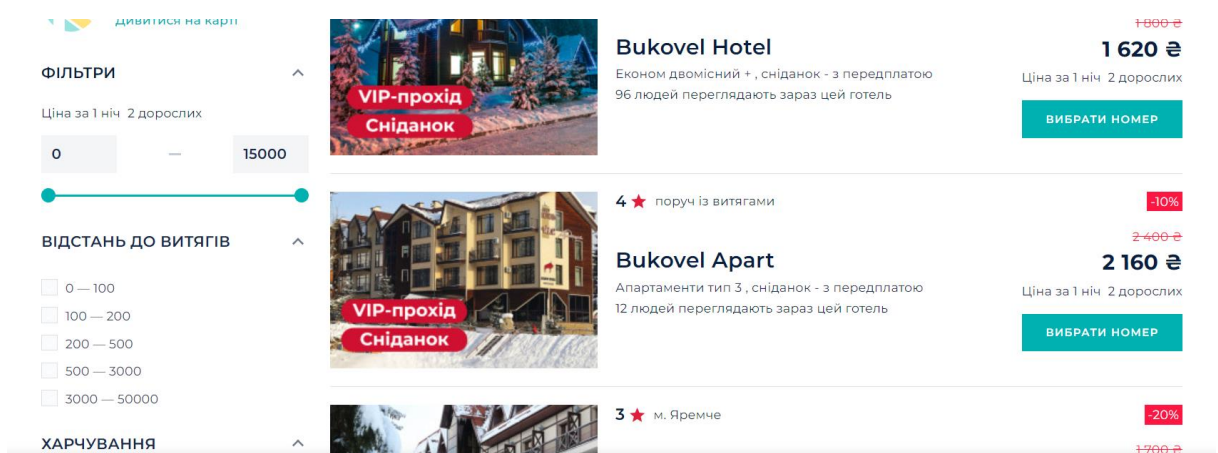


Рисунок 1.2 – Вкладка «Готелі»

Джерело: побудовано автором (знімок з екрану)

Також сайт наповнений відео- та фотоматеріалами та гугл картою (рис.1.3).

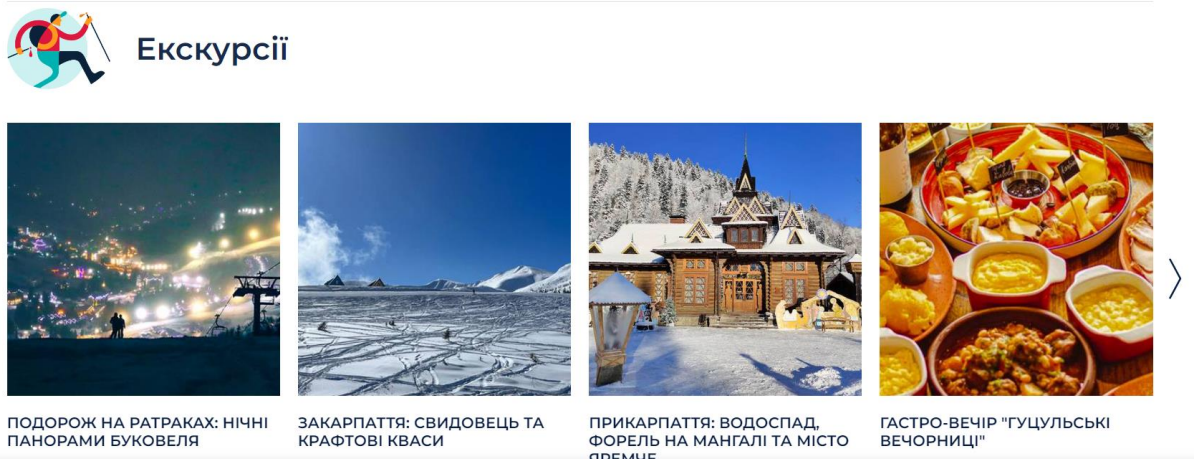


Рисунок 1.3 – Фотоматеріали

Джерело: побудовано автором (знімок з екрану)

Web-система «Буковель» є хорошим рішенням для клієнтів, що хочуть ознайомитись з можливістю відпочинку в Буковелі.

Далі розглянуто Web-систему готелю «Турист».

Web-система являє собою інформаційну сторінку, де користувачі мають змогу переглянути інформацію про готель (рис.1.4).

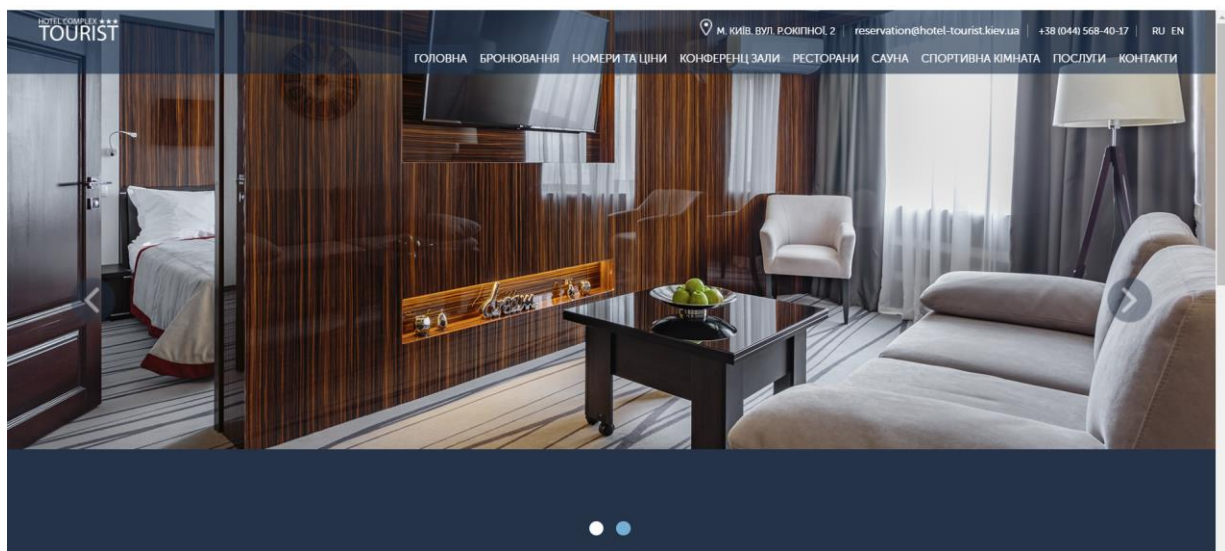


Рисунок 1.4 – Головна сторінка web-системи готелю

Джерело: побудовано автором (знімок з екрану)

На головній сторінці присутні контактні дані, завдяки чому користувачі можуть зв'язатись з адміністрацією готелю та забронювати собі номер. Також тут

присутня вкладка «Бронювання», де користувач має змогу забронювати номер на певну дату. Для оплати бронювання можна використовувати кредитну карточку. На вкладках «Номери та ціни», «Спортивна кімната», «Конференц зали» та «Ресторани» користувачі мають змогу ознайомитись з виглядом готелю (рис.1.5-1.6).



Рисунок 1.5 – Номери готелю

Джерело: побудовано автором (знімок з екрану)

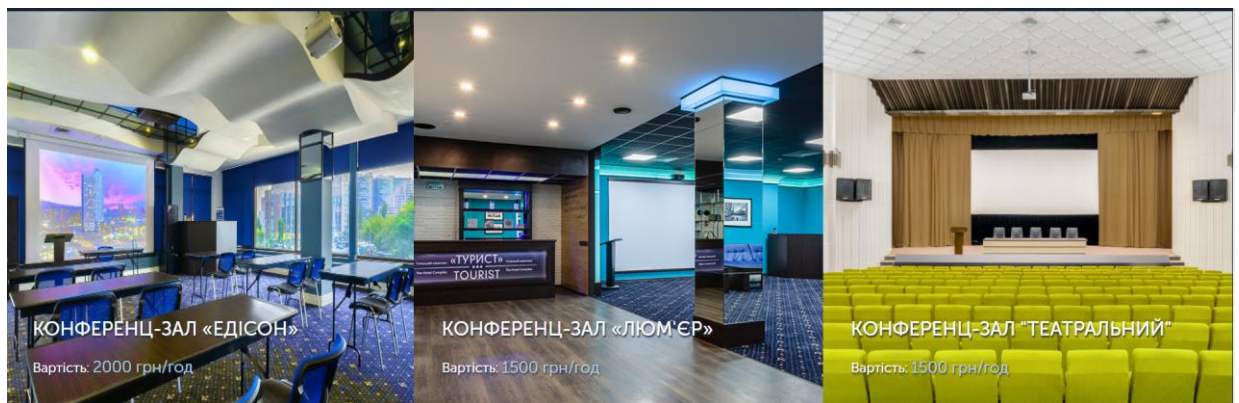


Рисунок 1.6 – Конференц зали

Джерело: побудовано автором (знімок з екрану)

На цих вкладках присутні фотогалереї для кращого ознайомлення. Додатковим функціоналом web-системи є можливість переключати мову сайту.

Загалом web-система «Турист» дозволяє ознайомитись з виглядом номерів готелю та визначитись з місцем проживання в готелі.

Web-система розважального комплексу «Чубі Бум». Web-система являє собою інформаційну сторінку, де користувачі мають змогу переглянути інформацію про розважальний комплекс (рис.1.7).

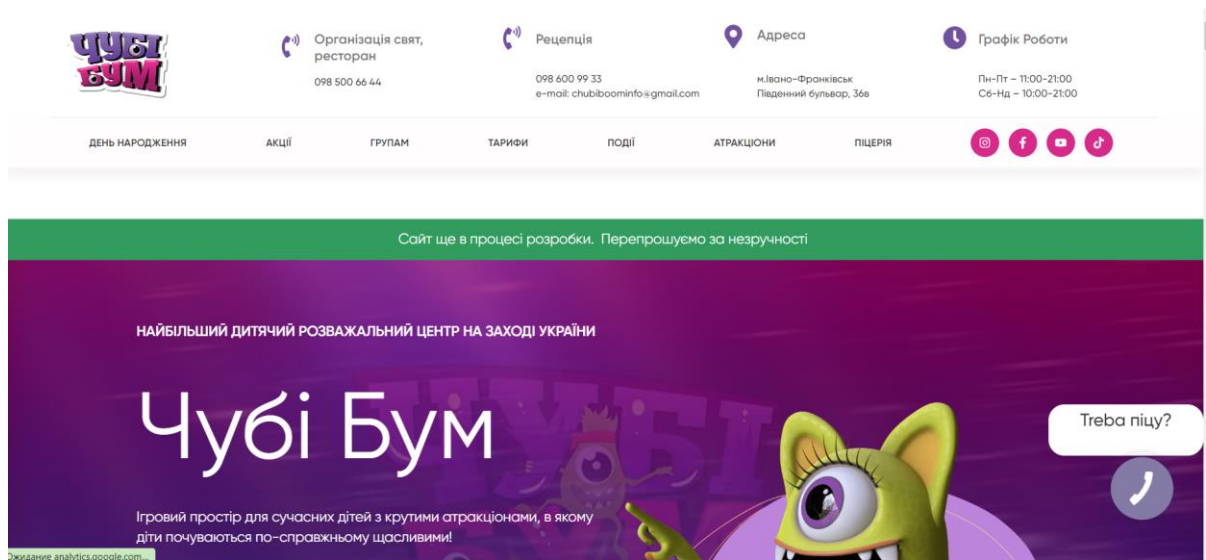


Рисунок 1.7 – Головна сторінка web-системи комплексу

Джерело: побудовано автором (знімок з екрану)

На головній сторінці присутні контактні дані, завдяки чому користувачі можуть зв'язатись з адміністрацією розважального комплексу та замовити собі організацію свят. Також тут присутня вкладка «Акції», де користувач може переглянути список доступних акцій.

Вкладка «Тарифи», де користувач може ознайомитись з доступними тарифами. У вкладках «Події», «Атракціони» та «Піцерія» користувач може ознайомитись з основними функціями розважального комплексу, ці вкладки, крім опису, також містять фотоматеріали (рис.1.8), що допоможе користувачам краще зрозуміти роботу web-комплексу. Додатково користувач може глянути адресу та графік роботи.



Нові розваги у Чубі Бум

Комплекс атракціонів та ігор віртуальної реальності, що цілком перенесе дітлахів та занурить у дивовижний світ магії!

Дозволений він діткам від 6 років, а у програмі є і VR-фільми (тривалістю 5-10хв), і VR-реальність – повне занурення



Рисунок 1.8 – Фотоматеріали комплексу

Джерело: побудовано автором (знімок з екрану)

Web-система «Чубі Бум» допоможе користувачам ознайомитись з можливостями цього розважального комплексу.

1.2 Висновок з аналізу

Згідно з проведеним аналізом було створено таблицю 1.1.

Таблиця 1.1 – Порівняльний аналіз web-систем

Джерело: побудовано автором

Назва web-системи	"Буковель"	"Турист"	"Чубі Бум"	"Трибуна"
Сучасний дизайн	+	+	-	+
Структурна навігація	+	+	+	+
Фотоматеріали	+	+	+	+
Бронювання/Замовлення послуг	+	+	-	+
Інформація про акції	+	-	-	+

Продовження таблиці 1.1.

Інтерактивна мапа	+	-	-	+
Фільтрування	+	+	+	+
Можливість залишати відгуки	+	+	-	+

Web-системи "Буковель", готелю "Турист" та розважального комплексу "Чубі Бум" відрізняються за наявністю функціоналу. "Буковель" надає розширений функціонал, включаючи географічну карту та фільтрацію готелів, що полегшує користувачам вибір. У готелю "Турист" та "Чубі Бум" більш обмежений функціонал може впливати на зручність користування та інформативність.

З огляду на аналіз таблиці 1.1 можна визначити, які функції можуть бути включені в web-систему "Трибуна":

1. Дизайн:

– Web-система має включати привабливий та сучасний дизайн, який може залучати користувачів.

2. Структурна навігація:

– Система має мати логічну та інтуїтивно зрозумілу структуру, що полегшує користувачам знаходження інформації.

3. Фотоматеріали:

– Система має можливість включення якісних та репрезентативних фотографій для представлення розважального комплексу.

4. Бронювання:

– Система має надавати можливість бронювання на різні послуги, відповідно послугам комплексу.

5. Інформація про акції:

– Система має мати можливість представлення актуальних та цікавих пропозицій для відвідувачів.

6. Фільтрування:

– Система має мати можливість використання гнучких фільтрів для швидкого пошуку інформації чи послуг.

7. Можливість залишати відгуки:

– Система має мати можливість створення платформи для відгуків користувачів, що дозволяє збирати та враховувати думки відвідувачів.

Основаючись на цих позитивних характеристиках та орієнтуючись на представлену таблицю, web-система "Трибуна" згрупувала в собі основні позитивні аспекти з існуючих web-систем, що допоможе їй краще розкрити свій потенціал.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Метою даної дипломної роботи є розробка та впровадження інтелектуальної web-системи, спрямованої на оптимізацію бізнес-процесів розважального комплексу «Трибуна». Ця система покликана полегшити управління та покращити обслуговування в комплексі, забезпечуючи інноваційні інструменти для ефективної взаємодії з клієнтами та оптимізації внутрішніх процесів.

Об'єкт дослідження – підтримка діяльності розважального комплексу «Трибуна».

Предмет дослідження – процес реалізації web-системи підтримки діяльності розважального комплексу «Трибуна».

Цільова аудиторія сайту – користувачі, які планують відвідати «Трибуна», попередньо ознайомившись з послугами, фотоматеріалами.

Основні задачі дипломної роботи:

- Проведення аналізу актуальних аспектів обраної предметної області з метою визначення сучасних тенденцій, викликів та можливостей, що можуть бути використані у контексті досліджуваного дипломного проекту.
- Проектування та моделювання об'єкту дослідження з метою створення концептуальної та технічної основи для подальшої імплементації.
- Розробка дизайн-макету системи для візуалізації структури web-системи та її представлення.
- Розробка та тестування web-системи підтримки діяльності для розважального комплексу.

