

Детектор Плагіату v. 2961 - Звіт оригінальності: 29.05.2026 9:42:17

Перевірений документ: Ліцкевич М.В text.docx Ліцензія: ВОЛОДИМИР МАТІЄВСЬКИЙ_License2

- Тип пошуку: Пошук Перепису
- Знайдено мову: Uk
- Тип перевірки: Інтернет-перевірка
- ТЕЕ і кодування: DocX n/a

Детальний аналіз документа документа:

Співвідношення:

Плагіат 1.64% Оригінал 78.88%
Цитати 6.66% ШІ 12.82%



Графік розподілу:



Джерела плагіату: 4

0.8%	771	1. https://metod.suitt.edu.ua/download/992
0.7%	678	2. https://ela.kpi.ua/bitstream/123456789/29259/1/Oseredchuk_bakalavr.pdf
0.3%	288	3. https://ela.kpi.ua/bitstreams/40465e77-6d7f-4286-9f84-6f95c73e3d12/download

Дані оброблених ресурсів: 234 - ОК / 26 - Помилка

Важливі примітки:

Wikipedia: [не знайдено]	Google Books: [не знайдено]	Сервіси платних робіт: [не знайдено]	Античіт: Знайдена протидія!
---------------------------------	------------------------------------	---	------------------------------------

Звіт проти обману UACE:

1. Статус: Аналізатор Увімкнений Нормалізатор Увімкнений схожість символів встановлено 100%
2. Виявлений відсоток забруднення UniCode: 24,6% з обмеженням: 4%
3. Відсоток невизнаних символів після нормалізації: 8,5%
4. Усі підозрілі символи будуть позначені фіолетовим кольором: Abcd...
5. Знайдено невидимі символи: 0
Рекомендація з оцінки: Аналізу цього звіту слід приділити особливу увагу! Підозрюється, що цей документ містить значну кількість символів, чужих мові документа. Це є прямим вказівкою на те, що автор документа використав спеціальне програмне забезпечення(онлайн -веб -сервіс, щоб ефективно заплутати текст, намагаючись уникнути потенційного виявлення плагіату. Наполегливо рекомендується ескалювати цю справу! У разі сумнівів зверніться до служби підтримки детектора плагіату!
Статистика алфавіту та аналіз символів:

Активні посилання (URL-адреси, витягнуті з документа):

URL не знайдені

Виключено:

URL не знайдені

Включено:

URL не знайдені

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ ЗАКЛАД	
Знайдено посилань: 0,01% id: 1	
„ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МЕНТРИСЬ ШЕВЧЕНКО”	
Знайдено плагиату: 0,01% https://metod.suitt.edu.ua/download/992 id: 2	
Навчально-науковий інститут математики та інформаційних технологій Кафедра математики та інформатики Ліцкевич Микита Володимирович	
Розробка системи бронювання квитків в кваліфікаційна робота здобувача вищої освіти першого (бакалаврського) рівня освітньої програми	
Знайдено посилань: 0,01% id: 3	
«Комп'ютерні науки та інформаційні технології»	
за спеціальністю 122 Комп'ютерні науки Особистий підпис _____ Микита Ліцкевич Науковий керівник _____ (м'я) ПРЗВИЩЕ, науковий ступінь, наукове звання кафедри Мі Завдувач кафедри _____ (м'я) ПРЗВИЩЕ, науковий ступінь, наукове звання кафедри МІ Полтава – 2026 Міністерство освіти і науки України Державний заклад	
Знайдено посилань: 0,01% id: 4	
„Луганський національний університет ментрис Шевченко”	
Інститут математики та інформаційних технологій Кафедра Математики та інформатики Освітній рівень бакалавр Напрямок підготовки (спеціальність) 122	
Знайдено посилань: 0,02% id: 5	
“Комп'ютерні науки” (к.д.н.зв.) Глузень 12 “інформаційні технології” (к.д.н.зв.) ЗТВЕРДЖУЮ Завдувач кафедри МТ (м'я) ПРЗВИЩЕ (підпис) (м'я, пр.зв.) “ ”	
2	
Знайдено плагиату: 0,03% id: 6	
026 року. 2. Строк підання до виконання завдання. 3. Визначення теми роботи. 4. Вибір теми роботи (проект) (визначаються кількісні (та) як кількісні, яким повинен бути в дати об'єкт розробки) 4. Змст розробки - пояснювальні записки (перелік питань, які потрібно розробити) (визначаються назви розділів (та) перелік питань, які повинні увійти до тексту ПЗ) Консультанти розділів проекту (роботи) Розділ Прізвище, ініціали та посада консультанта Підпис, дата закінчення видачі завдання прийняв Дата видачі завдання”” 2026 року. КАЛЕНДАРНИЙ ПЛАН № з/п Назва етапів дипломного проекту (роботи) Строк виконання етапів проекту (роботи) Примітка 1. Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника. до 1 лютого 2. Аналіз літературних джерел за темою роботи. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи та плану експериментальних досліджень. до 20 березня 3. Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання науковим керівником. до 1 квітня 4. Усунення зауважень, урахування рекомендацій наукового керівника. Подання теоретичної частини роботи на друге читання. до 15 квітня 5. Проведення експериментальної роботи. Поступовий аналіз та обговорення її результатів. Перевірка стану виконання роботи. до 30 квітня 6. Урахування рекомендацій наукового керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Розробка презентації. до 15 травня 7. Попередній захист роботи на кафедрі до 30 травня 8. Доспрацювання роботи з урахуванням рекомендацій після передзахисту. Подання роботи науковому керівникові та рецензентів на підготовку вдругу та рецензії за 10 днів до державної атестації 9. Подання на кафедру остаточного варіанта роботи, переплетеного та підписаного автором, науковим керівником і рецензентом. за 5 днів до державної атестації Здобувач вищої освіти підпис Микита Ліцкевич (м'я, прізвище) Керівник проекту (роботи) підпис (м'я) ПРЗВИЩЕ (м'я, прізвище) НОТАЦІЯ Ліцкевич М.В. Тема: Розробка системи бронювання квитків в Спеціальність: 122	
Знайдено посилань: 0% id: 7	
“ ”	
2	
Знайдено плагиату: 0,05% id: 8	
https://ela.kpi.ua/bitstream/123456789/29259/1/Oseredchuk_2026.pdf 026 року. 2. Строк підання до виконання завдання. 3. Визначення теми роботи. 4. Вибір теми роботи (проект) (визначаються кількісні (та) як кількісні, яким повинен бути в дати об'єкт розробки) 4. Змст розробки - пояснювальні записки (перелік питань, які потрібно розробити) (визначаються назви розділів (та) перелік питань, які повинні увійти до тексту ПЗ) Консультанти розділів проекту (роботи) Розділ Прізвище, ініціали та посада консультанта Підпис, дата закінчення видачі завдання прийняв Дата видачі завдання”” 2026 року. КАЛЕНДАРНИЙ ПЛАН № з/п Назва етапів дипломного проекту (роботи) Строк виконання етапів проекту (роботи) Примітка 1. Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника. до 1 лютого 2. Аналіз літературних джерел за темою роботи. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи та плану експериментальних досліджень. до 20 березня 3. Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання науковим керівником. до 1 квітня 4. Усунення зауважень, урахування рекомендацій наукового керівника. Подання теоретичної частини роботи на друге читання. до 15 квітня 5. Проведення експериментальної роботи. Поступовий аналіз та обговорення її результатів. Перевірка стану виконання роботи. до 30 квітня 6. Урахування рекомендацій наукового керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Розробка презентації. до 15 травня 7. Попередній захист роботи на кафедрі до 30 травня 8. Доспрацювання роботи з урахуванням рекомендацій після передзахисту. Подання роботи науковому керівникові та рецензентів на підготовку вдругу та рецензії за 10 днів до державної атестації 9. Подання на кафедру остаточного варіанта роботи, переплетеного та підписаного автором, науковим керівником і рецензентом. за 5 днів до державної атестації Здобувач вищої освіти підпис Микита Ліцкевич (м'я, прізвище) Керівник проекту (роботи) підпис (м'я) ПРЗВИЩЕ (м'я, прізвище) НОТАЦІЯ Ліцкевич М.В. Тема: Розробка системи бронювання квитків в Спеціальність: 122	
Знайдено посилань: 0% id: 9	
„Комп'ютерні науки”	
Установка: ДЗ ЛНУ імені Т.Шевченка, 2026 р. Кваліфікаційна робота містить: 53 с., 16 рис., 5 т. бл., 10 додат., 16 джерел. Об'єкт дослідження процес онлайн-	
Знайдено плагиату: 0,04% id: 10	
https://ela.kpi.ua/bitstream/10465/677-6d7f-4286-9f84-6405 бронювання квитків в на культурно-розважальні об'єкти. Предмет дослідження методи, засоби та технології веб-розробки бронювання квитків для культурно-розважальної системи. Мета роботи розробка сайту для бронювання квитків в на екскурсії музеями, зоопарками та закладами міста Познань в Польщі.	
. У результаті виконання кваліфікаційної роботи було створено вебсистему бронювання квитків, яка забезпечує перегляд інформації про культурно-розважальні заклади, вибір дати та часу відвідування, оформлення бронювання та генерацію електронного квитка у форматі PDF.	
AI generated text detected: ChatGPT 5.3, : 57,2% id: 11	
Впроваджена система дозволяє осучаснити процес резервування квитків, спростити взаємодію споживача із закладами та забезпечити зручний доступ до нагальної інформації через веб-інтерфейс. Під час виготовлення було вжито технологій HTML, CSS, JavaScript, PHP та MySQL. Створено структуру бази даних для збереження інформації про заклади, типи квитків, екскурсійні групи та бронювання користувачів. Також реалізовано адаптивний інтерфейс, який забезпечує правильну роботу системи на персональних комп'ютерах і мобільних девайсах.	
Отримані результати є підтвердженням ефективності використання сучасних вебтехнологій для створення систем онлайн-бронювання. Розроблена система може використовуватися як основа для подальшого розвитку повноцінного сервісу резервування квитків із підключенням платіжних систем, адмініструванням закладів та	