Зазначена web-система має наступні основні функціональні вимоги:

- *Перегляд Інформації*: Клієнт може зручно переглядати інформацію про сам розважальний комплекс, його структуру та послуги, про анонси подій та інші подробиці. Дізнаватися інформацію про кожен об'єкт комплексу за допомогою

інтерактивної мапи, та переглядати новини, щоб бути к курсі всіх акційних пропозицій та оновлень.

– *Фільтрація та Сортування Подій*: Система повинна надавати можливість клієнтам фільтрувати та сортувати події за різними критеріями.

– *Покупка квитків на події*: Можливість безпосередньо замовити квитки на обрані події, забезпечуючи легкий та ефективний процес оформлення. Автоматичне надсилання клієнту на пошту повідомлень з даними замовлення, а також деталями та інструкціями щодо участі в події.

– *Бронювання спорядження*: Можливість забронювати різні види спортивного спорядження, такі як ковзани та туби. Клієнт може обирати час, розмір або тип спорядження та необхідну кількість.

– *Можливість залишати відгуки*: Клієнт може залишати відгуки на сайті, при цьому йому не потрібно авторизуватись. Відгуки можна фільтрувати за категоріями до яких вони відносяться.

– *Навігація за допомогою інтерактивної мапи*: Візуалізація структури розважального комплексу з можливістю навігації по всім об'єктам для отримання детальної інформації.

– *Адміністративна панель*: Забезпечення зручної системи для управління всією інформацією на сайті, та полегшенню бізнес-процесів з боку адміністрації комплексу.

2.2 Вибір засобів реалізації

Для реалізації web-системи підтримки діяльності розважального комплексу "Трибуна" було ретельно обрано набір технологій, спрямований на досягнення високої ефективності, надійності та швидкості функціонування системи.

Мова програмування PHP є важливим елементом web-розробки та використовується для створення динамічних web-сайтів. В поєднанні з фреймворком Laravel, PHP стає високорівневою та продуктивною технологією. Laravel пропонує широкий набір інструментів для спрощення розробки,

включаючи легкий синтаксис, модульність та шаблонізацію з використанням Blade. Його засоби автоматизації, такі як міграції та Artisan, роблять розробку ефективною та легко підтримуваною.

Стильовий фреймворк Tailwind CSS дозволяє визначати стилі прямо в HTML, що спрощує процес верстки та забезпечує гнучкість у визначенні дизайну. Tailwind CSS базується на концепції utility-first, де стилі визначаються через класи. Це полегшує співпрацю між дизайнерами та розробниками, а також дозволяє швидко налаштовувати та мінімізувати CSS.

JavaScript, використовуючи різноманітні бібліотеки та фреймворки, стає основою динамічної взаємодії на стороні клієнта. У контексті розважального комплексу, JavaScript може бути використаний для реалізації асинхронних запитів, анімацій та взаємодії з користувачем без перезавантаження сторінки.

HTML виступає як мова розмітки, що визначає структуру web-сторінок. Використовуючи HTML, розробники визначають заголовки, параграфи, зображення та інші елементи, щоб створити логічну організацію контенту, яка легко розуміється браузером та робить сторінки доступними для інтерпретації.

MySQL, як система управління базами даних, грає критичну роль у забезпеченні ефективного зберігання та отримання даних. Завдяки мові SQL, розробники можуть легко виконувати складні запити, забезпечуючи швидкий доступ до інформації. Інтеграція MySQL з PHP дозволяє ефективно взаємодіяти з базою даних та забезпечує надійну роботу з великим обсягом інформації.

Обраний набір технологій відображає високий рівень виваженості та обдуманості при розробці web-системи для розважального комплексу "Трибуна". Він спрямований на оптимальну швидкодію, надійність та зручність використання для кінцевого користувача.

3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-СИСТЕМИ

3.1. Розробка структурно-функціональної моделі системи

Даний розділ представляє собою ключовий етап з моделювання та визначення структури web-системи підтримки діяльності розважального комплексу "Трибуна". На цьому етапі проводиться аналіз та проектування системи з використанням інструментів, що забезпечують структуровану та системну оцінку всіх її аспектів.

У сучасному суспільстві розважальні комплекси відіграють значущу роль, переймаючи функцію не лише надання розважальних послуг, але місцем, де можна відпочити, смачно поїсти, та навіть навчатися чомусь новому. З плином часу збільшується попит на заклади, де в межах однієї території можна спробувати багато вид занять, а не обмежувати себе чимось одним. Тому одночасно разом із цим зростає і конкуренція серед таких комплексів. Ефективне управління розважальним комплексом та підвищення рівня обслуговування стають викликом, що вимагає модернізації систем управління.

Один з ключових викликів для сучасних розважальних комплексів – це потреба у вдосконаленні систем управління. Багато з них залишаються в минулому, використовуючи застарілі методи, такі як запис на послугу чи бронювання місця вручну, що часто може призводити до непорозумінь і нестиковці даних. Це негативно впливає як на враження клієнтів, так і на репутацію закладу. Для вирішення цих питань розроблена стратегія з впровадження більш сучасних методів та технологій управління та організації роботи закладу. В результаті це позитивно вплине ефективність та якість обслуговування клієнтів.

При такому підході, розробка web-систем стає необхідністю. Створення спеціалізованих web-систем, які включають замовлення квитка на заходи онлайн, бронювання різних послуг, адмін-панель, яка забезпечує інтерфейс для управління та взаємодії з базою даних, в якій зберігається вся необхідна інформація про комплекс та його послуги - все це дозволяє розважальним комплексам оптимізувати бізнес процеси та взаємодії з клієнтами.

Взаємодія з клієнтом через онлайн платформи, соціальні мережі та електронну пошту, наразі є актуальним і ключовим інструментом для просування послуг та отримання зворотного зв'язку. Інтеграція таких інструментів у web-систему комплексу є критично необхідним рішенням для підтримки інфраструктури системи.

Таким чином, сучасні розважальні комплекси повинні застосовувати сучасні методи управління та інноваційні технології задля того, щоб підвищити якість обслуговування клієнтів, забезпечення їх комфорту, оптимізацію бізнес процесів та конкурентоспроможності у даній сфері. Одним із кроків для досягнення цілей по розвитку комплексу та успіху у його просуванні є впровадження інформаційних технологій у життєвий процес системи.

Першочерговою задачею є визначення моделі бізнес-процесів комплексу "Трибуна" за допомогою мови моделювання IDEF (Integrated Definition for Function Modeling). IDEF – це стандартизована мова моделювання, яка дозволяє представити структуру та функціональні взаємодії в системі. На цьому етапі визначаються основні бізнес-процеси, такі як обробка замовлень, керування запасами, облік клієнтів та інші, що мають важливе значення для діяльності "Трибуна".

Методології IDEF0 в рамках даного проекту дозволяє ретельно проаналізувати та сформулювати функції, які має виконувати web-система підтримки діяльності розважального комплексу "Трибуна". Це створює стійку фундамент для подальшого проектування та реалізації системи, забезпечуючи її оптимальну ефективність та функціональність. Вигляд методології IDEF0 відображено на рисунку 3.1.

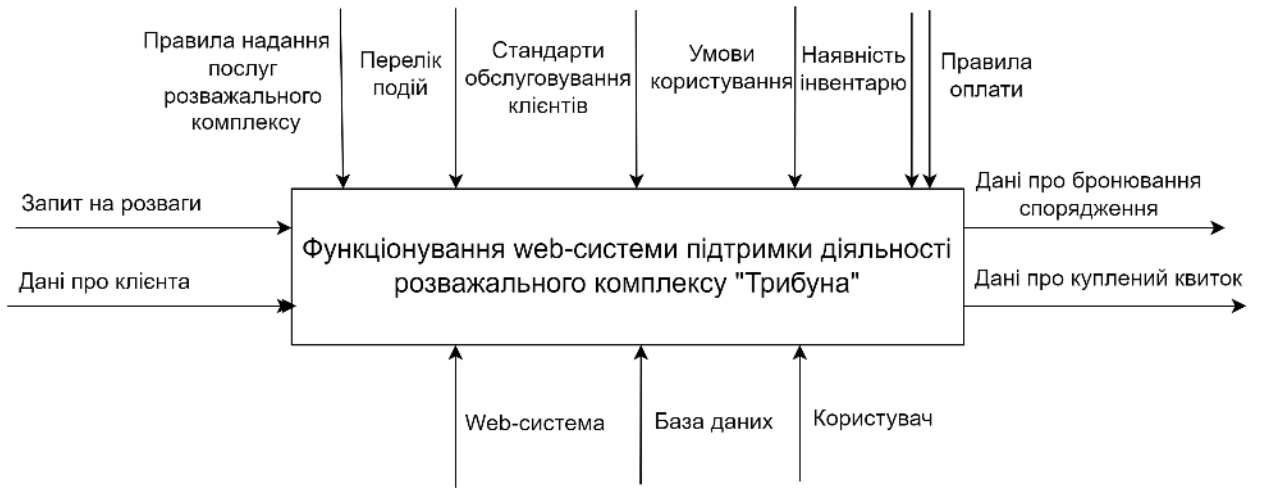


Рисунок 3.1 – IDEF0 методологія системи.

Джерело: побудовано автором (знімок з екрану)

Другий рівень декомпозиції IDEF0 представлений на рисунку 3.2.

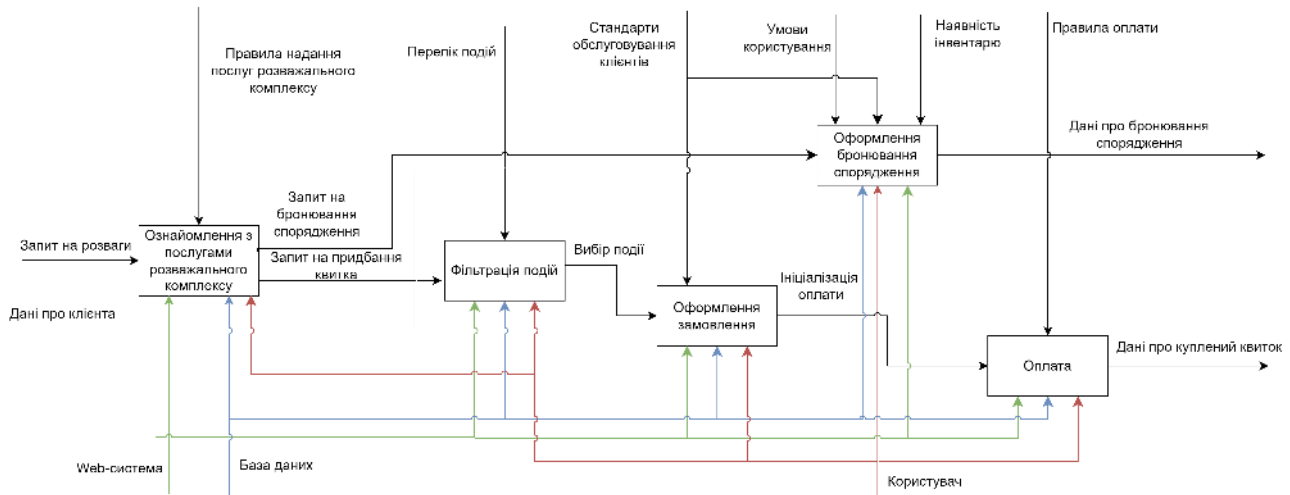


Рисунок 3.2 – Декомпозиція системи

Джерело: побудовано автором(знімок з екрану)

3.2. Моделювання варіантів використання системи

За допомогою UML (Unified Modeling Language) розробляються діаграми, які дозволяють детально описати структуру системи.

UML-діаграми є потужним інструментом для моделювання та проектування web-системи підтримки діяльності розважального комплексу "Трибуна". Вони надають зручний спосіб візуалізації структури та функціональності системи, що допомагає розробникам розуміти, аналізувати та вдосконалювати її перед тим, як перейти до фази реалізації. Рисунок 3.3 містить відображення основної структури системи у вигляді UML-діаграми.

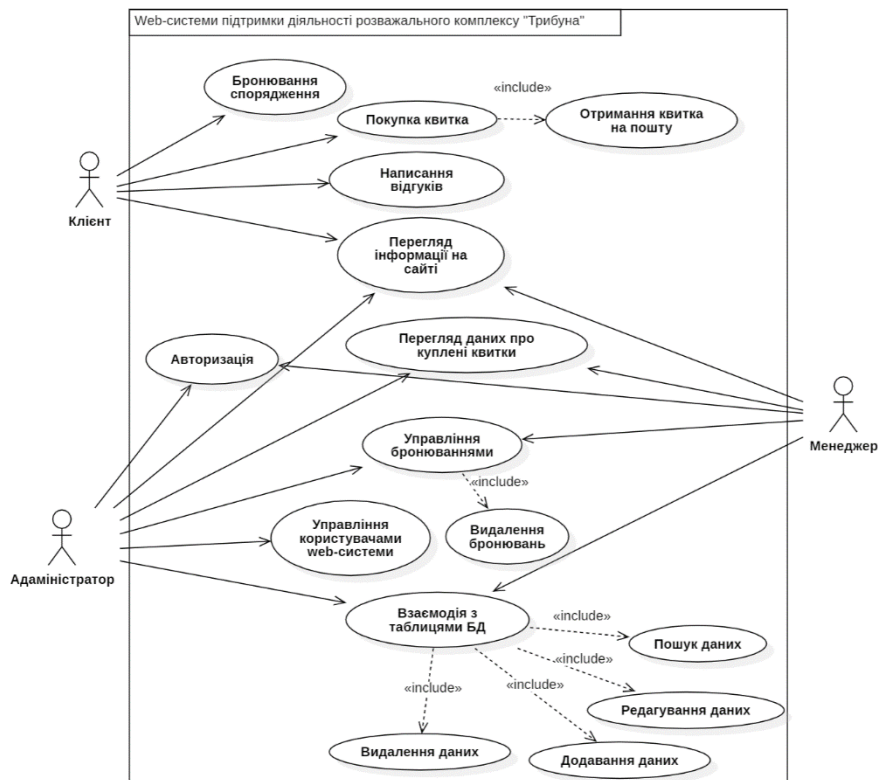


Рисунок 3.3 – UML Use-case діаграма системи
Джерело: побудовано автором(знімок з екрану)

3.3. Проектування моделі бази даних

Особливу увагу приділяють проектуванню моделі бази даних. Вона визначає структуру бази даних, таблиці, їх поля та зв'язки між ними. Зазвичай, для цього

використовують реляційні бази даних, і важливо правильно спроектувати схему бази даних, щоб забезпечити ефективне зберігання та доступ до інформації.

Важливо правильно спроектувати схему бази даних, визначивши таблиці, їх поля та зв'язки між ними. Наприклад, можна використовувати зовнішні ключі для зв'язування таблиць та забезпечення цілісності даних. Для оптимізації швидкодії системи, слід ретельно розглянути індексацію полів, яка дозволяє прискорити пошук та фільтрацію даних.

Для взаємодії з базою даних можна використовувати мову запитів SQL для виконання операцій вставки, оновлення, видалення та вибірки даних. Крім того, розглядається можливість використання ORM (Object-Relational Mapping) фреймворків, які дозволяють взаємодіяти з базою даних у більш об'єктно-орієнтованому способі.

У підсумку, проектування бази даних є ключовим етапом у розробці web-системи для розважального комплексу "Трибуна". Правильно спроектована та оптимізована база даних допоможе забезпечити ефективну роботу системи та задовольнити потреби користувачів, що є найважливішою метою цього проекту.

Задача створення структури бази даних для розважального комплексу "Трибуна" включає в себе кілька таблиць, кожна з яких має свою визначену роль у забезпеченні функціоналу системи. Наступний рисунок 3.4 відображає структуру бази даних системи.

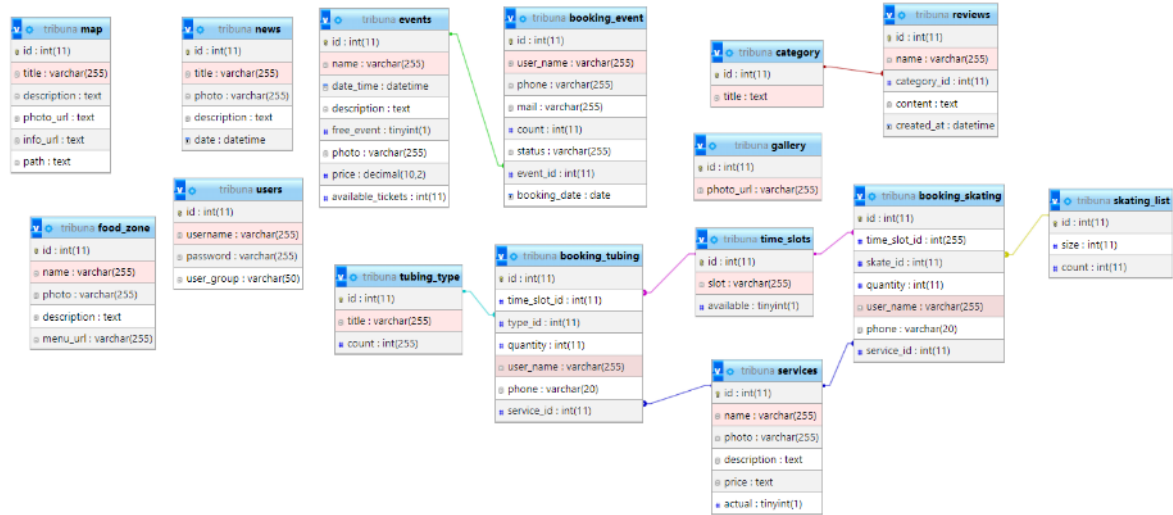


Рисунок 3.4 – База даних системи

Джерело: побудовано автором(знімок з екрану)

Ці таблиці визначають структуру бази даних для розважального комплексу "Трибуна", забезпечуючи необхідний функціонал, такий як перегляд інформації, замовлення квитків, написання відгуків, перегляд та редагування інформації у web-системі.

Далі представлено детальну структуру кожної з таблиць БД.

На рисунку 3.5 представлено структуру таблиці купівлі квитків.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 user_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	3 phone	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	4 mail	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	5 count	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	6 event_id	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	7 booking_date	date			Ні	current_timestamp()			Більше

↑ Позначити все Вибрані:

Рисунок 3.5 – Структура таблиці booking_events

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.6 представлено структуру таблиці бронювання ковзанів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 time_slot_id	int(255)			Ні	Немає			Більше
<input type="checkbox"/>	3 skate_id	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	4 quantity	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	5 user_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	6 phone	varchar(20)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	7 service_id	int(11)			Ні	Немає			Більше

Рисунок 3.6 – Структура таблиці booking_skating

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.7 представлено структуру таблиці бронювання тубінгів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 time_slot_id	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	3 type_id	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	4 quantity	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	5 user_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	6 phone	varchar(20)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	7 service_id	int(11)			Так	NULL			Більше

Рисунок 3.7 – Структура таблиці booking_tubing

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.8 представлено структуру таблиці категорій відгуків.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id 🔑	int(11)		Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2	title	text	utf8mb4_unicode_ci	Ні	Немає			Більше ▾

Рисунок 3.8 – Структура таблиці category

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.9 представлено структуру таблиці подій.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id 🔑	int(11)		Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2	name	varchar(255)	utf8mb4_unicode_ci	Ні	Немає			Більше ▾
<input type="checkbox"/>	3	date_time	datetime		Ні	Немає			Більше ▾
<input type="checkbox"/>	4	description	text	utf8mb4_unicode_ci	Так	NULL			Більше ▾
<input type="checkbox"/>	5	free_event	tinyint(1)		Ні	Немає			Більше ▾
<input type="checkbox"/>	6	photo	varchar(255)	utf8mb4_unicode_ci	Так	NULL			Більше ▾
<input type="checkbox"/>	7	price	decimal(10,2)		Так	NULL			Більше ▾
<input type="checkbox"/>	8	available_tickets	int(11)		Так	NULL			Більше ▾

Рисунок 3.9 – Структура таблиці events

Джерело: побудовано автором(знімок з екрану)

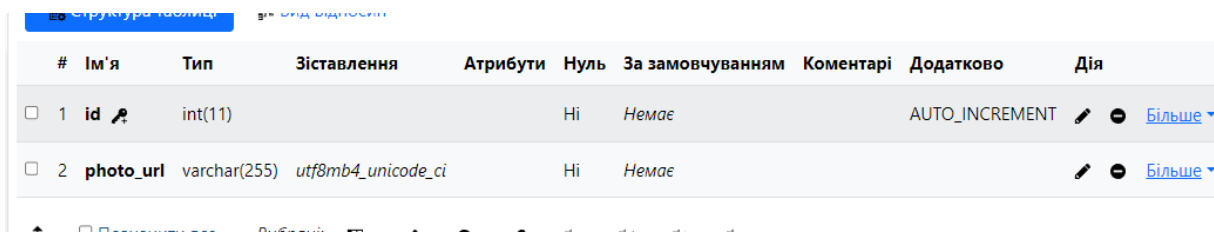
На рисунку 3.10 представлено структуру таблиці фуд-зон.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id 🔑	int(11)		Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2	name	varchar(255)	utf8mb4_unicode_ci	Ні	Немає			Більше ▾
<input type="checkbox"/>	3	photo	varchar(255)	utf8mb4_unicode_ci	Так	NULL			Більше ▾
<input type="checkbox"/>	4	description	text	utf8mb4_unicode_ci	Так	NULL			Більше ▾
<input type="checkbox"/>	5	menu_url	varchar(255)	utf8mb4_unicode_ci	Ні	Немає			Більше ▾

Рисунок 3.10 – Структура таблиці food_zones

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.11 представлено структуру таблиці галереї



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	photo_url	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше

Рисунок 3.11 – Структура таблиці gallery

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.12 представлено структуру таблиці новин.

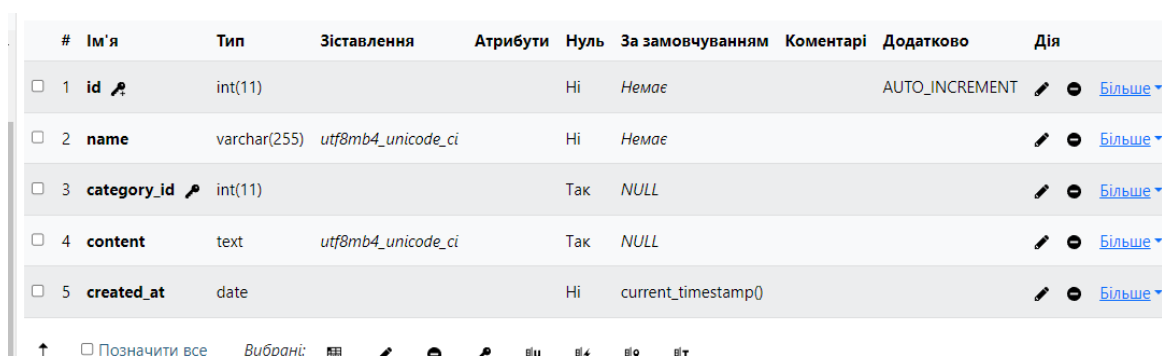


#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	photo	varchar(255)	utf8mb4_unicode_ci		Так	NULL			Більше
4	description	text	utf8mb4_unicode_ci		Так	NULL			Більше
5	date	datetime			Ні	current_timestamp()			Більше

Рисунок 3.12 – Структура таблиці news

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.13 представлено структуру таблиці відгуків.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	category_id	int(11)			Так	NULL			Більше
4	content	text	utf8mb4_unicode_ci		Так	NULL			Більше
5	created_at	date			Ні	current_timestamp()			Більше

Рисунок 3.13 – Структура таблиці reviews

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.14 представлено структуру таблиці послуг.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔗	int(11)			Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 photo	varchar(255)	utf8mb4_unicode_ci		Так	NULL			Більше ▾
<input type="checkbox"/>	4 description	text	utf8mb4_unicode_ci		Так	NULL			Більше ▾
<input type="checkbox"/>	5 price	text	utf8mb4_unicode_ci		Так	NULL			Більше ▾

Рисунок 3.14 – Структура таблиці services

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.15 представлено структуру таблиці користувачів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔗	int(11)			Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 username	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 password	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	4 user_group	varchar(50)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾

Рисунок 3.15 – Структура таблиці users

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.16 представлено структуру таблиці спорядження - ковзанів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔗	int			Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 size	int			Ні	Немає			Більше ▾
<input type="checkbox"/>	3 count	int			Ні	Немає			Більше ▾

Рисунок 3.16 – Структура таблиці спорядження – skiting_list

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.17 представлено структуру таблиці спорядження - туби.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2	size			Ні	Немає			Більше
<input type="checkbox"/>	3	count			Ні	Немає			Більше

Рисунок 3.17 – Структура таблиці спорядження – tubing_type

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.18 представлено структуру таблиці часових проміжків для бронювання спорядження

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2	slot	varchar(255)	utf8mb4_unicode_ci	Ні	Немає			Більше
<input type="checkbox"/>	3	available	tinyint(1)		Ні	Немає			Більше

Рисунок 3.18 – Структура таблиці спорядження – time_slots

Джерело: побудовано автором(знімок з екрану)

На рисунку 3.19 представлено структуру таблиці для інтерактивної мапи

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2	title	varchar(255)	utf8mb4_unicode_ci	Ні	Немає			Більше
<input type="checkbox"/>	3	description	text	utf8mb4_unicode_ci	Ні	Немає			Більше
<input type="checkbox"/>	4	photo_url	text	utf8mb4_unicode_ci	Ні	Немає			Більше
<input type="checkbox"/>	5	info_url	text	utf8mb4_unicode_ci	Ні	Немає			Більше
<input type="checkbox"/>	6	path	text	utf8mb4_unicode_ci	Ні	Немає			Більше

Рисунок 3.19 – Структура таблиці спорядження - map

Джерело: побудовано автором(знімок з екрану)

3.3. Створення дизайн-макету web-системи

Під час проектування web-системи створено макет застосунку. А саме наступні сторінки (представлено на рисунку 3.20):

- Головна сторінка разом із слайдером.
- Сторінка «Заходи» та сторінка окремого заходу.
- Сторінка «Послуги» та сторінка окремої послуги.

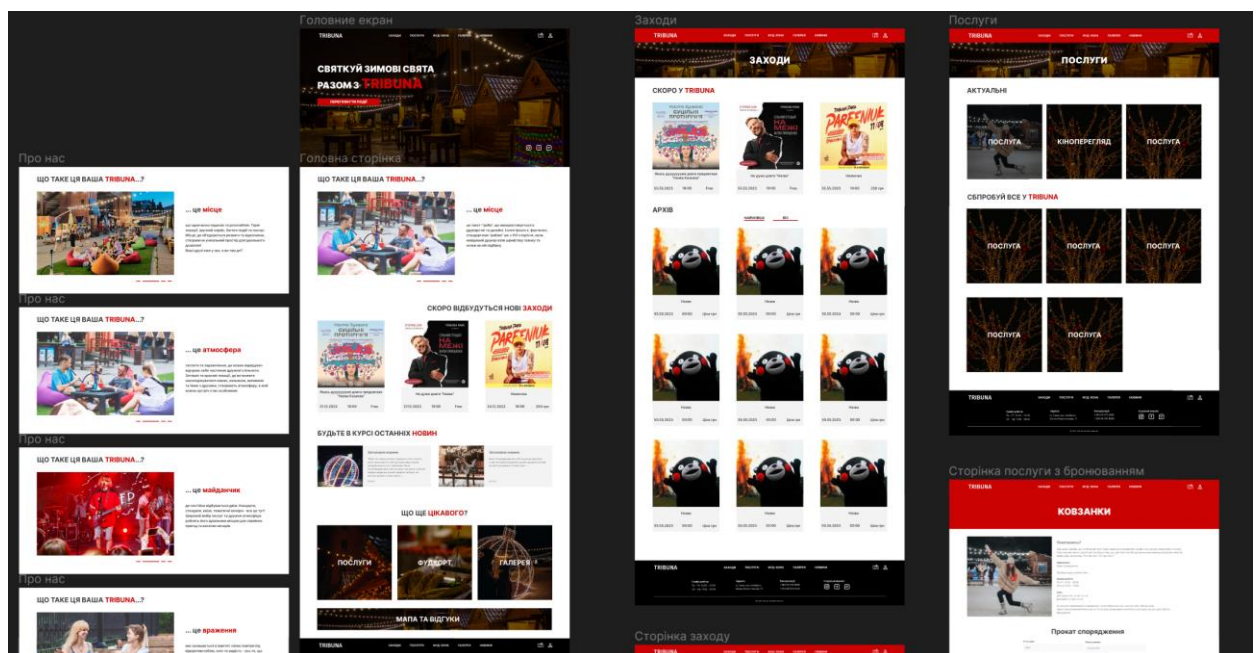


Рисунок 3.20– Макет сторінок: Головна, Заходи, Послуги

Джерело: побудовано автором(знімок з екрану)

Також наступні сторінки (представлено на рисунку 3.21):

- Сторінка «Фуд-зона» та сторінка окремого домику.
- Сторінка «Галерея»
- Сторінка «Новини» та сторінка окремого посту.
- Сторінка «Мапа та відгуки» з інтерактивної мапою та відгуками.

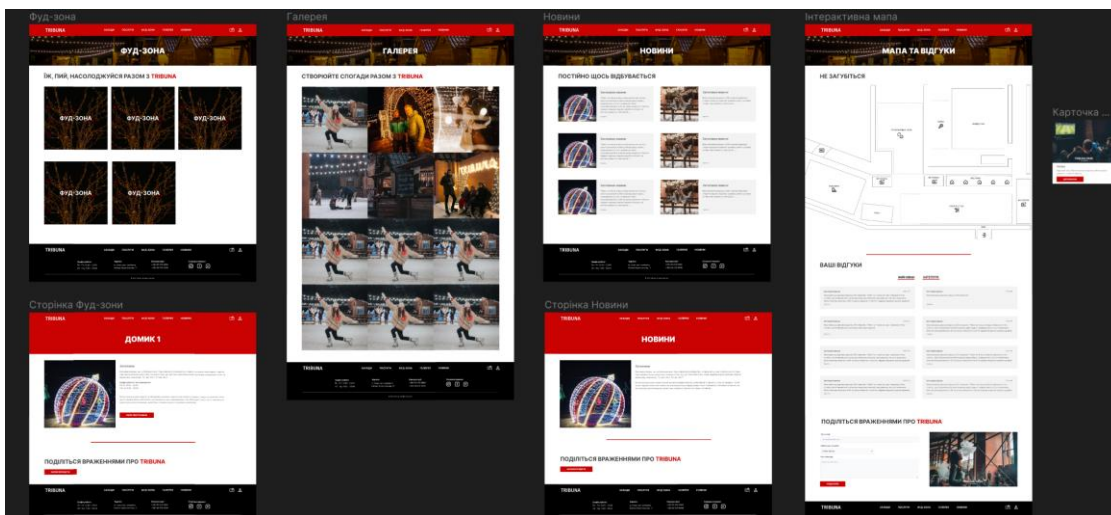


Рисунок 3.21 – Макет сторінок: Головна, Заходи, Послуги
Джерело: побудовано автором(знімок з екрану)

Було розроблено макети сторінок Адмін-панелі (рис. 3.22):

- Авторизація.
- Головна сторінка від вигляду Адміністратора та Менеджера.
- Сторінки «База даних» розбита по розділам.
- Сторінки «Квитки» та «Бронювання».
- Сторінка «Користувачі».