автоматизованою обробкою замовлень. Ключові слова: ВЕБСИСТЕМА, БРОНЮВАННЯ КВИТКІВ, HTML, CSS, JAVASCRIPT, PHP, MYSQL, БАЗА ДАНИХ, ЕЛЕКТРОННИЙ КВИТОК, ВЕБРОЗРОБКА, АВТОМАТИЗАЦІЯ. ABSTRACT Litskevych M. Theme: Development of a Ticket Booking System Specialty: 122	
 Знайдено посилань: 0%	id: 12
"Computer Science"	
Object of study: the process of online ticket booking for cultural and entertainment venues. Subject of study: methods, tools, and web development technologies for ticket booking systems in the cultural and entertainment sector. Purpose of the work: the development of a website for booking tickets to tours of museums, zoos, and castles in the city of Poznań, Poland. As a result of the qualification work, a web-based ticket booking system was developed, providing access to information about cultural and entertainment venues, selection of the date and time of visits, booking management, and generation of electronic tickets in PDF format. The implemented system modernizes the ticket reservation process, simplifies user interaction with venues, and ensures convenient access to relevant information through a web interface. During the development process, HTML, CSS, JavaScript, PHP, and MySQL technologies were used. A database structure was created to store information about venues, ticket types, tour groups, and user bookings. In addition, a responsive interface was implemented to ensure the correct operation of the system on both personal computers and mobile devices. The obtained results confirm the effectiveness of using modern web technologies for the development of online booking systems. The developed system can serve as a foundation for the further expansion of a full-featured ticket reservation service with payment system integration, venue administration, and automated order processing. Keywords: WEB SYSTEM, TICKET BOOKING, HTML, CSS, JAVASCRIPT, PHP, MYSQL, DATABASE, ELECTRONIC TICKET, WEB DEVELOPMENT, AUTOMATION.	
ЗМІСТ ВСТУП7 РОЗДІЛ 1. АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ8 1.1.Еволюція наукової думки та аналіз підходів до автоматизації систем бронювання8 1.2.Аналіз вимог технічних рішень та технологічних процесів10 1.3.Аналіз існуючих програмних продуктів та аналогів в системі13 РОЗДІЛ 2. АНАЛІЗ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ16 2.1.Специфікація вимог до програмного забезпечення та визначення пріоритетів розробки16 2.2.Вибір та аналіз програмного забезпечення для розробки вебсайту18 2.3.Проектування користувацького інтерфейсу вебсайту22 РОЗДІЛ 3. СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ39 3.1.Розробка бази даних39 3.2.Аналіз зв'язків між таблицями45 3.3.Розробка сайту46 ВИСНОВОК51 СПИСОК ДЖЕРЕЛ53 ВСТУП	
Останнім часом інформаційні технології роблять великі стрибки. Для виконання простих життєвих завдань як люди самі собі ставлять. Вже як буденність використовується Інтернет: купівля продуктів, побутової хімії, канцелярії для школярів, одягу та взуття, бронювання або купівля квитків на поїзд, літак, кіно, автобусний рейс. Достатньо для цього мати телефон та доступ в глобальну мережу або додаток того магазину чи закладу який тобі потрібен за ситуацією. Більшість послуг переходить у цифровий вигляд, що дає змогу користувачам швидко та зрозуміло отримувати потрібну інформацію, робити придбання та резервування через мережу Інтернет. Особливо важливим є впровадження вебсистем для бронювання квитків на культурно-розважальні події та туристичні локації. Музеї, замки та зоопарки міста Познань щодня відвідує значна кількість мандрівників та місцевих жителів. Звичні методи купівлі квитків нерідко спричиняють клопоти для українців за кордоном, які пов'язані з володінням мовою, чергами, обмеженим режимом праці кас та труднощами здобуття свіжої довідки про ціну й наявність квитків. Саме тому розробка сучасної системи Інтернет-резервування є нагальним завданням. Напрямом кваліфікаційної роботи є створення вебсистеми резервування перепусток для музеїв, замків та зоопарків міста Познань, яка гарантує користувачам спроможність перегляду відомостей про доступні заклади, види квитків, здійснення резервування та одержання їх електронного варіанту. Практична цінність кваліфікаційної роботи полягає у створенні зручної та сучасної вебсистеми, яку можна використати для спрощення процесу продажу та резервування квитків для туристично-культурних установ. АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ Еволюція наукової думки та аналіз підходів до автоматизації систем бронювання Розвиток наукової думки у сфері автоматизації процесів бронювання пройшов тривалий шлях від найпростіших локальних баз даних до складних розподілених інтелектуальних систем. Одна з перших в Україні систем онлайн бронювання була в	
 Знайдено посилань: 0%	id: 13
«Укрзал зниц».	
У 2008 році вона ввела систему електронного бронювання за 45 днів до потягу, з можливістю викупу квитка у будь-якій касі	
 Знайдено посилань: 0%	id: 14
«Укрзал зниц»	
протягом двох днів від дня бронювання.[1] Натомість в Польщі перша згадка про бронювання електронних квитків була у 2001 році, і як це не дивно також пов'язано з залізницею, а саме	
 Знайдено посилань: 0%	id: 15
«PKP Intercity».	
За інформацію, що розміщена на офіційному сайті, першого вересня 2001 року ця компанія відділилася від групи компаній	
 Знайдено посилань: 0%	id: 16
«PKP»,	
що дало їй можливість розвитку самостійно, хоча вона все ж таки залишилася дочірньою компанією, і саме в 2001 році вони офіційно відкрили можливість лише онлайн бронювання, в випадку як і з	
 Знайдено посилань: 0%	id: 17
«УЗ»	
псля бронювання сам квиток потрібно йти та викуповувати на будь-якій касі	
 Знайдено посилань: 0%	id: 18
«PKP Intercity».	
У 2002 році вони відкрили кол-центр, що дало змогу дзвонити та бронювати квиток не лише на сайті, що на той момент дало можливість трохи зменшити навантаженість на так не ідеальну систему. [2] На початкових етапах розробники були зосереджені на самому факті переходу від паперового обліку до цифрового збереження інформації, де основна увага приділялася надійному зберіганню даних на централізованих серверах. 2003 рік, в	
 Знайдено посилань: 0%	id: 19
«PKP Intercity»	
з'являється сайт з інформацією про тарифи, сполучення та акції, а вже в 2005 році запускається онлайн продаж квитків. [2] Проте перші програмні рішення були занадто	

громадзким та не дозволяли забезпечувати операційну гнучкість у реальному часі. Роботи попередників у цей період переважно базувалися на архітектурі

Знайдено посилань: 0%

id: 20

«клєнт-сервер», яка хоч і вирішувала питання централізації, проте створювалася значна затримка при одночасному зверненні великої кількості користувачів, що призводило до критичних помилок синхронізації. Критичний аналіз існуючих напрацювань свідчить про те, що значний час спільнота ігнорувала питання адаптивності та кросплатформеності, зосереджуючись виключно на функціональній повноті. Більшість класичних систем, створених на початку 2010-х років, страждають від надмірної складності інтерфейсів та неоптимального використання обчислювальних ресурсів, що робить їх важкими для підтримки та масштабування.

AI generated text detected: ChatGPT 5.3, : 60,29%

id: 21

Попри суттєві успіхи у впровадженні хмарних технологій, багато теоретичних моделей все ще залишаються занадто абстрактними та не враховують специфіку вузькопрофільних ринків. На сьогодні залишаються невирішеними питання: повної автоматизації конфліктних транзакцій у децентралізованих середовищах інтеграції систем бронювання з ризикованими зовнішніми сервісами без втрати продуктивності. Також помітна відсутність обґрунтованих підходів до створення легких архітектур. Такі архітектури повинні були б доступні для малого та середнього бізнесу. Тому важливо звернути увагу на впровадження механізмів динамічного керування потоками запитів та удосконалення алгоритмів перевірки доступності ресурсів у реальному часі. Розробка системи, яка поєднуватиме високу продуктивність із простим та зрозумілим інтерфейсом, є законним етапом розвитку сучасної програмної інженерії. Забезпечення конфіденційності, цілісності та доступності персональних і транзакційних даних користувачів детермінується як один із домінантних векторів розвитку сучасної індустрії інформаційних технологій.

Ретроспективний аналіз еволюції програмних систем свідчить, що на ранніх етапах проектування зазначені масиви інформації акумулювалися без застосування релевантних криптографічних та архітектурних механізмів безпеки, що зумовлювало високу уразливість до несанкціонованого доступу та витоку даних. Крім того, дисипація (розпорошення) уваги розробників щодо питань сумісності, кросплатформової інтеграції із сучасними платіжними шлюзами та державними цифровими екосистемами призвела до виникнення системних колізій під час масштабування, модернізації та комплаєнсу (відповідності) архітектури ПЗ актуальним нормативним стандартам безпеки. На сьогодні актуальною залишається проблема впровадження легких моделей прогнозу аналітики, здатних завчасно визначати можливі пікові навантаження на систему бронювання. Це дозволило б автоматизовано перерозподіляти ресурси ще до виникнення критичних ситуацій. Також важливим питанням є оптимізація кросплатформенної взаємодії, оскільки підтримка однакового функціоналу на мобільних і настільних пристроях часто призводить до значного ускладнення структури програмного коду. У науково-практичному дискусії проектування сучасних систем бронювання особливої актуальності набуває інтеграція механізмів багатфакторної автентифікації, зорієнтованих на мінімізацію когнітивного й ергономічного навантаження на користувача та збереження високого рівня юзабіліті інтерфейсу. Паралельно критичним аспектом виступає впровадження інструментарію автоматизованого стрес-тестування для верифікації відмовостійкості та стабільності функціонування архітектури в умовах пікових навантажень. Як домінантний вектор розвитку визначено застосування модульних архітектурних рішень (зокрема мікросервісної організації), що забезпечують безперервну інтеграцію (CI/CD) нових платіжних шлюзів та протоколів верифікації без деградації або зупинки працездатності магістральної системи. Такий підхід забезпечує більшу гнучкість програмного забезпечення та підвищує його стійкість до змін технологічного середовища і сучасних вимог безпеки. Аналіз введених технічних рішень та технологічних процесів в еволюції систем бронювання відбувався в простих монолітних програм до сучасних розподілених архітектур, побудованих із використанням хмарних технологій. На перших етапах основою таких рішень були реляційні бази даних із жорстко визначеною структурою, де запити оброблялися послідовно. Через це під час пікових навантажень виникали значні затримки в роботі системи. Основна увага під час розробки приділялася підтриманню цілісності даних, тоді як питання масштабованості залишалися другорядними. Більшість ранніх систем використовували синхронну модель обробки запитів, через що користувач був змушений очікувати завершення всієї операції перед отриманням відповіді. В умовах сучасності та високої динаміки онлайн-сервісів такий підхід виявився недостатньо ефективним. Експоненційне зростання аудиторії та інтенсифікація використання сучасних вебплатформ зумовили необхідність ревізії традиційних парадигм проектування систем резервування. Особливої науково-практичної гостроти набула проблема забезпечення детермінованої відмовостійкості сервісів у періоди пікового (масового) надходження транзакційних запитів. Зазначений чинник виступив каталізатором переходу до динамічних архітектурних концепцій, базовими векторами яких є горизонтальна масштабованість, висока толерантність до відмов, а також оптимізація латентності при обробці великих масивів даних (Big Data) та високої щільності вхідного трафіку. Ретроспективний аналіз технологічних рішень попереднього десятиліття свідчить про домінування монологічних і закритих систем, модернізація та супровід яких супроводжувалися високою ресурсомісткістю обчислювальних процесів. Дослідження архітектурних підходів вказує на те, що застосування реляційних баз даних із песимістичними механізмами блокування табличного рівня детермінувало регулярну генерацію взаємоблокувань (колізій типу