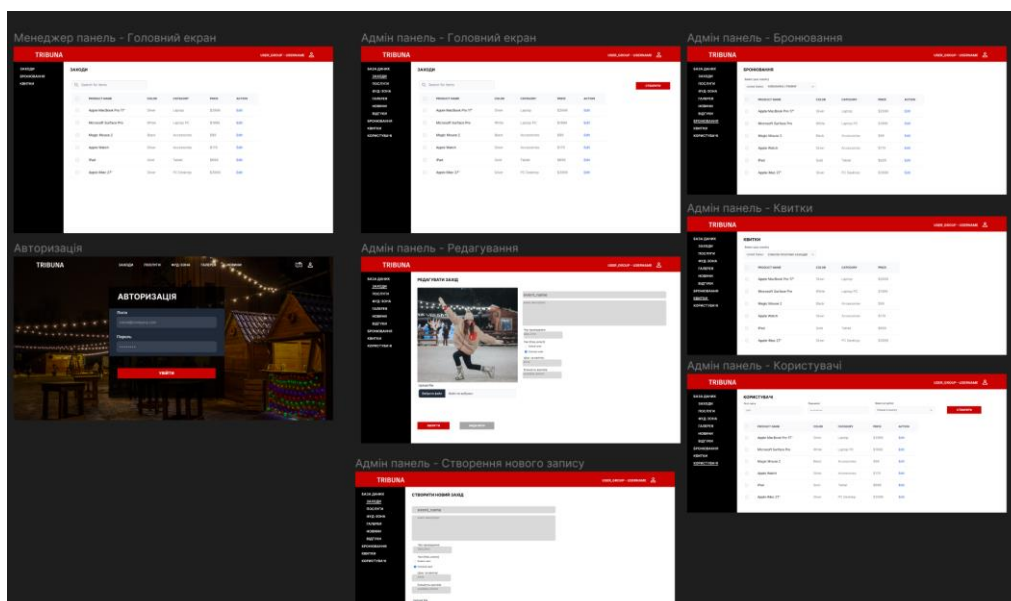


Рисунок 3.22 – Макет Адмін-панелі
Джерело: побудовано автором(знімок з екрану)

4 РОЗРОБКА WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ

4.1. Розробка web-системи

Під час виконання кваліфікаційної роботи розроблено наступні сторінки:

Головна сторінка (представлено на рисунку 4.1). На ній знаходиться основна інформація про «Трибуна». Виведені найближчі заходи, осанні новини та інші елементи навігації.



Рисунок 4.1 – Реалізація головної сторінки

Джерело: побудовано автором (знімок з екрану)

Сторінка з відображенням заходів (представлено на рисунку 4.2). Сторінка розділена на два блоки. Блок «Скоро у Трибуна» відображає всі події які мають відбутися у майбутньому. Блок «Архів» відображає минулі події, про них можна переглядати інформацію, але не доступне замовлення квитка.

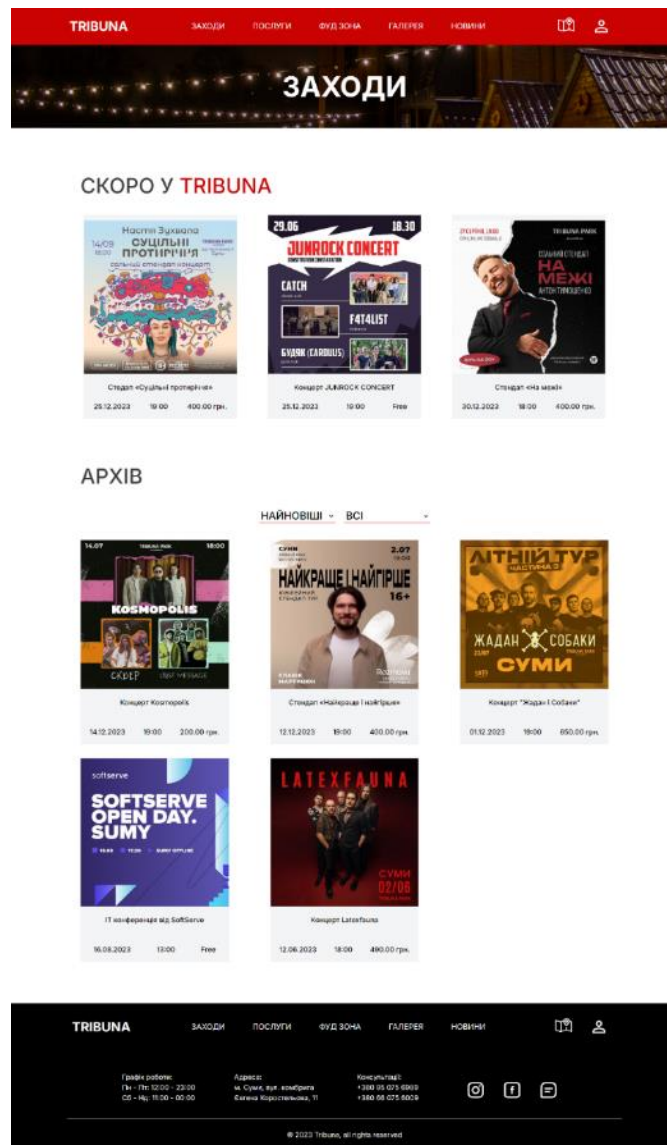


Рисунок 4.2 – Реалізація сторінки заходів

Джерело: побудовано автором(знімок з екрану)

Також розроблено сторінки для кожного з заходів (рис.4.3). На цій сторінці виведена основна інформація про захід. Заходи є безкоштовні та платні. Для платних заходів відображається форма замовлення квитка. Після відправки форми користувача переводить на сторінку оплати, де він вводить

дані картки. Після успішної оплати квиток приходить на вказану пошту користувача (додатково див. розділ 4.1. Тестування web-системи, рис. 4.11)

TRIBUNA ЗАХОДИ ПОСЛУГИ ФУД ЗОНА ГАЛЕРЕЯ НОВИНИ

ЗАХІД

Настя Зухвала
СУЦІЛЬНІ ПРОТИРІЧЧЯ
 сольний стендап концерт
 14/09 18:00
 TRIBUNA PARK
 вул. Коростельська, 11
 Суми

Степа «Суцільні протиріччя»

Настя Зухвала. Сольний стендап «Суцільні протиріччя» ішов боже який день. Боже якого місяця десятого року однієї війни з росіянами. Французі розливали під повітряну тривогу, шахеї летіли над розважальними центрами (якщо пощастить), цитієльні дивились TikTok, як військові танцюють і посміялися, військові дивилися TikTok як цитієльні вчать виши цитієльні вивчати і тек. На Барокках стали продавати вазочки, уламки снарядів і вазочки з уламків снарядів. Меми перемішалися зі зборами, пісні перемішалися зі зборами, стендапи перемішалися зі зборами... Єдиним острівець стабільності стала русофобія. А вона, як короче вано, з роками став тільки краще. Якщо 2023 рік, відбудеться все те, що не могло відбутися. Принаймні одночасно. Ви бачите в цьому комедію? Ми так! Якщо ви теж — ми вас запрошуємо на новий сольний стендап Насті «Суцільні протиріччя».

Час проведення:
2023-12-25 19:00:00

Ціна за квиток:
400.00

Кількість квитків:
300

Замовлення квитка

Ім'я Телефон

Пошта

Кількість квитків

Оплатити

ПОДІЛІТЬСЯ ВРАЖЕННЯМИ ПРО TRIBUNA

Написати відгук

TRIBUNA ЗАХОДИ ПОСЛУГИ ФУД ЗОНА ГАЛЕРЕЯ НОВИНИ

Графік роботи:
Пн - Пт 12:00 - 23:00
Сб - Нд 11:00 - 00:00

Адреса:
м. Суми, вул. комбрига
Євгена Коростельського, 11

Консультації:
+380 99 075 6989
+380 66 075 6059

© 2023 Tribuna, all rights reserved

Рисунок 4.3 – Реалізація сторінки певного заходу
Джерело: побудовано автором(знімок з екрану)

На рисунках 4.4 – 4.5 представлено лістинг коду який відповідає за замовлення квитків та їх відправку на пошту замовника.

```

public function ticket_init(){

    $user_name = request()->input( key: 'user_name');
    $event_id = request()->input( key: 'event_id');
    $phone = request()->input( key: 'phone');
    $email = request()->input( key: 'email');
    $tickets_count = request()->input( key: 'tickets_count');
    $status = request()->input( key: 'status');

    $data=array(
        'user_name'=>$user_name,
        "event_id"=>$event_id,
        "phone"=>$phone,
        "mail"=>$email,
        "count"=>$tickets_count,
        "status"=>$status
    );

    BookingEvent::create($data);
    $bookingEvent = BookingEvent::where('mail', '=', $email)->first();
    return view( view: 'payment', compact( var_name: 'bookingEvent'));
}

public function init_payment(BookingEvent $bookingEvent){
    $data=array('status'=>'Оплачено');
    $bookingEvent->update($data);
    session(['bookingEvent' => $bookingEvent->id]);

    return redirect()->route( route: 'send');
}

```

Рисунок 4.4 – Лістинг коду відповідального за замовлення квитків

Джерело: побудовано автором (знімок з екрану)

```

no usages
public function attachments(): array
{
    return [];
}

/**
 * Build the message.
 *
 * @return $this
 */
public function build()
{
    $bookingEvent = BookingEvent::where('id', '=', $this->bookingEvent_id)->first();
    $email = $bookingEvent->mail;
    $event = Event::where('id', '=', $bookingEvent->event_id)->first();
    return $this
        ->to($email)
        ->subject( subject: 'New Mail')
        ->markdown( view: 'email', compact(['bookingEvent', 'event']));
}
}

```

Рисунок 4.5 – Лістинг коду відповідального за відправку квитка на пошту

Джерело: побудовано автором (знімок з екрану)

Наступною було розроблено сторінку послуг (рис. 4.6). Вона також розділена на два блоки «Актуальне», де відображаються актуальні наразі послуги і «Спробуйте все у Трібуна», де відображаються всі інші послуги.

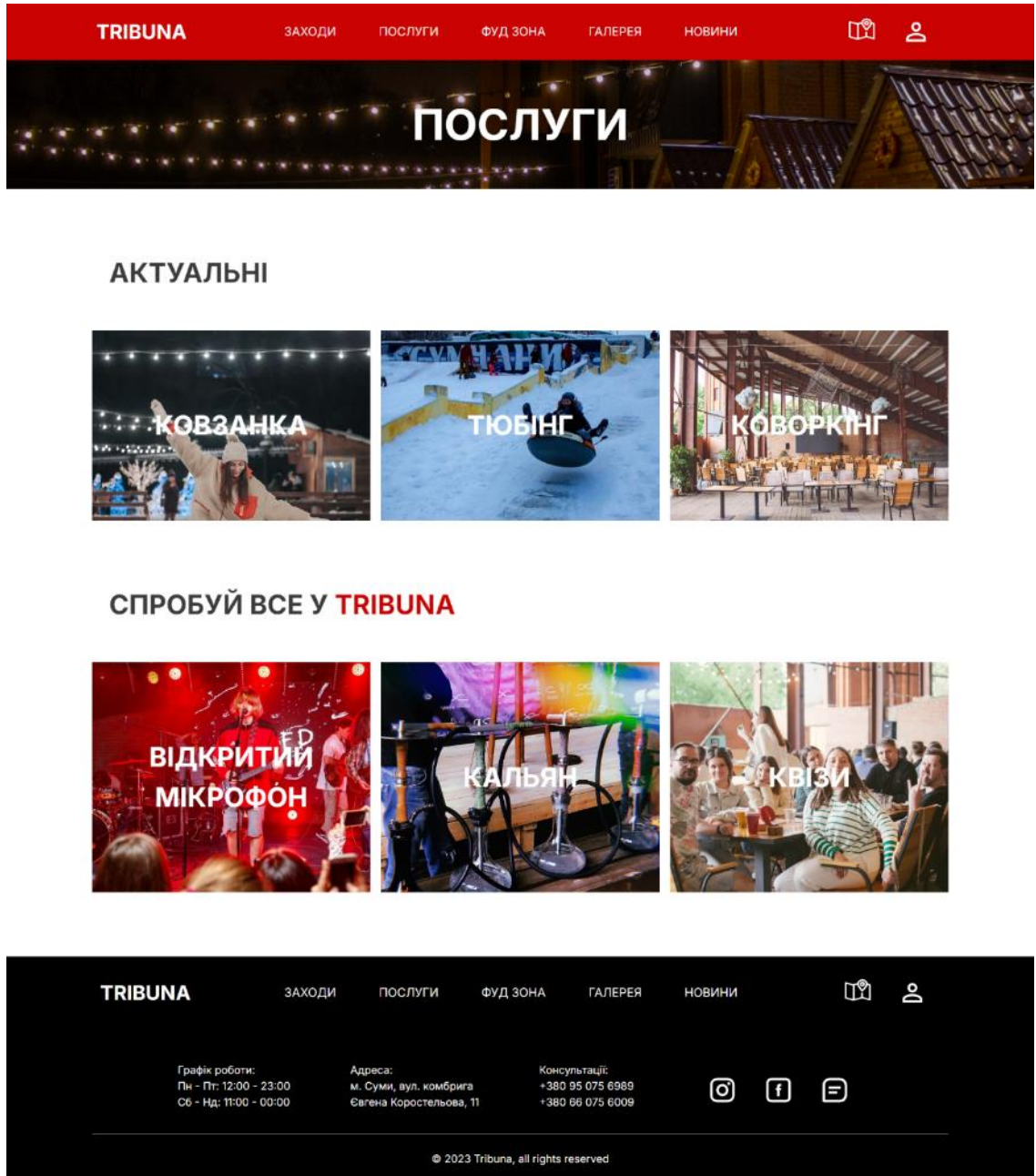


Рисунок 4.6 – Реалізована сторінка з послугами


Джерело: побудовано автором (знімок з екрану)

Також реалізовано можливість бронювати певне спорядження у таких послугах як «Ковзанка» та «Тюбінг», представлено на рисунку 4.7. Користувач має змогу обрати час на сьогодні, розмір або вид спорядження, і необхідну кількість.

TRIBUNA

[ЗАХОДИ](#)
[ПОСЛУГИ](#)
[ФУД ЗОНА](#)
[ГАЛЕРЕЯ](#)
[НОВИНИ](#)

ПОСЛУГА



КОВЗАНКА

"Хочете покататись на ковзанах ? Тоді вам до нас! Пориньте у казкову атмосферу ковзанки на території нашого комплексу! Ми дуже стараніємось і рівняємо її для вас, щоб ніщо не заважало вам отримувати насолоду від плавного катання! Якщо Вам подобається активний та веселий відпочинок, то Вам неодмінно варто відвідати ковзанку в розважальному комплексі "Tribuna". На Вас чекає велика арена площею більше 200 кв. м., доступні ціни, а також можливість приходити зі своїми ковзанами Ковзанка - це прекрасне місце відпочинку для всіх охочих. Графік роботи Пн-Нд 12:00 - 20:00 Ви можете забронювати спорядження на потрібний вам час у нас на сайті, або на місці. Увага! При записі більше ніж на 10 хв ваше бронювання може бути скасовано, якщо в цей час є багато відвідувачів."

"Ціни: Для дорослих: 130 грн / 1 год Для дітей: 80 грн / 1 год"

Ім'я

Телефон

Оберіть час

Оберіть розмір

Кількість


Забронювати

ПОДІЛІТЬСЯ ВРАЖЕННЯМИ ПРО TRIBUNA

Написати відгук

TRIBUNA

[ЗАХОДИ](#)
[ПОСЛУГИ](#)
[ФУД ЗОНА](#)
[ГАЛЕРЕЯ](#)
[НОВИНИ](#)



Графік роботи:
Пн - Пт: 12:00 - 23:00
Сб - Нд: 11:00 - 00:00

Адреса:
м. Суми, вул. комбрига
Світлана Коростельова, 11

Консультації:
+380 95 075 6989
+380 66 075 6009

© 2023 Tribuna, all rights reserved

Рисунок 4.7 – Реалізація додавання бронювання
Джерело: побудовано автором(знімок з екрану)

Сторінка зі списком доступних фуд-зон, представлено на рисунку 4.8. На ній перелічені всі фуд-зони, після відкриття сторінки відображається інформація про конкретний домок та меню і вигляді файлу.

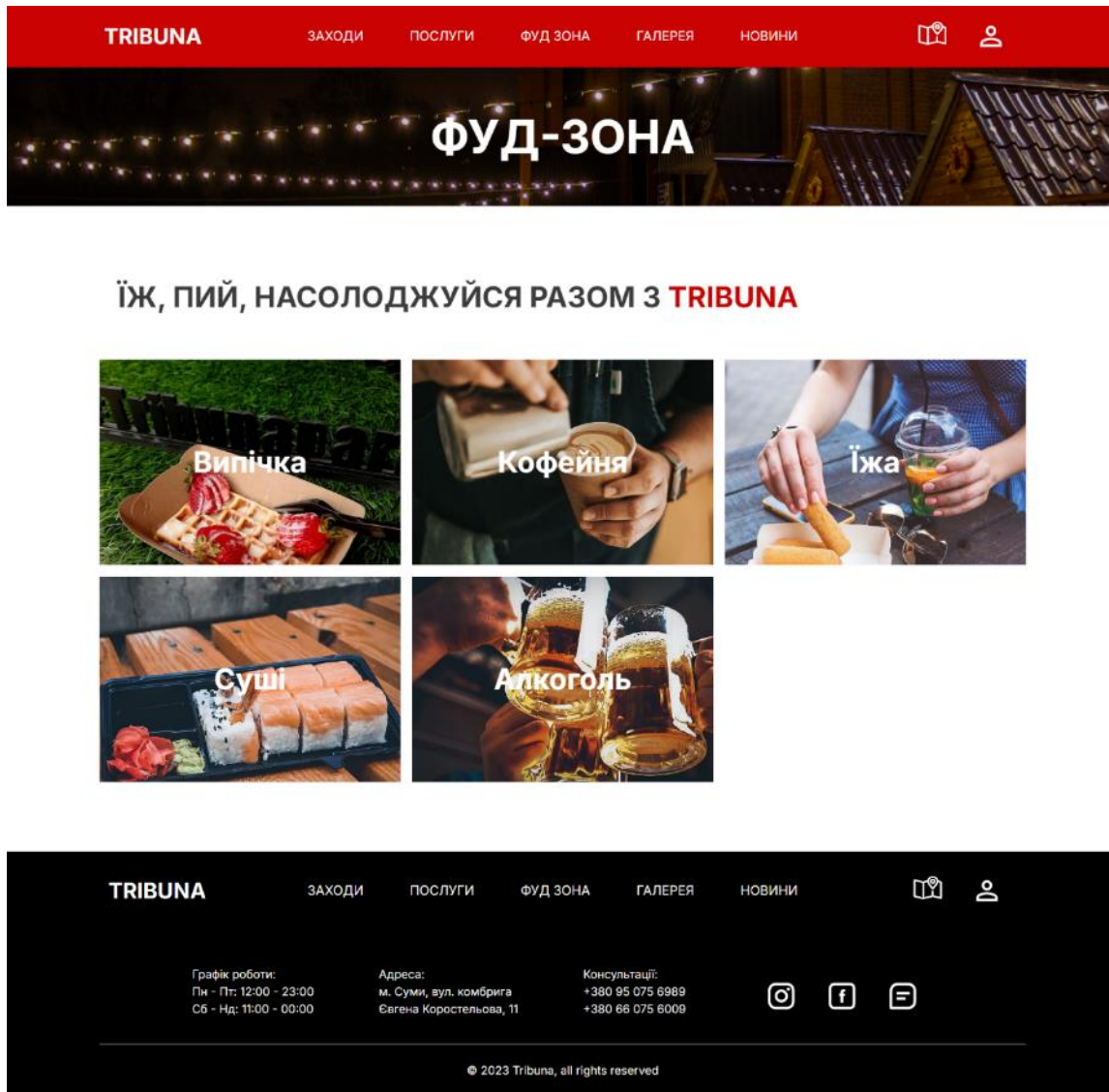


Рисунок 4.8 – Реалізація сторінки фуд-зон
Джерело: побудовано автором (знімок з екрану)

Сторінка зі списком новин, представлена на рисунку 4.9. На ній виводяться всі новини від новіших до старіших. Кнопка «Детальніше» веде нас на сторінку окремої новини.

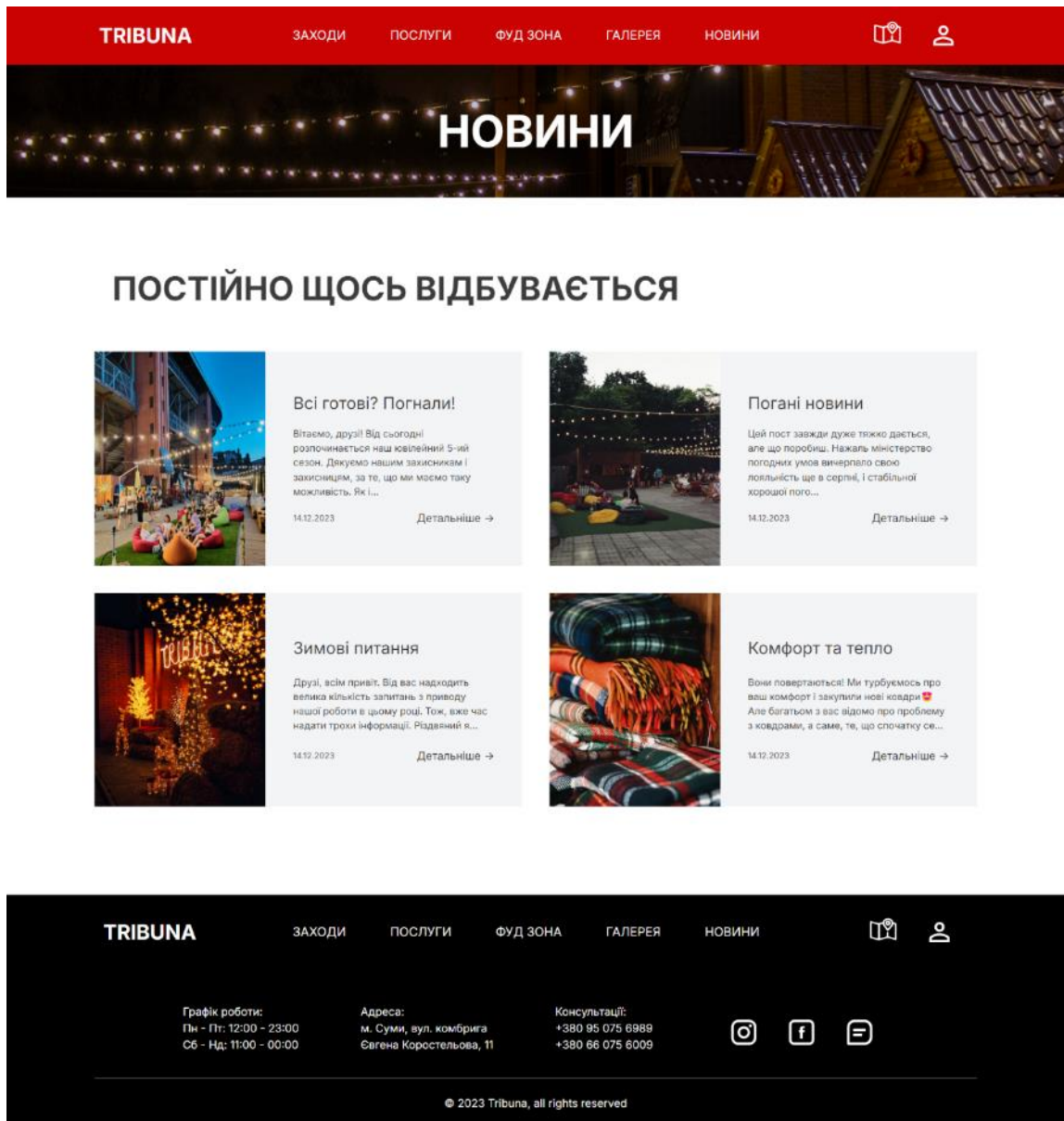


Рисунок 4.9 – Реалізація сторінки новин

Джерело: побудовано автором (знімок з екрану)

Сторінка галерея, представлена на рисунку 4.10.

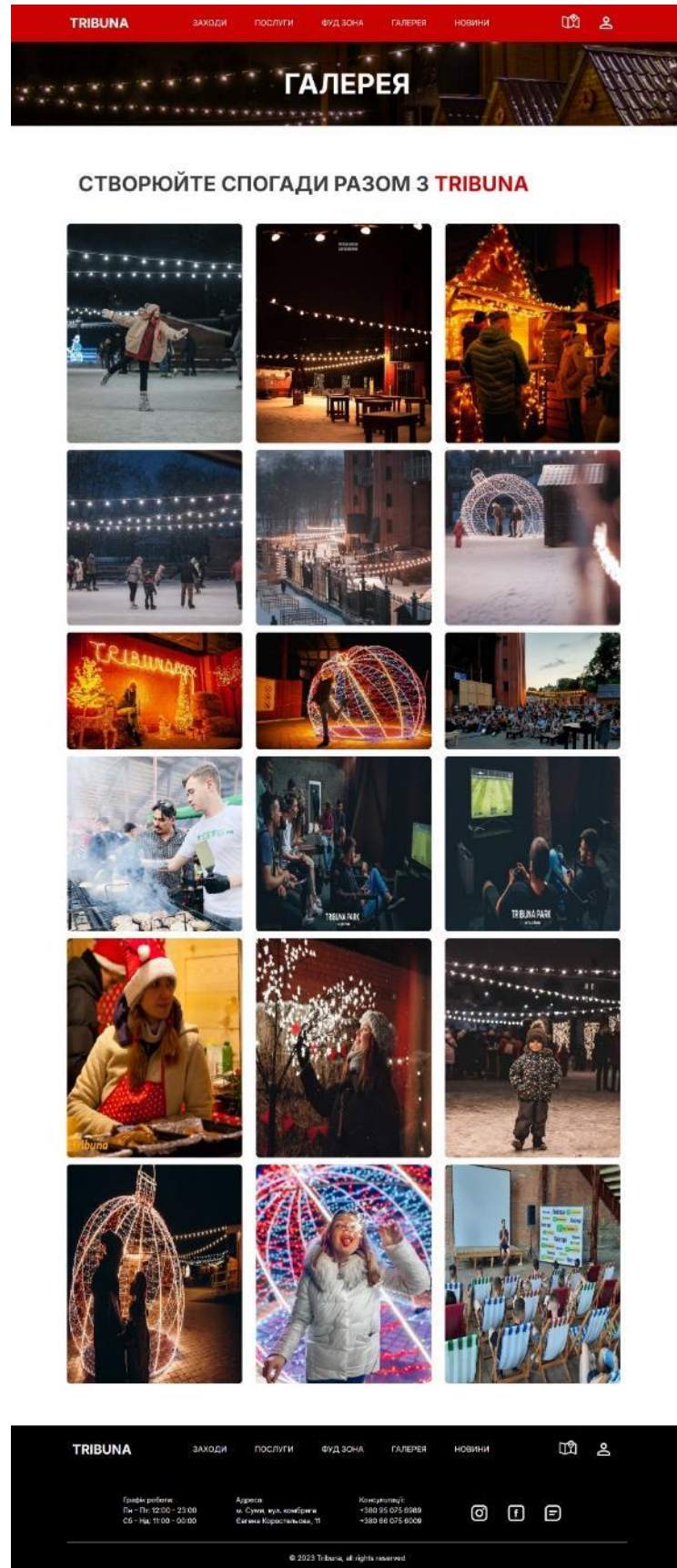


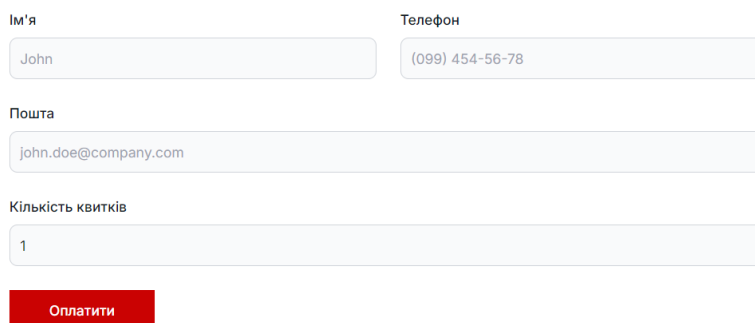
Рисунок 4.10 – Реалізація сторінки галерея
Джерело: побудовано автором (знімок з екрану)

4.3. Тестування web-системи

Після розробки всього необхідного функціоналу, web-система була протестована. Нижче представлені основні функціональні модулі.

Оформлення замовлення квитка на захід (рис. 4.11)

Замовлення квитка

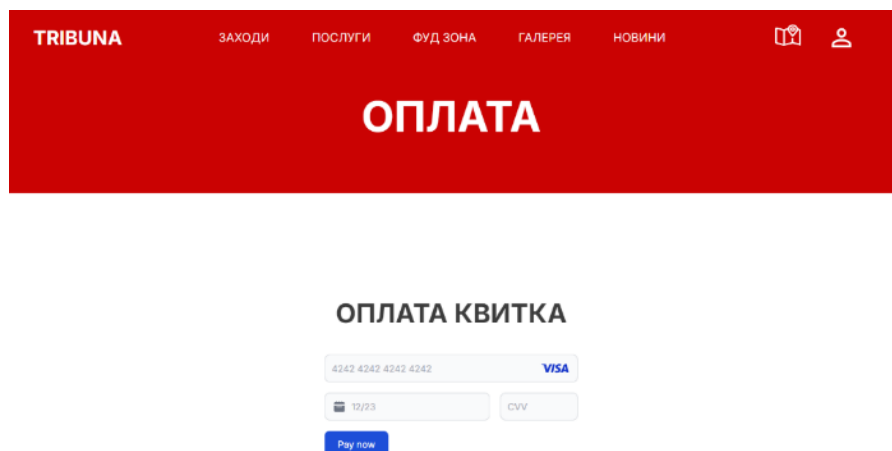


The screenshot shows a web form for booking a ticket. It includes the following fields and elements:

- Ім'я** (Name): Input field with the value "John".
- Телефон** (Phone): Input field with the value "(099) 454-56-78".
- Пошта** (Email): Input field with the value "john.doe@company.com".
- Кількість квитків** (Number of tickets): Input field with the value "1".
- Оплатити** (Pay): A red button located below the form fields.

Рисунок 4.11 – Відображення покупки білету
Джерело: побудовано автором(знімок з екрану)

Оплата замовлення (імітована) (рис. 4.12)



The screenshot shows a payment page for a ticket. It includes the following elements:

- TRIBUNA** logo and navigation menu (ЗАХОДИ, ПОСЛУГИ, ФУД ЗОНА, ГАЛЕРЕЯ, НОВИНИ) at the top.
- ОПЛАТА** (PAYMENT) title in large white text on a red background.
- ОПЛАТА КВИТКА** (PAYMENT TICKET) title below the main header.
- Payment details: A card number "4242 4242 4242 4242" with the **VISA** logo.
- Expiration date "12/23" and CVV field.
- Pay now** button.