Знайдено посилань: 0%

id: 22

«deadlock»

). Зазначена деструкція виникала за умов високої конкурентності запитів, коли множинні потоки користувачів синхронно зверталися до унітарного обчислювального ресурсу. При цьому імплементація методів серверного кешування виявлялася інваріантною і не забезпечувала повного нівелювання проблеми нерівномірної дисипації навантаження. Питання можливості у бік навантаження та автоматичного масштабування ресурсів у режимі: нагального часу залишаються недостатньо втрішеними, особливо у рамках бюджетних програмних розробок. Традиційні алгоритми диспетчеризації черг вхідних запитів характеризуються дефіцитом урахування просторово-географічної диспозиції клієнтських терміналів та динамічних параметрів мережевої латентності. Таке ігнорування топологічних характеристик мережі призводить до нерівномірного розподілу часу затримки пакетів та деградації обчислювальної ефективності вебресурсу. У підсумку це зумовлює зниження нормативних показників якості обслуговування та суб'єктивної оцінки якості сприйняття системи користувачем. Першочергової імплементації потребують обчислювальні процеси, що базуються на патернах асинхронної обробки повідомлень та подійно-орієнтованому неблокуючому введенні-виведенні. Задля підтримання константної

релевантності та консистентності даних доцільно залучати протоколи повнодуплексного зв'язку в режимі реального часу. Це забезпечує реактивну синхронізацію станів об'єктів резервування та динамічне оновлення клієнтського інтерфейсу без ініціалізації повторних HTTP-запитів користувачем. Фундаментальним вектором розвитку визначено інтеграцію ізольованих мікросервісних рішень із вбудованими механізмами патернів відмовоустійчивості, що гарантує збереження загальної працездатності системи за умов каскадної деградації її окремих компонентів. Розробка гнучких API-інтерфейсів для безпечної взаємодії з платіжними системами та сервісами ідентифікації є ключовим етапом у створенні конкурентоспроможного програмного забезпечення для бронювання квитків. Сучасний процес розробки систем бронювання все більше зміщується у бік використання архітектури [Single Page Application \(SPA\)](#) у поєднанні з потужними інструментами керування станом на сторінці клієнта. Це дозволяє перенести значну частину обчислювального навантаження з сервера на пристрій користувача, що забезпечує плавність інтерфейсу під час інтерактивного вибору місць на схемах залів або під час бронювання білетів на екскурсії до музеїв. Однак критичний огляд робіт показує, що такий підхід часто супроводжується проблемами з безпекою бізнес-логіки, яка може бути вразливою до маніпуляцій на рівні клієнтського коду. Велика кількість існуючих технічних рішень не враховувала необхідність впровадження наскрізного шифрування на рівні прикладних протоколів та використання токенізованої авторизації (наприклад, [JSON Web Token \(JWT\)](#)), що створювало ризики перехоплення сесій під час процесу оформлення квитка. У системі бронювання необхідно не забувати про безпеку клієнта, тому важливо забезпечити перевірку дій користувача зі сторони сервера, за причини того, що частина бізнес-логіки може бути змінена, або піддроблена у його код. Якщо про це забути, то виникає ризик несанкціонованого бронювання, змни вартості квитка або створення тимчасового дубляту профілю для повторного використання в закритій сесії. На даний момент для зменшення таких загроз сучасні вебсистеми використовують захищену передачу даних [HTTPS-протокол](#), токенізовану авторизацію [JWT](#), а також механізми перевірки автентичності запитів на сервері. [HTTPS](#) це протокол який є розширеною версією протокола [HTTP](#). У протоколі передача даних здійснюється за допомогою використання криптографічних протоколів [SSL](#) або [TLS](#). Завдяки їх використанню більша частина інформації, яка передається користувачем серверу, шифрується, зокрема [URL](#)-запити, параметри даних та [cookie](#)-файли, які можуть містити інформацію про сесію користувача. Назва хоста та номер порту не підлягають шифруванню через те, що необхідно для встановлення коректного мережевого з'єднання протоколами [TCP/IP](#). Незважаючи на активне впровадження хмарних сервісів, залишається невирішеним питання ефективної контейнеризації компонентів системи для забезпечення їх миттєвого розгортання в зольованих середовищах. Проблема

Знайдено посилань: 0%

id: 23

«х л д н г о старту»

окремих модулів системи під час різкого зростання трафіку довіди залишається актуальною для багатьох комерційних аналогів, що призводить до тимчасової недоступності сервісу в моменти відкриття продажів на популярні події. Також залишається недостатньо дослідженим питання автоматизованого забезпечення узгодженості даних у гібридних системах зберігання даних, де реляційні бази використовуються для обробки фінансових операцій, а [NoSQL](#)-рішення для швидкого кешування інформації про доступність квитків. У зв'язку з цим, першочерговим впровадження потребують технологічні процеси безперервної інтеграції та доставки (CI/CD), які дозволяють проводити оновлення системи без зупинки обслуговування користувачів. Пріоритетним технічним рішенням має стати впровадження механізмів розподіленого кешування на рівні окремих вузлів системи, що дозволить мінімизувати кількість прямих звернень до основної бази даних. Окрім цього, важливим аспектом є використання сучасних протоколів захисту під час взаємодії з банківськими еквайринговими сервісами, що дозволить забезпечити безпечне проведення фінансових операцій. Зосередження уваги на зазначених інженерних рішеннях дасть можливість наблизити розроблювану систему до сучасних галузевих стандартів, забезпечуючи її надійність, стабільність роботи та захищеність у реальних умовах використання. Аналіз існуючих програмних продуктів та аналогів в системі Ринок програмного забезпечення для бронювання квитків прийшов значний шлях розвитку — від закритих корпоративних мереж авіакомпаній до сучасних глобальних систем дистрибуції. Початкові етапи розвитку даного напрямку були пов'язані зі створенням централізованих систем обробки інформації, серед яких одними з найвідоміших стали системи [Sabre](#) та [Amadeus](#). Зазначені технологічні розробки сформували канонічний базис цифрового обліку ресурсів та автоматизації процесів резервування. Платформа [Sabre](#) виступила однією з перших глобальних комп'ютерних систем дистрибуції, що забезпечила автоматизацію транзакцій авіаперевезень і детермінувала вектор розвитку електронного сегмента індустрії. У свою чергу, система [Amadeus](#) була спроектована консорціумом європейських авіаперевізників як централізований комплекс для консолідованого бронювання пасажирських місць, готельного фонду та об'єктів культурно-розважальної інфраструктури. Системний аналіз сучасних рішень свідчить, що попри фундаментальний внесок у генезис автоматизованих систем, класичні платформи характеризуються високою інтеграційною складністю та вимагають значних фінансових операційних витрат, що обмежує їх імплементацію в секторі малого та середнього бізнесу. Новітні комерційні сервіси (зокрема [Booking.com](#) або [Tickets.ua](#)) володіють диверсифікованим функціоналом, проте переважно функціонують у межах посередницької парадигми. Це створює архітектурні бар'єри щодо латентності обміну даними та нівелює можливість гнучкої кастомізації системи під специфічні вимоги локальних провайдерів послуг. Попри проліферацію (розмноження) готового програмного забезпечення, низка науково-технічних задач залишається невирішеною. Серед них надмірна функціональна надмірність систем, яка зумовлює деградацію продуктивності у мобільних браузерах та на терміналах з обмеженими апаратними ресурсами. Більшість існуючих платформ базується на ригідних (жорстких) архітектурних паттернах, що ускладнює адаптацію логіки під нестандартні типи квитків або специфічні топології просторового розміщення (схеми розсадки) без масштабного рефакторингу вихідного коду ПЗ. Крім того, спостерігається дефіцит рішень, які когнітивно поєднують високі критерії криптографічного захисту фінансових транзакцій із прозорими, інтуїтивно зрозумілими інтерфейсними афродансами. У межах проектування запропонованої системи пріоритетну увагу доцільно приділити математичним механізмам динамічної візуалізації доступних ресурсів та оптимізованим алгоритмам диспетчеризації черг конкурентних запитів. Такий підхід дозволить запобігти виникненню транзакційних колізій під час масового синхронного звернення користувачів. Одним із ключових завдань дослідження є синтез легковагової архітектури, здатної мінімізувати час відгуку сервера та забезпечити оперативну реактивну синхронізацію статусів об'єктів між реляційною/нереляційною базою даних та клієнтською частиною. Реалізація таких рішень дасть можливість усунути ключові недоліки більшості сучасних аналогів, забезпечивши високу швидкість системи, її масштабованість та зручність використання. Під час виконання кваліфікаційної роботи було проаналізовано вебсайти музеїв, зоопарків і замків міста Познань. У більшості випадків на цих ресурсах відсутня функція онлайн-бронювання

квитків, а користувачам доступний лише актуальний прайс-лист, який змінюється залежно від сезону, дня або місяця. Також на сайтах переважно розміщуються фотографії закладів, проведених заходів та культурних подій.

AI generated text detected: ChatGPT 5.3, : 55,65%

id: 24

АНАЛІЗ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ Специфікація вимог до програмного забезпечення та визначення пріоритетів у розробці ПДХ є важливими етапами формування вимог для систем бронювання, часом суттєво змінилися — від опису базових операцій з збереження даних до створення комплексних специфікацій, що враховують доступність, безпеку та адаптивність системи. Розвиток методологій у цій сфері відбувався від жорстких моделей планування, де вимоги визначалися лише на початковому етапі розробки, до сучасних гнучких підходів, які дозволяють змінювати функціональність впродовж дозволу користувачів та ринкових умов.