Рисунок 4.12 – Відправлення квитка на пошту
Джерело: побудовано автором(знімок з екрану)

Відправка білету на пошту замовнику (представлено на рисунку 4.13)

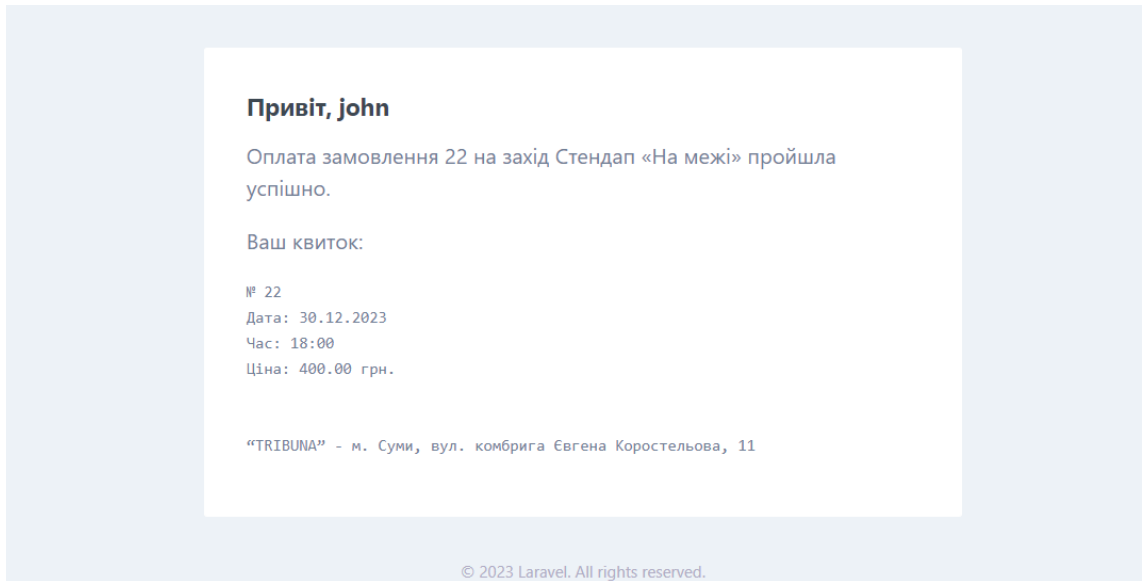


Рисунок 4.13 – Відправлення квитка на пошту
Джерело: побудовано автором(знімок з екрану)

Фільтрація заходів (представлено на рисунку 4.14)

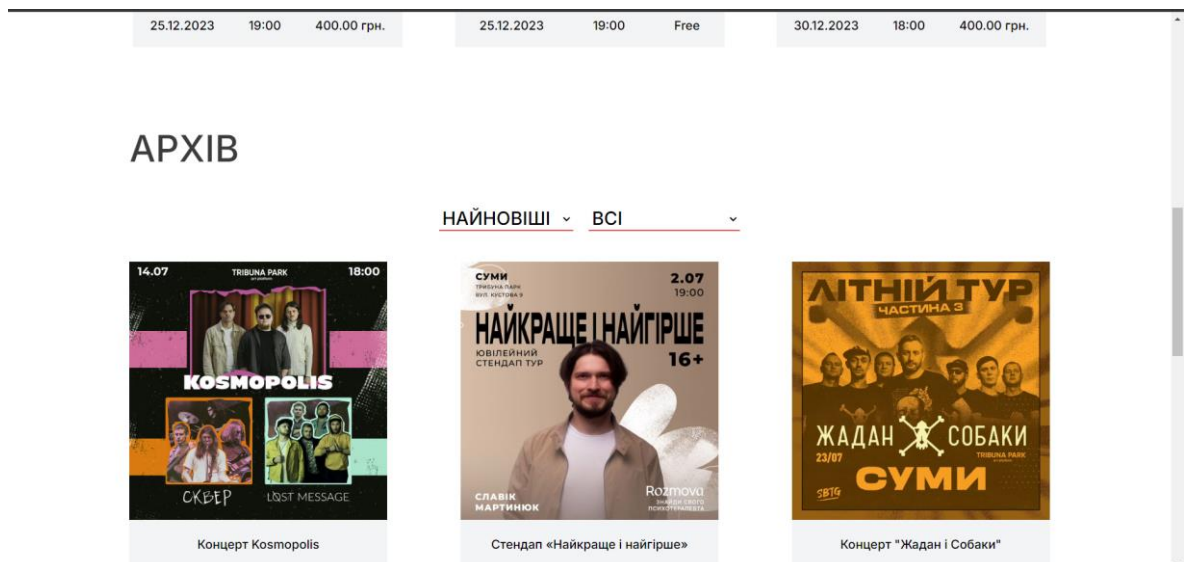


Рисунок 4.14 – Відображення фільтрації заходів
Джерело: побудовано автором (знімок з екрану)

Бронювання спорядження (представлено на рисунку 4.15)



The screenshot shows a reservation form with the following fields and values:

- Ім'я:** John
- Телефон:** (066)994-75-99
- Оберіть час:** 13:00 - 14:00
- Оберіть розмір:** 38
- Кількість:** 1

At the bottom of the form is a red button labeled "Забронювати".

Рисунок 4.15 – Відображення форми для бронювання ковзанів

Джерело: побудовано автором(знімок з екрану)

Адмін-панель для адміністратора (авторизація представлено на рисунку 4.16, сторінка Заходи адмін панелі представлена на рисунку 4.17, сторінка створення нового допису представлена на рисунку 4.18).

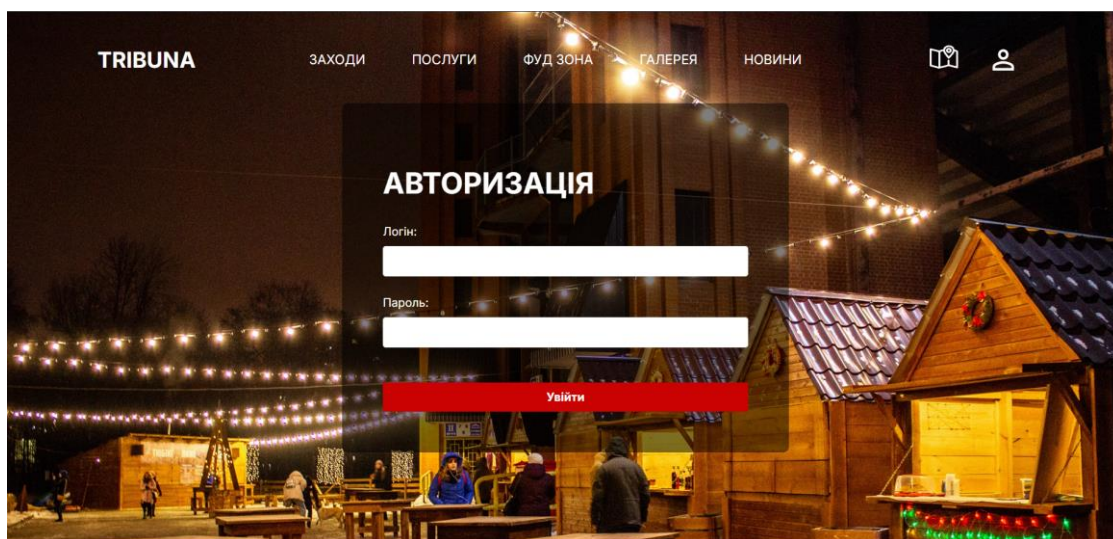


Рисунок 4.16 – Форма авторизації адміністратора

Джерело: побудовано автором (знімок з екрану)

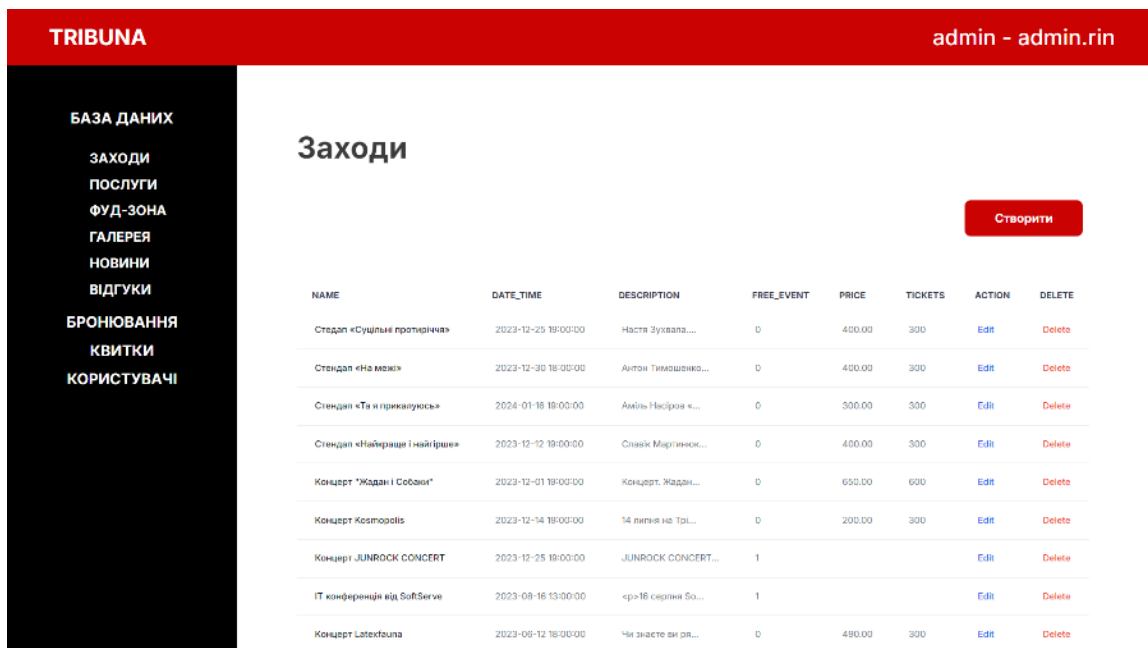


Рисунок 4.17 – Сторінка заходи адмін панелі
Джерело: побудовано автором(знімок з екрану)

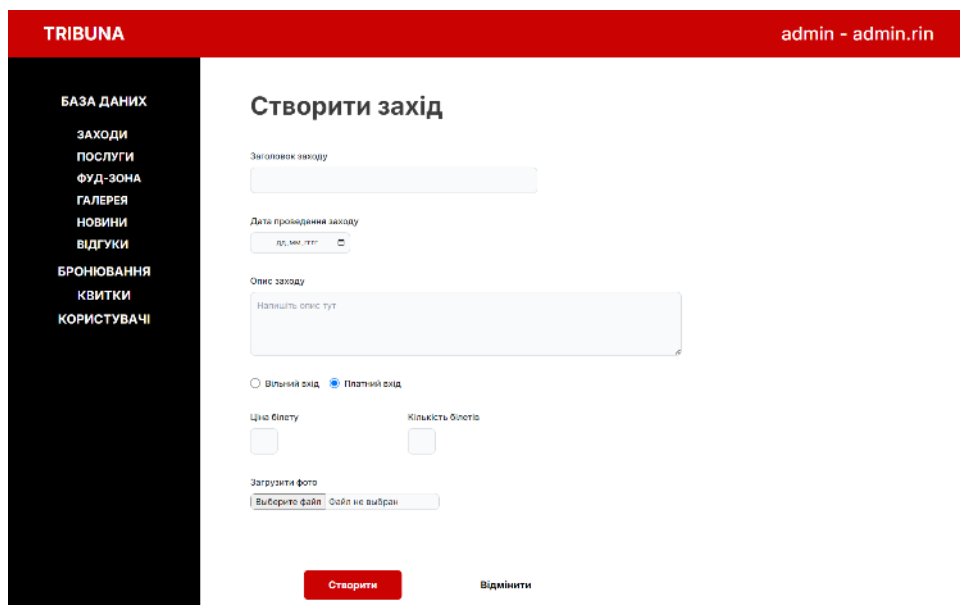


Рисунок 4.18 – Сторінка створення нового заходу адмін панелі
Джерело: побудовано автором(знімок з екрану)

Бу~~в~~но протестований~~ий~~е і інший функціонал, такий як написання відгуків та навігація по інтерактивній мапі (рис. 4.19). А також перевірено роботу посилань на різні сторінки сайту та соці~~альні~~ мережі.

TRIBUNA ЗАХОДИ ПОСЛУГИ ФУД ЗОНА ГАЛЕРЕЯ НОВИНИ

МАПА ТА ВІДГУКИ

НЕ ЗАГУБИТЬСЯ

Каса
Всі додаткові послуги, соціальні блокування та адміністратори знаходяться тут.

[Детальніше](#)

ВАШІ ВІДГУКИ

НАЙНОВІШІ ВСІ

<p>Ірина Послуги</p> <p>Бражася на козачина! Приємна атмосфера та чудовий вибір козачина. Зручний трюк і дружелюбний персонал. Рекомендую!</p> <p>2023-12-14 00:55:58</p>	<p>Валентина Заходи</p> <p>Тематичні коктейлі як у Трибуні - круті! Фудкорний трюк, міні-середки - завжди є, завжди с'яє, завжди. Пам'ятаю місце для відвідувачів з дитинства!</p> <p>2023-12-14 22:01:44</p>
<p>Микола Послуги</p> <p>Косоріг у Трибуні - ідеальне місце для приватності. Зручне робоче середовище, чудовий Wi-Fi - на крайній межі зручності. Завдяки!</p> <p>2023-12-14 20:39:38</p>	<p>g1234 Заходи</p> <p>Світлодіодна дельта народилася у Трибуні - і всі мрії є! Дуже круто, завжди є, і завжди є. Дякую за незабутній вечір!</p> <p>2023-12-14 23:20:13</p>
<p>Дмитрий Фудкор</p> <p>чужий</p> <p>2023-12-15 06:48:30</p>	

ПОДІЛІТЬСЯ ВРАЖЕННЯМИ ПРО TRIBUNA

Ім'я

Обрати категорію відгуку

Область інтересу

Ваша оцінка

Надіслати

TRIBUNA ЗАХОДИ ПОСЛУГИ ФУД ЗОНА ГАЛЕРЕЯ НОВИНИ

Графік роботи: Пн - Пн 12:00 - 23:00, Сб - Нд 12:00 - 00:00

Адреса: м. Суми, 901, коомплага Солона Коростельська, 11

Контактний дп: +380 96 072 8383, +380 96 010 0098

© 2023 Tribuna. All rights reserved.

Рисунок 4.19 – Сторінка з мапою та відгуками
Джерело: побудовано автором (знімок з екрану)

ВИСНОВКИ

У процесі реалізації дипломного проекту створено web-систему підтримки діяльності розважального комплексу «Трибуна» та звіт до неї, який включає чотири розділи та два додатки. Перед тим, як розпочати розробку web-системи, був проведений глибокий аналіз предметної області. У ході аналізу визначено основні цілі проекту, а також його актуальність на основі порівняння з існуючими системами.

Були виконані всі поставлені задачі дипломної роботи:

- Проведено аналізу актуальних аспектів обраної предметної області з метою визначення сучасних тенденцій, викликів та можливостей, що можуть бути використані у контексті досліджуваного дипломного проекту.
- Спроектовано та змодельовано об'єкт дослідження з метою створення концептуальної та технічної основи для подальшої імплементації.
- Розроблено дизайн-макету системи для візуалізації структури web-системи та її представлення.
- Розроблено та протестовано web-систему підтримки діяльності для розважального комплексу.

Додатково, розглянуто потенційні загрози та визначено стратегії управління ними в контексті даного комплексу. Всі ці етапи сприяли створенню високоякісного та ефективного проекту, який може значно покращити якість обслуговування та управління розважальним комплексом "Трибуна".

Результатом є готова web-системи підтримки діяльності розважального комплексу "Трибуна", що задовольняє визначеним функціональним вимогам та відповідає потребам кінцевого користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gonzalez R. C. and Woods R. E. "Digital Image Processing." Prentice Hall, 2008. 976 p. (дата звернення: 10.10.2023).
2. Prince S. "Computer Vision: Models, Learning, and Inference." Cambridge University Press, 2012. 665 p. (дата звернення: 10.10.2023).
3. Li Y., Yang Z., Chen X., Yuan H., Liu W. "A stacking model using URL and HTML features for phishing webpage detection." Future generations computer systems, 2019. 15 p. (дата звернення: 10.10.2023).
4. Shykula O. M. "Development of WEB-site of goods store." Scientific Notes of the State University of Telecommunications, 2021. 10 p. (дата звернення: 10.10.2023).
5. Hess D. "Engineering outreach web page." Proceedings Frontiers in Education 1997 27th Annual Conference. Teaching and Learning in an Era of Change, 1997. 8 p. (дата звернення: 17.10.2023).
6. Ballard D. H. and Brown C. M. "Computer Vision." Prentice Hall, 1982. 322 p. (дата звернення: 17.10.2023).
7. Zubaer A., Mondal S. K., Islam M., Hossain M. A., Naim M. M. H., et al. "Design and Development a Website using HTML, CSS, PHP and MySQL." Future generations computer systems, 2020. 12 p. (дата звернення: 17.10.2023).
8. Rifandi F., Adriansyah T. V., Kurniawati R. "Website Gallery Development Using Tailwind CSS Framework." Jurnal E-Komtek, 2022. 8 p. (дата звернення: 26.10.2023).
9. Oliver N., Rosario B., Pentland A. "A Bayesian Computer Vision System for Modeling Human Interactions." MIT Press, 2000. 240 p. (дата звернення: 26.10.2023).
10. Samadi-Parviznejad P. "Simulation of Sales Scenarios in Chain Store Marketing with a Futuristic Approach." International journal of Innovation in Marketing Elements, 2021. 12 p. (дата звернення: 26.10.2023).
11. Coulson L., Jephson B., "The HTML and CSS Workshop." 2019. 608 p. (дата звернення: 26.10.2023).

12. S S., Giridharadhayalan M. "Web Application Using HTML, CSS, JavaScript and Java." *International Journal of Innovative Research in Engineering*, 2023. 10 p. (дата звернення: 01.11.2023).
13. Qian N., Wang J., Mueller F., Bernard F., Golyanik V., et al. "HTML: A Parametric Hand Texture Model for 3D Hand Reconstruction and Personalization." *European Conference on Computer Vision*, 2020. 16 p. (дата звернення: 26.10.2023).
14. Sheeba T., Begum S. "Comparative Study of Developing Interactive Multimedia Applications using Adobe Flash and HTML/CSS." *International Journal of Advanced Research in Computer Science and Electronics Engineering*, 2018. 6 p. (дата звернення: 26.10.2023).
15. Pan L., Ma J. "HTML+CSS Implementation based on Image Intelligent Scene Recognition Algorithm." *International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2022. 12 p. (дата звернення: 26.10.2023).
16. Jafarov J. "Online Marketing for a New Opening Web Store." *Business*, 2017. 28 p. (дата звернення: 01.11.2023).
17. Nurninawati E., Effendy M. Y., Rianputra A. M. "Web-Based Product Marketing Information System Design at Definier Store." *International Journal of Cyber and IT Service Management*, 2022. 20 p. (дата звернення: 01.11.2023).
18. Буковель Bukovel. Всесезонний курорт у серці Карпат. Буковель. Гірськолижний та SPA курорт в Карпатах. URL: <https://bukovel.com> (дата звернення: 16.11.2023).
19. Готелі Києва: Готель Турист у Києві, Україна. Готелі Києва: Готель Турист у Києві, Україна. URL: <https://hotel-tourist.kiev.ua> (дата звернення: 16.11.2023).
20. Дитячий розважальний центр Чубі Бум. Піцерія Чубі Бум. URL: <https://chubi-boom.com> (дата звернення: 16.11.2023).
21. Мельник Р.А. Програмування web-застосувань. Львівська політехніка, 2018. 248 с. (дата звернення: 20.11.2023).
22. Берко А. Ю., Верес О. М. Організація баз даних: практичний курс: Навч. посіб. для студ. Нац. ун-т «Львів. політехніка». Л., 2003. 149 с.

23. Гайна Г. А. Основи проектування баз даних : навч. посіб. для студ. вищ. навч. закл., що навчаються за напрямом підгот. 0804 "Комп'ютерні науки". Київ : Кондор, 2008. 200 с. (дата звернення: 20.11.2023).
24. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. Х: ХНАМГ, 2010. 222 с. (дата звернення: 20.11.2023).
25. Documentation - Tailwind CSS. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. URL: <https://v2.tailwindcss.com/docs> (дата звернення: 27.11.2023).
26. Laravel - The PHP Framework For Web Artisans. Laravel - The PHP Framework For Web Artisans. URL: <https://laravel.com/docs/10.x> (дата звернення: 27.11.2023).
27. What is Laravel and what is it useful for?. Rootstack | Web, Mobile & Automation solutions agency | Rootstack. URL: <https://rootstack.com/en/learning/what-laravel-and-what-it-useful> (дата звернення: 27.11.2023).
28. Why Tailwind? A long term user perspective. ecklf.com. URL: <https://ecklf.com/blog/why-tailwind> (дата звернення: 27.11.2023).
29. Що таке база даних? | Кафедра АПЕПС ТЕФ КПІ – програмна інженерія. Інженерія програмного забезпечення та комп'ютерні науки в КПІ | Кафедра АПЕПС ТЕФ КПІ – програмна інженерія. URL: <https://apeps.kpi.ua/shco-take-basa-danykh> (дата звернення: 27.11.2023).
30. Типи баз даних: особливості, відмінності та приклади | DOU - Спільнота програмістів. URL: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення: 27.11.2023).

ДОДАТОК А ПЛАНУВАННЯ РОБІТ

А.1 Ідентифікація мети ІТ-проекту

В рамках даного дослідження розглядається задача ідентифікації мети ІТ-проекту, що стосується створення web-системи для підтримки діяльності розважального комплексу "Трибуна". З метою забезпечення якісної реалізації проекту та визначення його успішності в майбутньому, важливо визначити мету і завдання ініційованого ІТ-проекту методом SMART. Наступна таблиця А1.1 відображає деталізацію мети системи методом SMART.

Таблиця А1.1 – Деталізація мети методом SMART

Джерело: побудовано автором

<p>Specific (конкретна)</p>	<p>Мета проекту полягає в розробці та впровадженні web-системи для підтримки діяльності розважального комплексу "Трибуна". Ця система повинна забезпечувати автоматизоване управління ресурсами, бронюванням та обслуговуванням клієнтів, покращуючи ефективність та задоволення клієнтів.</p>
<p>Measurable (вимірювана)</p>	<p>Проект буде вимірюватися за допомогою ключових показників, включаючи підвищення обсягів продажів, покращення рівня задоволення клієнтів, зменшення рутинних операцій та підвищення продуктивності персоналу. Всі ці показники будуть кількісно вимірювані та аналізовані для оцінки успішності проекту.</p>
<p>Achievable (досяжна)</p>	<p>Розробка та впровадження web-системи для "Трибуни" вважається досяжною завданням, оскільки вони ґрунтуються на сучасних технологіях і методологіях розробки програмного забезпечення. Команда програмістів, які беруть участь у проекті, має необхідний досвід та навички для виконання завдань.</p>

Продовження таблиці А1.1

Relevant (реалістична)	Розробка web-системи для розважального комплексу "Трибуна" відповідає стратегічним цілям компанії і відповідає потребам сучасного ринку розважальних послуг. Проект реалістичний і має потенціал для забезпечення значущих переваг для компанії.
Time-framed (обмежена у часі)	Проект має чітку обмежену у часі рамку для виконання. Розробка та впровадження web-системи мають завершитися протягом 6 місяців з моменту старту проекту. Цей часовий каркас дозволить забезпечити своєчасне впровадження та почати збирати плоди проекту.

Розробка імплементація Web-системи підтримки діяльності розважального комплексу "Трибуна" - завдання великої складності, яке потребує ретельного розгляду та визначення мети IT-проекту. Визначення цієї мети - ключовий етап, який визначить подальший успіх проекту та ефективність його реалізації.

Мета проекту "Web-система підтримки діяльності розважального комплексу "Трибуна" має бути ретельно деталізована з використанням методу SMART (Specific, Measurable, Achievable, Relevant, Time-bound). Перш за все, мета повинна бути конкретною та специфікованою, що включає в себе розробку імплементацію web-системи для оптимізації управління та автоматизації різних аспектів діяльності розважального комплексу "Трибуна".

Далі, мета повинна бути вимірюваною. Це означає, що метрики успіху повинні бути встановлені для оцінки результатів проекту. Ці метрики можуть включати в себе час відповіді на запити, кількість оброблених замовлень, зменшення помилок та інші параметри, які дозволять об'єктивно виміряти ефективність системи.

Мета також повинна бути досяжною. Це означає, що проект повинен бути реалістичним, враховуючи обмеження ресурсів, доступність технологій та терміни виконання. Визначення реальних очікувань та можливостей допомагає уникнути надмірно оптимістичних цілей, які не можуть бути досягнуті в рамках проекту.

Мета повинна бути також релевантною для діяльності розважального комплексу "Трибуна". Це означає, що система повинна відповідати потребам та завданням комплексу, сприяючи його оптимізації та покращенню управління різними аспектами бізнесу.

Нарешті, мета повинна бути часово обмеженою. Визначення чіткого графіку виконання проекту дозволяє забезпечити його своєчасну реалізацію та визначити терміни досягнення поставлених цілей.

Отже, мета ІТ-проекту "Web-система підтримки діяльності розважального комплексу "Трибуна" повинна бути конкретною, вимірюваною, досяжною, релевантною та часово обмеженою. Визначення такої мети є важливим кроком у визначенні курсу розробки та імплементації проекту для досягнення успішних результатів.

Цей підход до ідентифікації мети проекту та планування розробки допоможе забезпечити ефективне виконання ІТ-проекту та досягнення поставлених цілей.

А.2 Планування змісту структури робіт ІТ-проекту

Планування змісту структури робіт ІТ-проекту (WBS) для розробки web-системи підтримки діяльності розважального комплексу "Трибуна" - це складний і технічно вимогливий процес, який передбачає розподіл завдань та визначення ієрархії робіт, необхідних для успішної реалізації проекту. Варто почати з розбиття проекту на основні фази та підзавдання, що відповідають ключовим функціональним вимогам системи "Трибуна".

Першою фазою в проекті є аналіз вимог. У цьому кроці спеціалісти повинні провести детальний аналіз потреб розважального комплексу, визначити вимоги до функціональності системи, технічні вимоги, а також параметри продуктивності та безпеки. Ця фаза передбачає роботу бізнес-аналітиків та системних архітекторів.

Другою фазою є проектування системи. На цьому етапі ведеться розробка архітектури, визначення технологічного стеку, розробка схем баз даних, інтерфейсів користувача, а також розробка плану тестування та забезпечення безпеки.

Третьою фазою є реалізація програмного забезпечення. Програмісти виконують розробку, кодування та інтеграцію системи, створюючи функціональні компоненти та забезпечуючи їх взаємодію.

Четвертою фазою є тестування та верифікація. Тут проводяться функціональні, інтеграційні та валідаційні тести для перевірки якості програмного забезпечення. Також виконується верифікація відповідності системи вимогам.

П'ятою фазою є впровадження та підтримка. На цьому етапі система "Трибуна" впроваджується в реальному середовищі розважального комплексу. Технічна підтримка, навчання персоналу та моніторинг роботи системи входять в обов'язки ІТ-спеціалістів.

Кожна з цих фаз може бути розгалужена на підзавдання, які розподіляються серед відповідних команд і фахівців. Такий підхід дозволяє забезпечити систематичний і керований процес розробки, зменшити ризики та впевнитися у

виконанні всіх завдань в обговорені строки. Рисунок А2.1 містить відображення WBS таблиці системи.

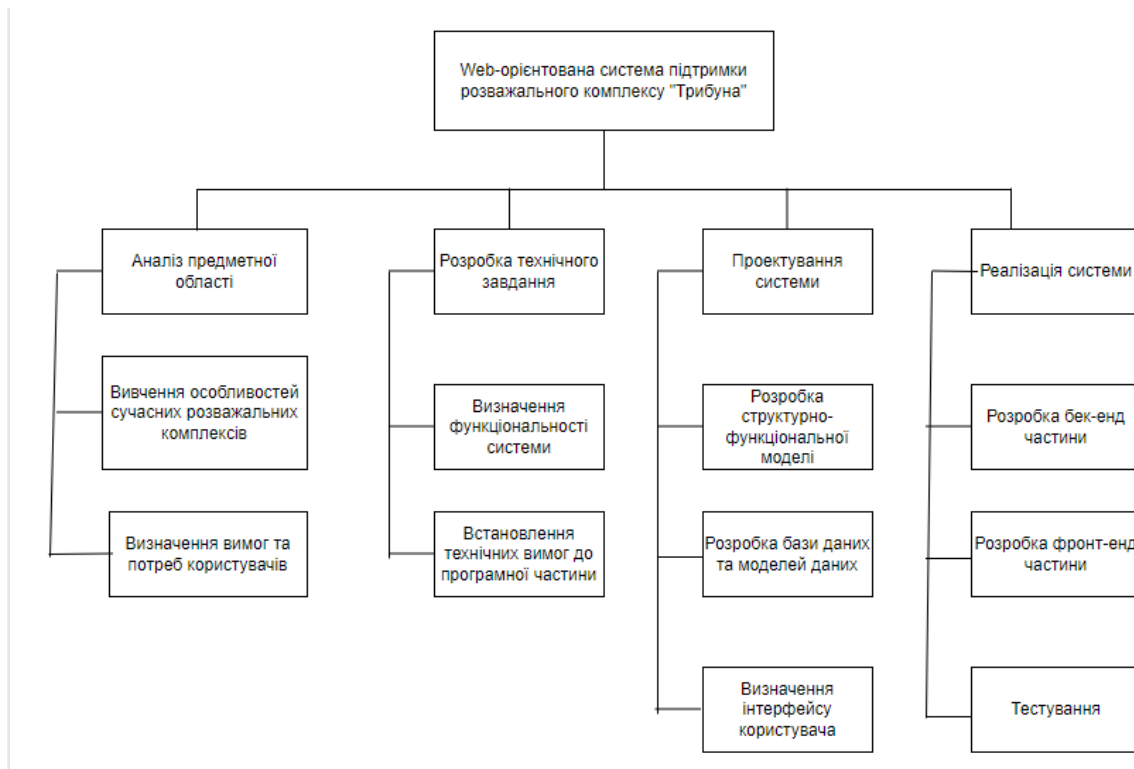


Рисунок А2.1 – WBS таблиця системи

Джерело: побудовано автором

В рамках розвитку проекту "Web-система підтримки діяльності розважального комплексу "Трибуна" не менш важливою, ніж технічний аспект, є організаційна структура проекту, або OBS (Organizational Breakdown Structure). Цей елемент проекту дозволяє визначити і структурувати ключові функції, ролі та відповідальності в контексті реалізації web-системи, яка має забезпечити ефективну функціональність та підтримку роботи розважального комплексу.

Здійснення досліджень та розробка "Web-системи підтримки діяльності розважального комплексу "Трибуна" вимагає інтеграції зі структурами компанії, що вже існують. Організаційна структура проекту повинна бути спроектована з урахуванням специфіки комплексу, а також інновацій, які впроваджуються.

На початковому етапі, необхідно виділити команду проекту, включаючи роль проектного менеджера, архітектора системи, розробника програмного

забезпечення, тестувальника, та інших фахівців з обов'язками, що відображають їхню експертизу та функціональний вклад.

Проектний менеджер відповідає за управління проектом в цілому, планування ресурсів, встановлення термінів і моніторинг виконання завдань. Архітектор системи визначає архітектурні рішення, інтеграцію з існуючими системами та загальну архітектурну відповідність. Розробник програмного забезпечення відповідає за створення програмного коду, що реалізує функціональність системи. Тестувальник перевіряє якість та стабільність системи через відповідні тести.

Організаційна структура проекту також включає в себе інші ключові ролі та функції, такі як аналітик, дизайнер і адміністратор бази даних. Усі ці фахівці взаємодіють і співпрацюють для досягнення спільних цілей проекту. Рисунок A2.2 містить відображення OBS таблиці системи.

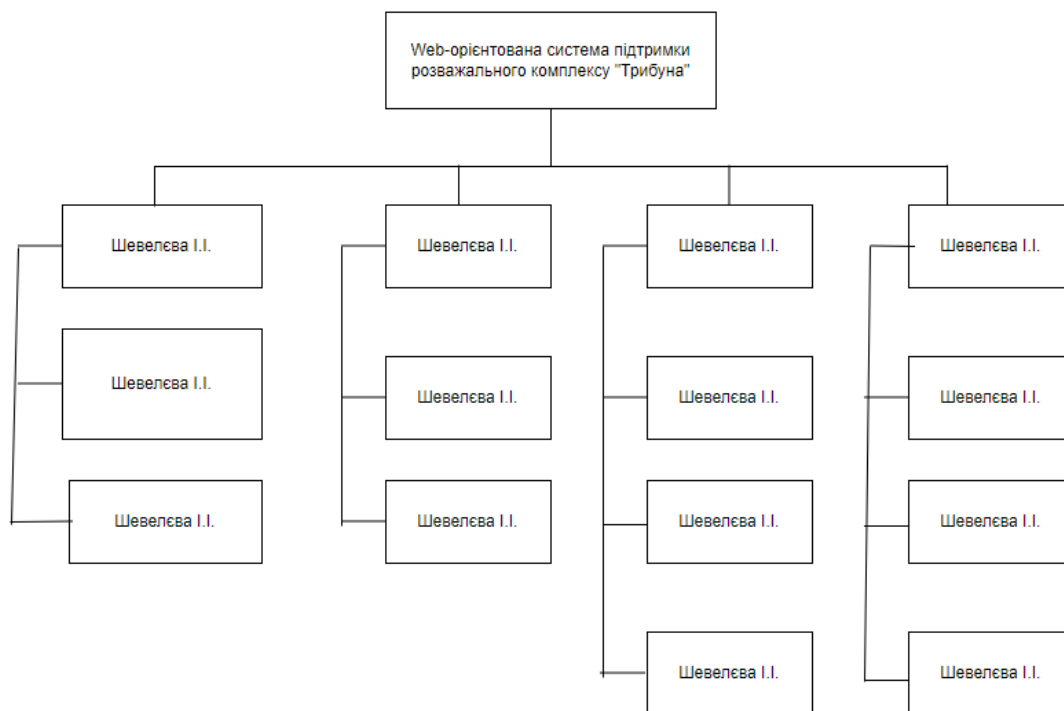


Рисунок A2.2 – OBS таблиця системи

Джерело: побудовано автором

Згідно з принципами OBS, структура проекту має бути гнучкою та відкритою для змін. Вона повинна відображати реалії та специфіку проекту, а також

враховувати внутрішні та зовнішні фактори, що можуть вплинути на процес виконання. Ретельне планування та розробка OBS допомагають забезпечити ефективний контроль над проектом та його успішне завершення.