Аналіз попередніх підходів до документування вимог показує, що значна увага приділялася кількісним характеристикам функціоналу, тоді як зручність взаємодії користувача з інтерфейсом часто залишалася другорядною. Як наслідок, у процесі розробки синтезувалися програмні комплекси, що формально відповідали вихідним технічним вимогам, проте під час експериментальної експлуатації демонстрували високу ергономічну складність та втрату стабільності в умовах пікових навантажень. Сучасний науково-методичний інструментарій недостатньо враховує специфіку конвергенції (об'єднання) вебплатформ із гетерогенними платіжними шлюзами та сервісами автоматизованого сповіщення, що суттєво обмежує потенціал наскрізної автоматизації бізнес-процесів. Актуальними невирішеними задачами наразі залишаються гарантування відмовостійкості систем за умов нестабільного мережевого з'єднання, а також підтримка транзакційної складності та втрату стабільності при паралельному доступі. У чинних технічних специфікаціях спостерігається дефіцит деталізованих сценаріїв обробки виняткових ситуацій при ініціалізації фінансових транзакцій, що виступає тригером колізій на рівні СУБД. Крім того, фіксується істотна дивергенція між теоретичними моделями безпеки персональних даних та їх практичною імплементацією у вебдодатках, що функціонують у незахищених розподілених мережах. Відсутність уніфікованих стандартів синхронізації станів об'єктів у багатокористувацькому середовищі призводить до латентних архітектурних дефектів, які верифікуються виключно під час навантажувального стрес-тестування. У цьому контексті першочергової детермінації потребують вимоги щодо реактивного (миттєвого) оновлення статусів квитків та забезпечення високого рівня живучості серверної інфраструктури. Пріоритетним є створення специфікації, яка визначатиме параметри швидкодії, вимоги до безпеки автентифікації користувачів та логіку взаємодії з базою даних через асинхронні блокуючі запити. Також важливо сформулювати вимоги до функцій, що забезпечуватимуть повний цикл бронювання — від вибору місця до автоматичної генерації підтверджувальних документів. Реалізація зазначених пріоритетів дозволить створити надійну основу для подальшої розробки програмного забезпечення впродовж до сучасних вимог якості та надійності.

AI generated text detected: ChatGPT 5.3, : 79,31%

id: 25

У межах кваліфікаційної роботи планується розробка вебсайту, який мститиме три основні напрямки бронювання квитків: Музеї Зоопарки Замки Для кожного напрямку буде створено перелік закладів міста Познань та його околиць. Назви закладів вображатимуться польською мовою, тоді як опис буде представлений українською, оскільки сайт орієнтований переважно на українських користувачів, які планують відвідати культурно-розважальні заклади Познань та прилеглих територій. Для кожного закладу передбачається окрема сторінка з коротким описом, інформацією про кількість вільних місць в екскурсійній групі, можливістю вибрати дату відвідування та актуальними цінами, що були чинними на момент створення кваліфікаційної роботи. У межах роботи буде реалізовано демонстраційний модуль оплати, оскільки створення повноцінної платіжної системи потребує впродовж дозволу та офіційної реєстрації підприємницької діяльності. Система бронювання мститиме обов'язкові поля для введення імені та прізвища користувача, а також можливість вибрати дату, тип та кількість квитків. Після вибору квитків система автоматично вображатиме загальну вартість замовлення. Також буде реалізована функція завантаження електронного квитка, у якому зазначатимуться: Назва закладу Ім'я Прізвище Дата Номер екскурсійної групи Тип та кількість квитків Ціна Підтвердження про зачасне прибуття до місця події Оскільки заклади можуть відвідувати діти, у системі передбачатиметься інформаційне попередження про те, що діти без супроводу дорослих не можуть знаходитися на екскурсії самостійно. Також підтвердження дозволяє користувачам уникнути помилки під час оформлення квитків для дітей та забезпечить можливість перебування дітей в екскурсійній групі разом із батьками або супроводжувачами. Вибір та аналіз програмного забезпечення для розробки вебсайту Основа кваліфікаційної роботи є створення програмного продукту, орієнтованого на спрощення процесу бронювання квитків до музеїв, зоопарку та замку. Вебсайт, який планується розробити, буде спрямований на реалізацію зручного та доступного механізму бронювання для користувачів.

Туристичні вебсайти на яких в даний момент можна забронювати квитки до музеїв, зоопарків та замків, пропонують бронювати лише комплексні види екскурсій. Це не завжди є доцільним, особливо коли людина не бачила лише один музей з переліку в інтернеті. Через це програмний продукт буде розроблений на можливість вибору бронювання екскурсії для одного конкретного музею, зоопарку, замку. Майже всі заклади мають дні в які є знижки на квитки або взагалі безкоштовні дні відвідування, і про це туристичні сайти навіть не повідомляють. Для створення вебсайту буде використовуватися низка програм таких як: Sublime Text 3 Рисунок 2.1 Емблема програми Sublime Text 3 XAMPP Рисунок 2.2 Емблема програми XAMPP Apache Рисунок 2.3 Емблема вебсервера Apache MySQL Рисунок 2.4 Емблема програми MySQL Google Chrome Рисунок 2.5 Емблема браузера Google Chrome Sublime Text 3 – це швидкий кросплатформний текстовий редактор. Підтримує плагіни, розроблені за допомогою мови програмування Python.[5] Його інтерфейс простий та зручний, він підтримує написання html, css, php, javascript, що є необхідним для комфортної розробки вебсайту. Додатковою перевагою редактора є висока швидкість роботи навіть при відкритті великої кількості файлів, а також підтримка підсвічування синтаксису та автоматичного форматування коду. Це значно спрощує процес написання та редагування програмного коду під час створення вебпроекту. HTML – це мова гіпертекстової розмітки для створення вебсайту, основа та фундамент сайту. Саме HTML відповідає за структуру вебсторінки, розміщення тексту, кнопок, таблиць, зображень та інших елементів інтерфейсу. За допомогою HTML формується логічна структура сторінок вебсайту, що забезпечує правильне відображення контенту у браузері користувача. CSS – відповідає за візуальну складову та зовнішній вигляд вебсторінок, шрифти, колір, розмір тексту, блоків, малюнків, таблиць, та інших елементів. CSS дозволяє створити єдиний стиль оформлення всіх сторінок вебсайту. Також використовується для створення анімацій, ефектів наведення та покращення загального дизайну

користувачького інтерфейсу. PHP – це популярна мова програмування, особливо серед веб-розробників. Автором початкової версії є Рasmus Лердорф, ідея якого полягала у розробці набору інструментів спрощення процесу створення динамічних веб-сторінок. Незважаючи на те, що сучасний PHP є мовою загального призначення, найчастіше його використовують як серверний інструмент для генерації HTML-коду, який потім інтерпретується браузером. [6] а JavaScript (JS) – це високорівнева скриптова мова програмування, яка широко використовується для створення інтерактивних застосунків і сайтів. Вона може створювати анімацію, спливаючі повідомлення і змушує інтерфейс реагувати на наші дії.[7] У зв'язі з HTML і CSS JavaScript надає змогу розробити повнофункціональний вебсайт із динамічним оновленням інформації без перезавантаження сторінки. У межах проекту JavaScript буде використовуватись для перевірки правильності введення даних у формах бронювання, автоматичного обчислення вартості квитків, взаємодії з елементами інтерфейсу та покращення загального користувацького досвіду. XAMPP – це простий в налаштуванні та безкоштовний локальний вебсервер. Така назва не просто так, це є абревіатура яка розшифровується на компоненти. X – визначає її кросплатформенність, вона працює на Windows, Linux, MacOS. A – Apache, компонент програми, безкоштовний вебсервер із відкритим кодом, який є центральним пакетом програми. M – MySQL (MariaDB), компонент програми, це система керування реляційними базами даних. Завдяки XAMPP можна працювати з MySQL в зручному інтерфейсі phpMyAdmin. Рисунок 2.6 Емблема інтерфейсу phpMyAdmin P – PHP, компонент програми, який допомагає пов'язати базу даних з сайтом. P – Perl, компонент програми, мова програмування для автоматизації та розробки сайтів, однак цю мову витіснила мова PHP. Цей компонент програми не знадобиться для розробки сайту. Google Chrome – це безкоштовний веббраузер, розроблений компанією Google на основі браузера з відкритим кодом Chromium та іншого відкритого програмного забезпечення.[8] Він потрібен для перевірки коректності розміщення блоків сайту, та його вигляду. Проектування користувацького інтерфейсу вебсайту Проектування користувацького інтерфейсу вебсайту є важливою складовою розробки програмного забезпечення про яку неможна забувати. Від якості реалізації залежить швидкість пошуку необхідної інформації, зручність навігації та загальне враження користувача від роботи з вебсайтом. Інтерфейс повинен бути зрозумим навіть для користувачів з малим досвідом роботи в мережі Інтернет, оскільки складна структура сторінок або перезавантаження елементів можуть ускладнювати процес бронювання квитків.

AI generated text detected: ChatGPT 5.3, : 63,05%

id: 26

Під час проектування вебсайту потрібно враховувати такі основні аспекти: Простота сторінок Логічність та послідовність розміщення елементів Функціональність інтерфейсу Зручність навігації Адаптивність для різних пристроїв Основні функції вебсайту повинні бути доступними без виконання великої кількості дій. Особлива увага повинна бути направлена на навігацію сайтом.

Користувач повинен мати можливість швидко перейти до необхідного розділу, переглянути доступні закладки, ознайомитися з інформацією про події та здійснити бронювання квитка без складних переходів між сторінками. Для цього потрібно реалізувати зрозуміле меню навігації, логічну структуру сторінок та помітні кнопки переходу між основними розділами системи. Під час розробки інтерфейсу сайту також необхідно враховувати психологічні аспекти сприйняття інформації. Інформаційна надмірність, деструктивні колористичні схеми та перевантаженість композиційних шаблонів вебсторінок підвищують рівень когнітивного навантаження на користувача, що лімітує швидкість перцепції (сприйняття) та верифікації релевантних даних. З огляду на це, обґрунтованою є імплементація принципів мінімалістичного дизайну, що передбачає нормовану квантифікацію колірної палітри, однозначну семантику елементів керування та логічну кластеризацію контенту. Для фокусування уваги оператора на магістральних функціональних вузлах системи застосовано методи візуального акцентування цільових компонентів інтерфейсу — зокрема інтерактивних модулів ініціалізації резервування, селекторів часових інтервалів (дат) та фінального підтвердження транзакцій. Окремої детермінації потребує адаптивність клієнтської частини. Зважаючи на стійку тенденцію превалювання мобільного трафіку (смартфонів та планшетних EOM), архітектура фронтенду має гнучко трансформувати просторову топологію елементів відповідно до геометричних параметрів екрана (*responsive design*), гарантуючи стабільний рівень кросплатформного юзабіліті (*\$usability\$*). Ефективність взаємодії з платформою також корелює з часом інтерфейсного відгуку. Система повинна оперативно обробляти користувацькі івенти, забезпечуючи низьку латентність (*\$latency\$*) при динамічній реплікації даних про доступний квитковий фонд, що мінімізує операційні помилки та максимізує користувацький досвід (*\$UX\$*). Важливою підсистемою є модуль покрокового оформлення (*\$wizard-pattern\$*) та рендерингу електронних ваучерів (квитків). Процедурі лінійної дистрибуції оптимізовано шляхом декомпозиції на послідовні дискретні етапи без надмірних транзакційних кроків. Інтерфейс забезпечує перманентну валідацію введених параметрів, відображає квоти вільних місць у режимі реального часу, а після завершення сесії автоматично генерує електронний документ у детермінованому форматі (наприклад, PDF) із підтримкою механізмів візуальної ієрархії для швидкого зчитування атрибутів замовлення. Інтерфейсні рішення, що супроводжуються контекстними повідомленнями та валідаційними триггерами, спроектовані в уніфікованому стилі з дотриманням гайдлайнів доступності (*accessibility*, *W3C Web Accessibility Initiative*) для різних вікових груп. Прототипування та композиційне моделювання інтерфейсу виконано у середовищі векторного графічного редактора Figma. Створення макетів у середовищі Figma дозволяє попередньо оцінити зовнішній вигляд вебсайту, структуру сторінок та взаємодію між елементами інтерфейсу ще до початку програмної реалізації. Це значно спрощує процес проектування та дозволяє своєчасно виявити можливі недоліки структури або дизайну. Також використання макетів допомагає сформувати чітке уявлення про кінцевий вигляд програмного продукту та забезпечує більш ефективну організацію подальшої розробки вебсайту.

AI generated text detected: ChatGPT 5.3, : 55,32%

id: 27

На сайт планується розробити такі сторінки: Головна сторінка Сторінка музеїв Сторінка зоопарку Сторінка замку Загальна сторінка для кожного музею Загальна сторінка для кожного зоопарку Загальна сторінка для кожного замку Сторінка бронювання Сторінка перевірки правильності введеної інформації Сторінка успішності бронювання, з можливістю завантаження PDF варіанту квитка.

Макет головної сторінки виглядає ось так: Рисунок 2.7 Макет головної сторінки сайту Під час проектування особлива увага приділялася простоті сприйняття інформації, зручності навігації та логічному розміщенню основних елементів інтерфейсу. Усі основні блоки розташовані таким чином, щоб користувач міг швидко та легко знайти потрібну інформацію та перейти до потрібної категорії закладів без виконання великої кількості дій. Під номером 1 фон сторінки. Він виконує роль контейнера для всіх блоків сайту та забезпечує цілісність візуального оформлення вебсторінки. Також фон створює загальний стиль інтерфейсу та впливає на комфортне сприйняття інформації користувачем. У позиції

2 розташовано ідентифікатор вебресурсу (назву сайту), ключове призначення якого полягає в інформаційній підтримці та спрощенні візуальної навігації користувача. У подальших модифікаціях інтерфейсу допускається масштабування цієї області шляхом розміщення графічного логотипу чи допоміжних елементів керування. Під номером 3 буде знаходитись короткий опис сайту. Даний блок призначений для ознайомлення користувача з основними можливостями веб-ресурсу. Це дозволить новим користувачам швидко зрозуміти призначення системи. Під номером 4 знаходяться картки категорій. Вони відповідають за кожну категорію музеї, зоопарки, замки. В кожній картці буде знаходитись кнопка відповідної категорії, що дозволить користувачу швидко візуально відзняти розділи між собою. Використання карток робить інтерфейс більш сучасним та покращує зручність навігації по вебсайту. Під номером 5 – назва категорії та її короткий опис. Цей блок дозволяє користувачу швидко ознайомитися зі змістом відповідного розділу та зрозуміти його призначення.

AI generated text detected: ChatGPT 5.3, : 65,37%id: 28

Після натискання на картку користувач буде переходити на окрему сторінку обраної категорії, де буде відображатися перелік доступних закладів та можливість переходу до системи бронювання квитків. Під час створення макета також враховується адаптивність вебсайту для різних типів пристроїв. Розташування елементів інтерфейсу повинно автоматично змінюватися залежно від розміру екрана користувача, що забезпечить комфортне користування вебсайтом як на персональних комп'ютерах, так і на смартфоні або планшеті. Макет сторінки категорії виглядає ось так: Рисунок 2.8 Макет сторінки категорії сайту. Під час проектування даної сторінки особлива увага придлялася зручності перегляду списку доступних закладів та швидкому переходу до більш детальної інформації про кожний заклад. Структура сторінки проектується таким чином, щоб користувач міг швидко знайти потрібний заклад та перейти до оформлення бронювання. Під номером 1 фон сторінки. Він виконує роль контейнера для блоків в сторінки сайту. Фон сторінки планується робити в одному стилі для забезпечення єдиного стилістичного оформлення сайту. Використання єдиного фону для всіх сторінок дозволить створити єдиний цілісний дизайн вебсайту та покращує сприйняття інформації користувачем. Під номером 2 буде знаходитись назва категорії, яку вибрав користувач. Даний блок допоможе користувачеві швидко зрозуміти, в якому саме розділі сайту він буде знаходитись. Назва блоку буде зроблена великим та помітним шрифтом для покращення навігації та зручності користувача. Під номером 3 буде знаходитись кнопка повернення на головну сторінку сайту. Наявність такої кнопки надає можливість користувачеві швидко переходити між сторінками сайту та спрощує навігацію по ньому. Це особливо важливо, коли користувач хоче змінити категорію блоку в якому буде знаходитись. Під номером 4 знаходяться картки закладів в обраній категорії: музеї, зоопарки, замки. Саме вони виконують роль головних інформаційних елементів в сторінки та забезпечують компактне відображення інформації про доступні заклади. Використання карток дозволяє зробити структуру сторінки більш зрозумілою та сучасною. Під номером 5 буде знаходитись назва картки закладу. Назва дозволить швидко ідентифікувати заклад або знайти потрібний серед запропонованих. Назви планується робити м'явою оригіналу. Під номером 6 короткий опис закладу. У даному блоці буде розміщена коротка інформація про музеї, зоопарк або замок, що дозволить користувачу попередньо ознайомитися з особливостями обраного місця без необхідності переходу на окрему сторінку. Під номером 7 буде знаходитись кнопка для переходу на сторінку закладу для ознайомлення з ним більш детально. Під час проектування сторінки категорії також враховується адаптивність інтерфейсу для різних типів пристроїв. Розташування карток закладів в повинно автоматично змінюватися залежно від ширини екрана користувача, що забезпечить комфортне користування вебсайтом як на персональних комп'ютерах, так і на мобільних пристроях. Макет сторінки опису закладу виглядає ось так: Рисунок 2.9 Макет сторінки опису закладу. Дана сторінка призначена для детального ознайомлення користувача з вибраним закладом та подальшого переходу до системи бронювання квитків. Під час проектування сторінки основна увага придлялася зручному розташуванню інформації, простоті навігації та забезпеченню комфортного сприйняття контенту користувачем. Під номером 1 фон сторінки.

Він виконує роль контейнера для блоків в сторінки сайту. Фон сторінки планується робити в одному стилі для забезпечення єдиного стилістичного оформлення сайту. Використання єдиного фону для всіх сторінок дозволить створити єдиний цілісний дизайн вебсайту та покращує сприйняття інформації користувачем. У позиції 2 розташовано ідентифікатор вебресурсу (назву сайту), ключове призначення якого полягає в інформаційній підтримці та спрощенні візуальної навігації користувача. У подальших модифікаціях інтерфейсу допускається масштабування цієї області шляхом розміщення графічного логотипу чи допоміжних елементів керування. Під номером 3 буде знаходитись кнопка повернення на попередню сторінку сайту. Наявність такої кнопки надає можливість користувачеві швидко переходити між сторінками сайту та спрощує навігацію по ньому. Це суттєво важливо, коли користувач хоче повернутися до попереднього розділу вебсайту. Під номером 4 буде знаходитись картка закладу обрана користувачем, з більшою кількістю інформації. Даний блок є основним елементом сторінки та містить всю необхідну інформацію про заклад. У позиції 5 відображається автентична назва установи мовою оригіналу. Такий підхід забезпечує збереження автентичності об'єктів (музеїв, зоопарків, замкових комплексів тощо), а також оптимізує процес просторової орієнтації та пошуку локацій користувачем безпосередньо під час перебування в урбаністичному просторі міста Познань. Під номером 6 назва закладу українською або короткий опис. Даний блок допомагає користувачам краще зрозуміти призначення закладу та отримати коротку інформацію про нього без необхідності читання повного опису. Під номером 7 детальний опис закладу. У цьому блоці планується розміщення інформації про історію закладу, особливості екскурсій, доступні експозиції та інші важливі дані. Це дозволить користувачеві більш детально ознайомитися із закладом перед оформленням бронювання квитків. У позиції 8 інтегровано фотоматеріали досліджуваного об'єкта. Залучення автентичного візуального контенту оптимізує когнітивне сприйняття вебсторінки та сприяє формуванню у користувача цілісного уявлення про екстер'єрні характеристики локації. Візуалізація в даному контексті поєднує в собі інформативну та естетико-декоративну функції, що безпосередньо підвищує ергономічність, сучасність та атрактивність користувацького інтерфейсу. У позиції 9 локалізовано інтерактивний елемент (кнопку) для ініціалізації процедури бронювання квитків. Зазначений компонент є ключовим елементом цільової конверсії сторінки, оскільки забезпечує архітектурний перехід користувача до підсистеми оформлення транзакцій. З метою оптимізації користувацької уваги та мінімізації часу пошуку, даний керуючий елемент потребує високого рівня візуальної контрастності та інтуїтивної зрозумілості, а його просторове розміщення має гарантувати миттєвий доступ до функціоналу бронювання.