Загалом, організаційна структура проекту "Web-система підтримки діяльності розважального комплексу "Трибуна" є ключовим елементом для успішної реалізації проекту, оскільки вона допомагає визначити ролі та відповідальності учасників, забезпечуючи координацію та співпрацю всіх сторін.

А.3 Побудова календарного графіку виконання ІТ – проекту

Побудова календарного графіка виконання ІТ-проекту - це важливий етап у розробці системи підтримки діяльності розважального комплексу "Трибуна". Даний етап вимагає від розробників відданості, точності та фаховості, оскільки календарний графік є основою для успішної реалізації проекту.

Перед початком побудови календарного графіка необхідно провести аналіз всіх складових проекту. Цей аналіз має включати в себе оцінку обсягу робіт, визначення ключових завдань, ідентифікацію залежностей між завданнями та визначення ресурсів, необхідних для виконання кожного етапу проекту.

Для ефективною побудови календарного графіка слід використовувати спеціальні інструменти та методи, такі як методологія керування проектами, робочі розклади та сучасне програмне забезпечення для управління проектами. Таблиця А3.1 створена для того, щоб правильно розподілити час та умови виконання в створенні системи.

Таблиця А3.1 – Календарний графік виконання

Джерело: побудовано автором

Задача	Час (дні)	Виконання
Аналіз вимог	1	Програміст – Шевелєва І.І.
Проектування системи	15	Архітектор – Шевелєва І.І.
Розробка інтерфейсу	25	Дизайнер – Шевелєва І.І.
Розробка бази даних	15	Програміст – Шевелєва І.І.
Розробка функціональності	40	Програміст – Шевелєва І.І.
Тестування і налагодження	4	Тестувальник – Шевелєва І.І.
Завершення проекту	1	Менеджер – Шевелєва І.І.

Для таблиці, що подано вище, представимо діаграму Ганта (рис.А3.1).

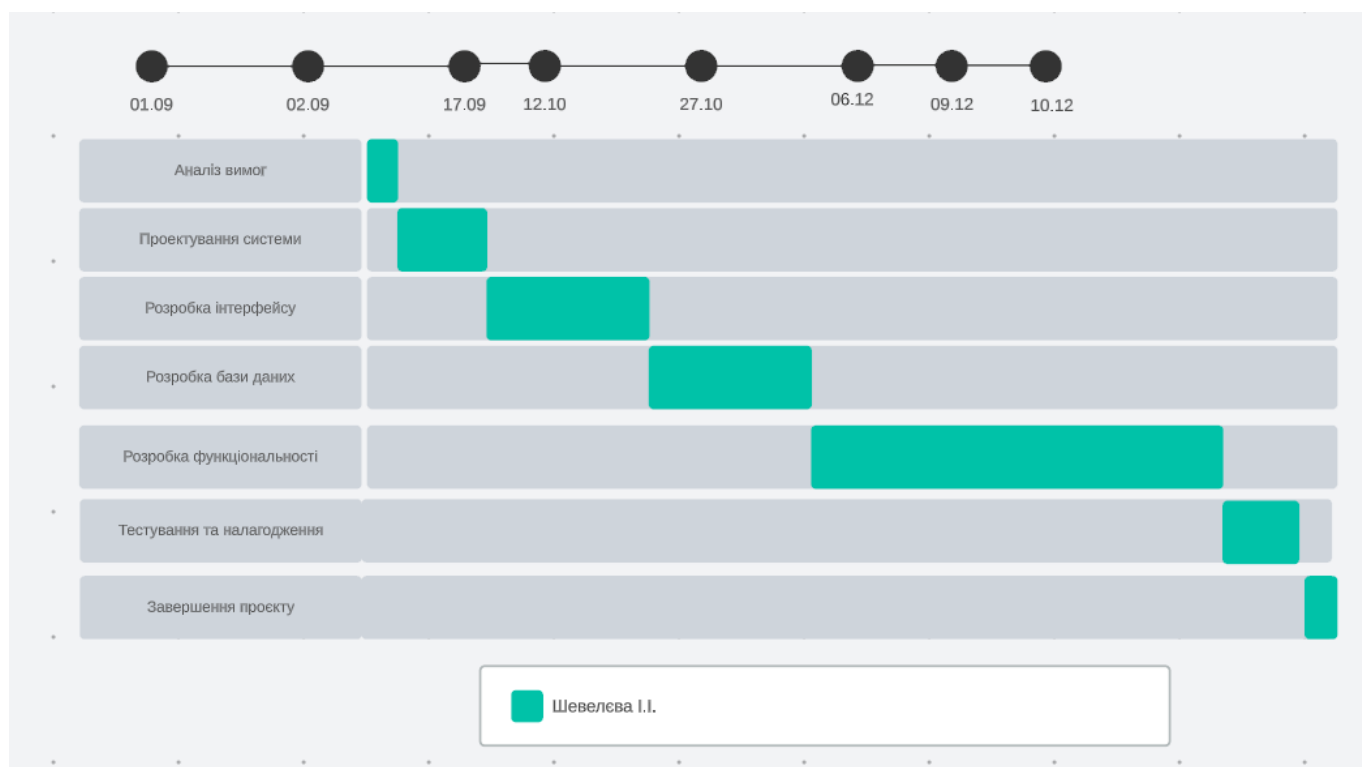


Рис. А3.1 – Діаграма Ганта проекту

Джерело: побудовано автором

Розглянемо детально кожну задачу та виділені значення часу:

1. Аналіз вимог (1 день): Аналіз вимог є початковим етапом проекту, і він вимагає глибокого розуміння потреб користувачів і завдань системи. Оскільки ця задача важлива і вимагає уваги до деталей, обрано 2 тижні для її виконання.

2. Проектування системи (15 днів): Проектування системи включає в себе створення архітектурної концепції та визначення структури проекту. Ця задача важлива для успішної подальшої розробки, тому виділено 4 тижні для глибокого проектування системи. Виконання цієї задачі доручено архітектору, який має глибокі знання в галузі архітектури програмного забезпечення.

3. Розробка інтерфейсу (25 днів): Дизайн інтерфейсу графічного користувача (UI) важливий для забезпечення зручності користувачів системи. Для розробки якісного та привабливого інтерфейсу відводиться 3 тижні. Виконання цієї задачі доручено дизайнеру, який має експертизу у графічному дизайні.

4. Розробка бази даних (15 днів): Розробка бази даних важлива для забезпечення надійного зберігання даних системи. Для цієї задачі виділено 2 тижні, і вона виконується програмістом, який спеціалізується на роботі з базами даних.

5. Розробка функціональності (40 днів): Розробка функціональності системи – це основний етап, на якому програмісти створюють функціональність системи. Для забезпечення якості та швидкості виконання цієї задачі виділено 6 тижнів.

6. Тестування і налагодження (4 дні): Тестування системи та виправлення виявлених помилок є важливим етапом перед впровадженням. Для цього виділено 2 тижні, і ця задача виконується тестувальником.

7. Завершення проекту (1 день): Останній тиждень проекту відведено на завершення всіх документів, підготовку до впровадження та остаточну перевірку системи перед використанням. Виконавцем цієї задачі є менеджер.

Обрані строки та виконавці відповідають специфіці проекту, його обсягу та важливості різних етапів. Вони дозволяють оптимально розподілити ресурси та забезпечити успішну реалізацію проекту, дотримуючись рекомендацій з керування проектами та розробки програмного забезпечення.

Згідно з розробленим календарним графіком, визначаються точні строки початку та завершення кожного завдання, а також встановлюються терміни для мілістоунів та важливих подій в проекті.

Календарний графік також враховує фактори ризику та можливість затримок у виконанні завдань. Для цього розробники повинні аналізувати можливі загрози та розробляти стратегії для їх управління.

Після побудови календарного графіка, важливо систематично відстежувати виконання проекту та вчасно коригувати графік в разі виникнення непередбачуваних обставин.

Заключний календарний графік є ключовим інструментом для успішної реалізації IT-проекту, оскільки він дозволяє уникнути затримок та перебоїв у роботі, що може суттєво вплинути на результати проекту. Тому побудова календарного графіка вимагає особливої уваги та фаховості з боку розробників.

А.4 Планування ризиків проекту

В ході розробки плану ризиків для IT-проекту " Web-система підтримки діяльності розважального комплексу "Трибуна"" ідентифікуються та оцінюються можливі загрози та приймаються стратегії управління ними.

Таблиця А.4.1 – Ризики проекту

Джерело: побудовано автором

Ризик	Назва ризику	Опис ризику
1	Зміни вимог	Внесення змін у вимоги може виникнути внаслідок непорозумінь або зміни потреб замовника.
2	Технічні труднощі	Розробка може зіткнутися з технічними проблемами або несподіваними труднощами,

		зокрема пов'язаними із вибором технологій чи інтеграцією.
--	--	---

Продовження таблиці А4.1

3	Затримка в розробці	Затримка через непередбачені обставини або низьку продуктивність виконавця.
4	Проблема з програмним забезпеченням	Несправність у вигляді порушення роботи програмного забезпечення.
5	Відсутність резервних копій	Невиконання процесу резервного копіювання. Відсутність збережень резервних копій на різних носіях.
6	Виконання додаткових функцій	Реалізація додаткових непотрібних функцій.
7	Людський фактор	Зниження продуктивності через неуважність.
8	Зміна терміну виконання робіт	Зміна календарного плану виконання робіт.
9	Недостатня кваліфікація виконавця	Брак досвіду та знань, що впливає на продуктивність роботи

Таблиця А.4.2 – Реєстрація ризиків

Джерело: побудовано автором

Ризик	Вірогідність виникнення	Ступінь впливу	Значення ризику
1	3	3	8
2	2	2	5
3	2	2	4
4	1	3	8
5	1	2	6
6	1	1	4
7	2	2	3

8	2	1	5
9	2	1	4

Таблиця А.4.3 – Матриця впливу

Джерело: побудовано автором

Вірогідність виникнення	Матриця впливу		
3			1
2	8, 9	2, 3, 7	
1	6	5	4
Ступінь впливу	1	2	3

Таблиця А.4.4 – Матриця впливу

Джерело: побудовано автором

Ризик	Назва ризику	Реакція на ризик
1	Зміни вимог	Узгоджувати кожний етап із замовником.
2	Технічні труднощі	Проаналізувати технології перед реалізацією проекту.
3	Затримка в розробці	Створити календарний план та дотримуватись його.
4	Проблема з програмним забезпеченням	Налаштувати програмне забезпечення, в іншому випадку замінити його.
5	Відсутність резервних копій	Налаштувати автоматичне копіювання. Зберігати резервні копії на носіях.
6	Виконання додаткових функцій	Створити та узгодити план із замовником та чітко дотримуватися його.
7	Людський фактор	Робити на кожному етапі перевірки, для запобігання помилок.
8	Зміна терміну виконання робіт	Переговорити із замовником та зменшити кількість додаткових функцій.
9	Недостатня кваліфікація виконавця	Пройти курси для підвищення кваліфікації.

ДОДАТОК Б ЛІСТИНГ КОДУ

Б.1 Лістинг файлу web.php

```

<?php

use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your
application. These
| routes are loaded by the RouteServiceProvider and all of them
will
| be assigned to the "web" middleware group. Make something
great!
|
*/

Route::get('/',
'\App\Http\Controllers\PageController@mainPage')->name('main');

Route::get('/events',
'App\Http\Controllers\PageController@eventPage' )-
>name('events.index');

Route::get('/services',
'\App\Http\Controllers\PageController@servicePage')-
>name('services.index');

Route::get('/food_zones',
'\App\Http\Controllers\PageController@foodZonePage')-
>name('food_zones.index');

Route::get('/gallery',
'\App\Http\Controllers\PageController@galleryPage')-
>name('gallery');

```



```

Route::get('/news',
'\App\Http\Controllers\PageController@newsPage')-
>name('news.index');

Route::get('/map',
'\App\Http\Controllers\PageController@mapPage')->name('map');

Route::get('/send',
'\App\Http\Controllers\MailController@send')->name('send');

Route::get('/get-events',
'\App\Http\Controllers\EventController@getEvents')->name('get-
events');

Route::get('/events/{event}',
'App\Http\Controllers\EventController@show' )->name('events.show');

Route::post('/ticket_init','App\Http\Controllers\EventControlle
r@ticket_init')->name('ticket_init');

Route::post('/payment/{bookingEvent}',
'App\Http\Controllers\EventController@init_payment' )-
>name('init_payment');

Route::get('/news/{news}',
'App\Http\Controllers\NewsController@show' )->name('news.show');

Route::get('/services/{service}',
'App\Http\Controllers\ServiceController@show' )-
>name('services.show');

Route::get('/food_zones/{food_zone}',
'App\Http\Controllers\FoodZoneController@show' )-
>name('food_zones.show');

Route::get('/admin/login',
'App\Http\Controllers\Admin\AuthController@index' )-
>name('admin.login');

```

```

Route::post('/admin/login_process',
'App\Http\Controllers\Admin\AuthController@login' )-
>name('admin.login_process');

Route::post('/admin/logout',
'App\Http\Controllers\Admin\AuthController@logout' )-
>name('admin.logout');

Route::get('/admin/main',
'App\Http\Controllers\Admin\AuthController@main' )-
>name('admin.main');

Route::post('/booking_tubing',
'App\Http\Controllers\ServiceController@booking_tubing' )-
>name('services.booking_tubing');

Route::post('/booking_skating',
'App\Http\Controllers\ServiceController@booking_skating' )-
>name('services.booking_skating');

```

Б.2 Лістинг файлу EventController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\BookingEvent;
use App\Models\Event;
use Carbon\Carbon;
use Illuminate\Http\Request;

class EventController extends Controller
{
    public function getEvents(Request $request)
    {
        $sort = $request->input('sort');

```

```
|  
  
$type = $request->input('type');  
  
$now = Carbon::now();  
$events = Event::where('date_time', '<', $now);  
  
if ($type === '2') {  
    $events->where('free_event', 1);  
} elseif ($type === '3') {  
    $events->where('free_event', 0);  
}  
  
if ($sort === '1') {  
    $events->orderBy('date_time', 'desc');  
} else {  
    $events->orderBy('date_time');  
}  
  
$events = $events->get();  
  
return response()->json($events);  
}  
  
public function show(Event $event){  
  
    return view('events.show', compact('event'));  
}  
  
public function ticket_init(){  
  
    $user_name = request()->input('user_name');  
    $event_id = request()->input('event_id');  
    $phone = request()->input('phone');
```

```
|
$email = request()->input('email');
$tickets_count = request()->input('tickets_count');
$status = request()->input('status');

$data=array(
    'user_name'=>$user_name,
    "event_id"=>$event_id,
    "phone"=>$phone,
    "mail"=>$email,
    "count"=>$tickets_count,
    "status"=>$status
);

BookingEvent::create($data);
$bookingEvent = BookingEvent::where('mail', '=',
$email)->first();
return view('payment', compact('bookingEvent'));
}

public function init_payment(BookingEvent $bookingEvent){
    $data=array('status'=>'Оплачено');
    $bookingEvent->update($data);
    session(['bookingEvent' => $bookingEvent->id]);

    return redirect()->route('send');
}
}
```