AI generated text detected: ChatGPT 5.3, : 60,56%id: 29

Під час проектування сторінки опису закладу також враховується адаптивність інтерфейсу для різних типів пристроїв. Усі елементи сторінки повинні коректно змінюватися

своє розташування залежно від розміру екрана користувача, що забезпечить комфортне користування вебсайтом як на персональних комп'ютерах, так і на смартфонах або планшетах. Макет сторінки бронювання виглядає ось так: Рисунок 2.10 Макет сторінки бронювання. Дана сторінка призначена для бронювання квитків. Під час проектування сторінки основна увага приділялася зручному розташуванню інформації, простоті навігації та забезпеченню комфортного сприйняття контенту користувачем. Інтерфейс сторінки повинен забезпечувати швидке та зрозуміле оформлення бронювання без виконання зайвих дій.

Усі основні елементи розташовуються у логічній послідовності для покращення взаємодії користувача із системою. Під номером 1 фон сторінки. Він виконує роль контейнера для блоків сторінки сайту. Фон сторінки планується робити в одному стилі для забезпечення єдиного стилістичного оформлення сайту. Застосування уніфікованого фонового рішення для всього масиву вебсторінок забезпечує формування цілісної візуально-графічної архітектури вебресурсу та інтенсифікує процеси перцептивного сприйняття інформації користувачем. Системна однорідність фонового простору нівелює когнітивний дисонанс при переході між розділами, оптимізуючи взаємодію суб'єкта з контентом. У позиції 2 розташовано ідентифікатор вебресурсу (назву сайту), ключове призначення якого полягає в інформаційній підтримці та спрощенні візуальної навігації користувача. У подальших модифікаціях інтерфейсу допускається масштабування цієї області шляхом розширення графічного логотипу чи допоміжних елементів керування. Позиція 3 відведена під елемент навігації (кнопку) повернення до попереднього стану інтерфейсу. Наявність цього компонента оптимізує реверсивну навігацію вебресурсу, мінімізує часові витрати користувача при міжсторінкових переходах та спрощує загальну архітектуру взаємодії, що є критично важливим за потреби оперативного повернення суб'єкта до попереднього інформаційного розділу. Позиція 4 містить інтерактивну форму бронювання, яка виступає функціональним ядром сторінки. Зазначений блок акумулює повний масив релевантної інформації, необхідної для оформлення квитків, та забезпечує мінімізацію ергономічних зусиль при введенні персональних даних. Позиція 5 репрезентує номінативний маркер установи мовою оригіналу. Таке інженерне рішення спрямоване на збереження автентичності власних назв історико-культурних та інфраструктурних об'єктів (музеїв, зоопарків, замкових комплексів тощо), а також оптимізує процеси просторової орієнтації та верифікації локацій користувачем під час перебування в урбаністичному середовищі міста Познань. Позиція 6 відведена під україномовний аналог найменування або лаконічний дескриптор об'єкта. Цей функціональний елемент забезпечує інформаційну валідацію, допомагаючи користувачеві верифікувати правильність вибору цільового закладу. Позиція 7 інтегрує текстові блоки експлікації (підписи полів), які виконують когнітивно-навігаційну функцію та координують дії користувача у суміжних зонах інтерфейсу (а саме спонукають до введення ідентифікаційних даних та вибору дати візиту). Використання експліцитних маркерів текстових полів підвищує семантичну прозорість форми та знижує ймовірність помилок дій оператора під час заповнення. Позиція 8 поєднує два дискретні текстові поля для введення персональних даних (імені та прізвища) з метою ініціалізації процедури резервування. Зазначені поля детерміновані як обов'язкові атрибути для генерації електронного документа (квитка) та забезпечення подальшої автентифікації користувача в системі. Позиція 9 представлена інтерфейсним компонентом типу *Time Picker* (вибір часу), за допомогою якого здійснюється селекція відповідного часового слота на основі динамічного моніторингу доступного квотного ліміту квитків. Позиція 10 містить елемент вибору календарної дати (*Date Picker*). Диференціація часових параметрів у формі окремого візуального компонента оптимізує моніторинг доступних варіантів та спрощує алгоритм фінального оформлення замовлення. Позиція 11 локалізує блок селекції категорій квитків. У межах цього модуля реалізовано можливість диференціації тарифних планів (зокрема, дорослий, дитячий, пільговий), що дозволяє системі в автоматичному режимі обчислювати інтегральну вартість замовлення на основі обраних пресетів. Позиція 12 функціонує як інформаційне поле класу квитка, призначене для візуалізації специфікацій та умов обраної категорії, що підвищує рівень поінформованості користувача під час прийняття рішень. Позиція 13 містить інтерактивне поле введення кількісних показників квитків для резервування. На рівні програмного забезпечення реалізовано модуль валідації вхідних значень для блокування некоректного або аномального введення даних. Позиція 14 репрезентує блок динамічного калькулювання сукупної вартості замовлених послуг. Цей елемент *UI*-дизайну забезпечує миттєвий зворотний зв'язок (візуалізацію підсумкової суми), що суттєво підвищує ергономічність та зручність експлуатації системи бронювання. Позиція 15 відведена під фінальний керуючий елемент (кнопку підтвердження), який завершує транзакційну процедуру та ініціює перехід до етапу верифікації або генерації вихідного електронного квитка. Задля оптимізації перцептивної уваги користувача даний елемент має виражений візуальний акцент (акцентний колір, розмір) на тлі інших компонентів сторінки. Додатково, у процесі архітектурного проектування сторінки бронювання було враховано вимоги інженерної психології щодо обов'язкової інтеграції модулів модальних сповіщень, системних попереджень та інформаційних тригерів для забезпечення безпомилкової взаємодії суб'єкта з вебплатформою. Наприклад, система повинна повідомляти про необхідність перевірки правильності введених даних, обмеження щодо дитячих квитків або відсутності вільних місць на обрану дату. Це дозволить підвищити надійність роботи системи та зменшити кількість помилок під час оформлення бронювання. Окрему увагу приділено адаптивності сторінки бронювання для різних типів пристроїв.

AI generated text detected: ChatGPT 5.3, : 62,6%

id: 30

Усі елементи форми повинні коректно змінювати своє розташування залежно від ширини екрана користувача, що забезпечить комфортне користування системою бронювання як на персональних комп'ютерах, так і на смартфонах або планшетах. Макет сторінки для перевірки інформації виглядає ось так: Рисунок 2.11 Макет сторінки для перевірки інформації. Під номером 1 фон сторінки. Він виконує роль контейнера для блоків сторінки сайту. Фон сторінки планується робити в одному стилі для забезпечення єдиного стилістичного оформлення сайту. Використання єдиного фону для всіх сторінок дозволить створити єдиний цілісний дизайн вебсайту та покращує сприйняття інформації користувачем. Під номером 2 буде знаходитися форма для перевірки інформації введеної користувачем. Даний блок є основним елементом сторінки та містить інформацію яку ввв в користувач для бронювання квитків. Під номером 3 назва блоку. Під номером 4 назва музею. Під номером 5 ім'я та прізвище написано англійською. Під номером 6 час на який буде бронюватися квиток.

Під номером 7 дата на яку буде бронюватися квиток. Під номером 8 номер групи в якій буде заброньовано квиток. Під номером 9 буде кількість та тип квитків в які планується забронювати. Під номером 10 загальна вартість за квитки. Під номером 11 кнопка бронювати. Наступна сторінка це сторінка підтвердження бронювання, вона буде такою самою як і для перевірки інформації тільки там вже буде кнопка, яка дозволить скачати в PDF формат квиток, який забронював користувач. СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ Розробка бази даних. Представлена структура бази даних: описує

інформаційну систему бронювання квитків для музеїв або інших туристичних/екскурсійних об'єктів. Основне призначення системи – забезпечення зручного процесу вибору локації, типу квитка, часу виведення та створення бронювання користувачем. Система побудована за реляційною моделлю даних та складається з декількох взаємопов'язаних таблиць. Основні функції системи: збергання інформації про музеї та місця виведення; підлітків за категоріями; управління типами квитків; контроль часу екскурсій; обмеження максимальної кількості виведувачів; створення та облік бронювань; зв'язок між квитками, музеями та групами виведення. Для проектування структури бази даних системи бронювання квитків було використано ER-модель рис 3.1 (Entity-Relationship Model). Дана модель дозволяє описати основні сутності системи, їх атрибути та зв'язки між ними. На основі ER-моделі була побудована реляційна структура бази даних, яка складається із взаємопов'язаних таблиць. ER-діаграма містить основні сутності системи: booking, excursion_group, place, place_type та ticket_type, а також первинні та зовнішні ключі, що забезпечують цілісність і коректність зв'язків між даними. Використання ER-моделі дозволило логічно організувати структуру даних, уникнути дублювання інформації та забезпечити масштабованість системи бронювання. Рисунок 3.1 ER-модель бази даних У системі присутні 5 основних таблиць: booking excursion_group place place_type ticket_type Таблиця booking є центральною таблицею системи. Вона зберігає інформацію про всі створені бронювання. Таблиця booking Поле Тип Призначення id_booking Int Унікальний ідентифікатор бронювання customer_name Varchar Ім'я користувача customer_surname Varchar Прізвище користувача id_group Int Посилання на групу екскурсій id_ticket Int Посилання на тип квитка ticket_count Int Кількість квитків в visit_time Time Обраний час виведення booking_date Date Дата виведення created_at Timestamp Дата та час створення бронювання Таблиця booking (Таблиця 3.1) призначена для збергання інформації про бронювання квитків користувачами на виведення музею або екскурсій. Вона містить дані про користувача, тип квитка, кількість придбаних квитків, дату та час виведення, а також дату створення бронювання. Кожен запис у таблиці є окремим бронюванням і має унікальний ідентифікатор. Поле 'id_group' використовується для зв'язку з таблицею груп екскурсій, а 'id_ticket' — для зв'язку з таблицею типів квитків. Поле 'created_at' автоматично фіксує дату та час створення запису, що дозволяє вести облік усіх бронювань у системі. Таблиця booking забезпечує: реєстрацію користувачів; створення бронювання; зв'язок із конкретною екскурсією; зв'язок із типом квитка; контроль кількості квитків; можливість ведення статистики; перевірку доступності місць. Первинний ключ у таблиці booking є поле id_booking. Первинний ключ використовується для: унікальної ідентифікації бронювання; швидкого пошуку записів; створення зв'язків між таблицями. Зовнішні ключі таблиці booking є поля id_group та id_ticket. Поле id_group посилається на таблицю excursion_group. Функція: визначає групу або часовий слот екскурсій; дозволяє контролювати кількість місць. Поле id_ticket посилається на таблицю ticket_type. Функція: визначає тип квитка; дозволяє використовувати різні ціни. Таблиця excursion_group відповідає за збергання інформації про часові групи екскурсій. Кожен запис у таблиці представляє окремий часовий слот для проведення екскурсії у певному місці виведення. Це дозволяє організувати екскурсії за розкладом та контролювати кількість учасників у кожній групі. Поля таблиці (Таблиця 3.2) Таблиця 3.2 Таблиця excursion_group Поле Тип Призначення id_group Int Унікальний номер групи id_place Int Посилання на місце виведення excursion_time Time Час проведення екскурсії max_people Int Максимальна кількість людей Таблиця забезпечує: управління часовими слотами; обмеження кількості виведувачів; контроль перевантаження музею; планування екскурсій; можливість створення декількох сеансів на день. Первинний ключ таблиці excursion_group є поле id_group Використовується для: ідентифікації екскурсійної групи; створення зв'язку з бронюваннями. Зовнішній ключ таблиці excursion_group є поле id_place і посилається він на таблицю place. Функція: визначає музей або локацію; пов'язує часовий слот із конкретним місцем. Таблиця place містить інформацію про місця виведення. Поля таблиці (Таблиця 3.3) Таблиця 3.3 Таблиця place Поле Тип Призначення id_place Int Унікальний номер місця id_type Int Посилання на тип місця name Varchar Назва музею або локації Функціонал таблиці: збергання списку музеїв; групування за категоріями; зв'язок із квитками; зв'язок із часовими слотами. Первинний ключ таблиці place є поле id_place. Ключ дозволяє: ідентифікувати музей; зв'язувати інші таблиці з конкретним місцем. Зовнішній ключ таблиці place є поле id_type він посилається на таблицю place_type. Функція зовнішнього ключа полягає в визначенні категорії місця. Таблиця place_type зберігає типи або категорії місць. Поля таблиці (Таблиця 3.4) Таблиця 3.4 Таблиця place_type Поле Тип Призначення id_type Int Унікальний номер типу type_name Varchar Назва категорії Таблиця дозволяє: фільтрувати місця; групувати музеї; спрощувати навігацію; створювати категорії на сайті. Таблиця ticket_type містить інформацію про типи квитків. Поля таблиці (Таблиця 3.5) Таблиця 3.5 Таблиця ticket_type Поле Тип Призначення id_ticket Int Унікальний номер квитка id_place Int Посилання на музей ticket_name Varchar Назва квитка price Decimal Вартість квитка Таблиця забезпечує: різні типи квитків; різні ціни; підтримку пільг; гнучке управління тарифами. Приклади типів квитків: дорослий; пільговий; дитячий/молодь; Зовнішній ключ таблиці ticket_type це поле id_place він посилається на таблицю place. Функція зовнішнього ключа, визначає музей, для якого діє квиток. Аналіз

 Знайдено плагіату: 0,02% https://metod.suitt.edu.ua/download/992	id: 31
зв'язк м між таблицями place_type → place Тип зв'язку: Один до багатьох (1:N) Один тип м сця м же м стити багато муз ів. Приклад: Категор я	
 Знайдено посилань: 0%	id: 32
“муз й”	
 Знайдено плагіату: 0,06% https://metod.suitt.edu.ua/download/992	id: 33
мож е м стити: Muzeum Narodowe; Muzeum Archeologiczne; Muzeum P u y. place → ticket_type Тип зв'язку: Один до багатьох (1:N) Один муз й мож е мати багато тип в квитк в. place → excursion_group Тип зв'язку: Один до багатьох (1:N) Один муз й мож е мати багато час ових сл іт в. excursion_group → booking Тип зв'язку: Один до багатьох (1:N) Одна екскурс йна група мож е м стити багато бронювань	
ticket_type → booking Тип зв'язку: Один до багатьох (1:N) Один тип квитка мож е використовуватись у багатьох бронюваннях. Розробка сайту Розробка вебсайту системи бронювання квитків здійснювалася з метою створення зручного та сучасного сервісу для перегляду інформації про музеї, вибору квитків і оформлення бронювання в онлайн-режимі. Під час створення вебсайту основна увага придлялася простоті використання; зрозумілому інтерфейсу; швидкій навігації; адаптивному дизайну; зручності оформлення бронювання. Для реалізації клієнтської частини вебсайту були використані технології HTML, CSS та JavaScript. HTML використовувався для створення структури сторінок вебсайту та розміщення основних елементів інтерфейсу, таких як: навігаційне меню; картки музеїв; форми бронювання; кнопки взаємодії; інформаційні блоки. CSS застосовувався для стилізації сторінок та створення сучасного дизайну. За допомогою CSS було реалізовано: кольорове оформлення; адаптивне розташування елементів; оформлення кнопок і карток; анімації та ефекти наведення; структурування контенту. JavaScript використовувався для	

реалізації динамічного функціоналу вебсайту. За допомогою JavaScript було реалізовано: завантаження інформації про музеї; обробку параметрів URL; вибір типів квитків; підрахунок вартості замовлення; перевірку введених даних; передачу інформації між сторінками; генерацію електронного квитка. Інформація про музеї зберігається у вигляді JavaScript-об'єкта, що містить: назву музею; короткий опис; детальний опис; зображення; типи квитків; вартість квитків. На головній сторінці вебсайту (Рис. 3.2) розташовані основні категорії доступних місць для відвідування, а також навігаційні елементи системи. Рисунок 3.2 Головна сторінка вебсайту. Інтерфейс сторінки розроблено у сучасному стилі з використанням карток категорій та адаптивного дизайну. На рисунку 3.3 зображено сторінку вибору музеїв. Дана сторінка призначена для перегляду доступних музеїв та переходу до детальної інформації про кожен об'єкт. У верхній частині сторінки розташована навігаційна панель, яка містить кнопку повернення на попередню сторінку та назву поточного розділу. Рисунок 3.3 Сторінка вибору музеїв. Основна частина сторінки реалізована у вигляді адаптивної сітки карток. Кожна картка музею містить: назву музею; короткий опис; кнопку переходу до детальної сторінки. На сторінці представлені різні музеї: Познань, зокрема художні, археологічні, військові та тематичні музеї. Для покращення зручності користування було реалізовано компактне розташування елементів, що дозволяє швидко переглядати доступні варіанти. Дизайн сторінки виконаний у сучасному стилі з використанням градієнтного фону, карткового інтерфейсу та інтерактивних кнопок. Кнопка

🔍 Знайдено посилань: 0%

id: 34

«Дальше»

забезпечує перехід до сторінки конкретного музею, де користувач може переглянути розширену інформацію та перейти до оформлення бронювання квитків. Для кожного музею реалізовано окрему сторінку з детальною інформацією (Рис 3.4). На сторінці відображаються: Рисунок 3.4 Сторінка одного з музеїв. Фотографія музею; назва; короткий опис; розширена інформація; доступні типи квитків; кнопка переходу до бронювання. Система бронювання (Рис 3.5) дозволяє користувачу: обрати музей; вибрати тип квитка; вказати кількість квитків; вибрати дату та час відвідування; переглянути загальну вартість замовлення; підтвердити бронювання. Для підвищення зручності користування було реалізовано адаптивний дизайн, який дозволяє коректно відобразити вебсайт як на персональних комп'ютерах, так і на мобільних пристроях. Рисунок 3.5 Сторінка системи бронювання. Під час розробки структури даних вебсайту було використано ER-модель (Entity-Relationship Model), яка дозволила описати основні сутності системи, їх атрибути та зв'язки між ними. На основі ER-моделі була створена реляційна структура бази даних для збергання інформації про музеї, квитки, часові слоти та бронювання користувачів. У результаті розробки було створено вебсайт, який забезпечує зручний процес онлайн-бронювання квитків, швидкий доступ до інформації про музеї та сучасний інтерфейс взаємодії з користувачем. ВИСНОВОК. У процесі виконання кваліфікаційної роботи було проведено аналіз сучасних підходів до створення систем онлайн-бронювання квитків, досліджено особливості існуючих програмних рішень та визначено основні проблеми, які виникають під час організації електронного резервування культурно-розважальних заходів.

🚫 AI generated text detected: ChatGPT 5.3, : 74,35%

id: 35

Особливу увагу було приділено питанням зручності користування, швидкодії системи, безпеки передачі даних та адаптивності вебресурсів для різних типів пристроїв. У результаті виконаної роботи було спроектовано та реалізовано вебсистему бронювання квитків для музеїв, замків та зоопарків міста Познань. Створена система дозволяє користувачам переглядати інформацію про доступні заклади, обирати дату та час відвідування, тип квитка, кількість місць, а також оформлювати бронювання з можливістю отримання електронного квитка у форматі PDF. Під час розробки було реалізовано структуру бази даних, яка забезпечує коректне збергання інформації про заклади, типи квитків, екскурсійні групи та бронювання користувачів. Під час роботи було використано сучасні вебтехнології HTML, CSS, JavaScript, PHP та MySQL, які дозволили створити функціональний та зрозумілий інтерфейс користувача. Також було враховано принципи адаптивного дизайну із забезпеченням коректної роботи вебсайту як на персональних комп'ютерах, так і на мобільних пристроях. Окрему увагу звернено у бік легкості структури сторінок, простоти навігації та мінімізації кількості дій, необхідних для оформлення бронювання. Під час виконання кваліфікаційної роботи було підтверджено, що автоматизація процесу бронювання квитків значно спрощує взаємодію користувачів із культурними закладами, зменшує навантаження на каси та дозволяє швидко отримувати актуальну інформацію про доступність квитків. Розроблена система є особливо актуальною для українців, які перебувають у Польщі та потребують зрозумілого українськомовного сервісу для планування культурного відпочинку та екскурсій. Практична цінність роботи полягає у створенні готового вебпроекту, який може бути використаний як основа для подальшого розвитку повноцінної системи онлайн-бронювання.

У перспективі система може бути доповнена інтеграцією з платіжними сервісами, системою реєстрації користувачів, електронною поштою для автоматичного надсилання квитків, а також механізмами адміністрування закладів та статистики відвідувань. Отже, поставлена мета кваліфікаційної роботи була досягнута, а всі основні завдання, пов'язані з аналізом, проектуванням та створенням вебсистеми бронювання квитків, виконані у повному обсязі. СПИСОК ДЖЕРЕЛ. Укрзалізниця ввела бронювання квитків через Інтернет [Електронний ресурс] // [Finance.ua](#). – Режим доступу: [Finance.ua Historia firm](#)у [Електронний ресурс] // [PKP Intercity](#). – Режим доступу: [PKP Intercity Sabre \(travel reservation system\)](#) [Електронний ресурс] // [Wikipedia](#). – Режим доступу: [Wikipedia – Sabre Amadeus IT Group](#) [Електронний ресурс] // [Wikipedia](#). – Режим доступу: [Wikipedia – Amadeus IT Group Sublime Text](#) [Електронний ресурс] // [Wikipedia](#). – Режим доступу: [Wikipedia – Sublime Text](#) Що таке PHP? [Електронний ресурс] // [FreeHost](#). – Режим доступу: [FreeHost – PHP](#) Що таке JavaScript і для чого він потрібен [Електронний ресурс] // [GoIT](#). – Режим доступу: [GoIT – JavaScript](#) Google Chrome [Електронний ресурс] // [Wikipedia](#). – Режим доступу: [Wikipedia – Google Chrome HTML: HyperText Markup Language](#) [Електронний ресурс] // [Mozilla Developer Network](#). – Режим доступу: [MDN – HTML CSS: Cascading Style Sheets](#) [Електронний ресурс] // [Mozilla Developer Network](#). – Режим доступу: [MDN – CSS JavaScript Guide](#) [Електронний ресурс] // [Mozilla Developer Network](#). – Режим доступу: [MDN – JavaScript Guide PHP Manual](#) [Електронний ресурс] // [PHP Documentation Group](#). – Режим доступу: [PHP Manual MySQL Documentation](#) [Електронний ресурс] // [Oracle](#). – Режим доступу: [MySQL Documentation XAMPP Installers and Downloads](#) [Електронний ресурс] // [Apache Friends](#). – Режим доступу: [XAMPP Official Website](#) Figma: the collaborative interface design tool [Електронний ресурс] // [Figma](#). – Режим доступу: [Figma Official Website](#) jsPDF – JavaScript PDF generation library [Електронний ресурс]. – Режим доступу: [jsPDF Official Website](#) Додаток А. Код головної сторінки сайту !DOCTYPE html html lang=

🔍 Знайдено посилань: 0%

id: 36

" "

head meta charset=	
Знайдено посилань: 0%	id: 37
"UTF-8"	
title TicketWave /title link rel=	
Знайдено посилань: 0%	id: 38
"stylesheet"	
href=	
Знайдено посилань: 0%	id: 39
"style.css"	
/head body header class=	
Знайдено посилань: 0%	id: 40
"header"	
!-- Верхній банер (шапка сайту) -- h1 TicketWave /h1 !-- Головний заголовок сайту -- p Швидке бронювання квитків у музеї, зоопарки та замки Познані /p !-- Підзаголовок з описом сервісу -- /header section class=	
Знайдено посилань: 0%	id: 41
"menu"	
!-- Основний блок меню з картками -- a href=	
Знайдено посилань: 0%	id: 42
"muzeums.html"	
class=	
Знайдено посилань: 0%	id: 43
"card"	
!-- Картка з посиланням на сторінку музеїв -- div class=	
Знайдено посилань: 0%	id: 44
"icon"	
/div h3 Музеї /h3 p Обери музей та забронюй квиток /p /a div class=	
Знайдено посилань: 0%	id: 45
"card disabled"	
!-- Неактивна картка (замки) -- div class=	
Знайдено посилань: 0%	id: 46
"icon"	
/div h3 Замки /h3 /div div class=	
Знайдено посилань: 0%	id: 47
"card disabled"	
!-- Неактивна картка (зоопарки) -- div class=	
Знайдено посилань: 0%	id: 48
"icon"	
/div h3 Зоопарки /h3 /div /section footer © 2026 TicketWave /footer /body /html Додаток Б. Код сторінки музеїв !DOCTYPE html !-- Оголошення HTML5 документа -- html lang=	
Знайдено посилань: 0%	id: 49
"<"	
!-- Початок HTML сторінки, мова українська -- head !-- Початок блоку метаданих -- meta charset=	
Знайдено посилань: 0%	id: 50
"UTF-8"	
!-- Кодування символів UTF-8 -- title TicketWave Музеї /title !-- Заголовок вкладки браузера -- link rel=	
Знайдено посилань: 0%	id: 51
"stylesheet"	
href=	
Знайдено посилань: 0%	id: 52
"style.css"	
!-- Підключення стилів -- /head !-- Кінець head -- body !-- Початок видимого вмісту сторінки -- header class=	
Знайдено посилань: 0%	id: 53
"topbar"	
!-- Верхня панель сайту -- a href=	
Знайдено посилань: 0%	id: 54
"index.html"	
class=	
Знайдено посилань: 0%	id: 55
"back"	
Назад /a !-- Кнопка повернення на головну -- h2 Музеї Познані /h2 !-- Заголовок сторінки -- /header !-- Кінець верхньої панелі -- main class=	
Знайдено посилань: 0%	id: 56
"container"	
!-- Основний контейнер сторінки -- section class=	
Знайдено посилань: 0%	id: 57
"grid"	
!-- Сітка з картками музеїв -- div class=	
Знайдено посилань: 0%	id: 58
"card"	
!-- Картка музею -- h3 Muzeum Narodowe w Poznaniu /h3 !-- Назва музею -- p Головний художній музей міста Познань. /p !-- Опис музею -- a href=	
Знайдено посилань: 0%	id: 59

"museum.htm?id=1"	
class=	
 Знайдено посилань: 0%	id: 60
"img"	
Детальніше /a!-- Посилання на сторінку музею -- /div!-- Кінець картки -- div class=	
 Знайдено посилань: 0%	id: 61
"img"	
!-- Картка музею -- h3 Muzeum Sztuk Użytkowych /h3!-- Назва музею -- p Музей декоративного мистецтва. /p!-- Опис музею -- a href=	
 Знайдено посилань: 0%	id: 62
"museum.htm?id=2"	
class=	
 Знайдено посилань: 0%	id: 63
"img"	
Детальніше /a!-- Посилання -- /div!-- Кінець картки -- div class=	
 Знайдено посилань: 0%	id: 64
"img"	
!-- Картка музею -- h3 Muzeum Uzbrojenia /h3!-- Назва музею -- p Музей військової техніки та історії. /p!-- Опис музею -- a href=	
 Знайдено посилань: 0%	id: 65
"museum.htm?id=3"	
class=	
 Знайдено посилань: 0%	id: 66
"img"	
Детальніше /a!-- Посилання -- /div!-- Кінець картки -- div class=	
 Знайдено посилань: 0%	id: 67
"img"	
!-- Картка музею -- h3 Muzeum Archeologiczne /h3!-- Назва музею -- p Музей археологічних стародавніх цивілізацій. /p!-- Опис музею -- a href=	
 Знайдено посилань: 0%	id: 68
"museum.htm?id=4"	
class=	
 Знайдено посилань: 0%	id: 69
"img"	
Детальніше /a!-- Посилання -- /div!-- Кінець картки -- div class=	
 Знайдено посилань: 0%	id: 70
"img"	
!-- Картка музею -- h3 Muzeum Instrumentów Muzycznych /h3!-- Назва музею -- p Музей музичних інструментів світу. /p!-- Опис музею -- a href=	
 Знайдено посилань: 0%	id: 71
"museum.htm?id=5"	
class=	
 Знайдено посилань: 0%	id: 72
"img"	
Детальніше /a!-- Посилання -- /div!-- Кінець картки -- div class=	
 Знайдено посилань: 0%	id: 73
"img"	
!-- Картка музею -- h3 Centrum Szyfrów Enigma /h3!-- Назва музею -- p Музей криптографії та шифрів. /p!-- Опис музею -- a href=	
 Знайдено посилань: 0%	id: 74
"museum.htm?id=6"	
class=	
 Знайдено посилань: 0%	id: 75
"img"	
Детальніше /a!-- Посилання -- /div!-- Кінець картки -- div class=	
 Знайдено посилань: 0%	id: 76
"img"	
!-- Картка музею -- h3 Rogalowe Muzeum /h3!-- Назва музею -- p Музей кулінарної історії Познані. /p!-- Опис музею -- a href=	
 Знайдено посилань: 0%	id: 77
"museum.htm?id=7"	
class=	
 Знайдено посилань: 0%	id: 78
"img"	
Детальніше /a!-- Посилання -- /div!-- Кінець картки -- div class=	
 Знайдено посилань: 0%	id: 79
"img"	
!-- Картка музею -- h3 Muzeum Broni Pancernej /h3!-- Назва музею -- p Музей танків та бронетехніки. /p!-- Опис музею -- a href=	
 Знайдено посилань: 0%	id: 80
"museum.htm?id=8"	
class=	
 Знайдено посилань: 0%	id: 81
"img"	
Детальніше /a!-- Посилання -- /div!-- Кінець картки -- div class=	

Знайдено посилань: 0%	id: 82
"card"	
!-- Картка музею -- h3 Музей Руху /h3 !-- Назва музею -- p Музей картоплі /p !-- Опис музею -- a href=	
Знайдено посилань: 0%	id: 83
"museum.html?id=9"	
class=	
Знайдено посилань: 0%	id: 84
"img"	
Детальше /a !-- Посилання -- /div !-- Кінець картки -- /section !-- Кінець сітки -- /main !-- Кінець основного контейнера -- footer !-- Футер сторінки -- © 2026 TicketWave !-- Авторські права -- /footer !-- Кінець футера -- /body !-- Кінець body -- /html !-- Кінець HTML документа -- Додаток В. Код сторінки музею !DOCTYPE html !-- Оголошення HTML5 документа -- html lang=	
Знайдено посилань: 0%	id: 85
"uk"	
!-- Початок HTML сторінки, мова українська -- head !-- Початок блоку метаданих -- meta charset=	
Знайдено посилань: 0%	id: 86
"UTF-8"	
!-- Кодування сторінки UTF-8 -- title Музей /title !-- Заголовок вкладки браузера -- link rel=	
Знайдено посилань: 0%	id: 87
"stylesheet"	
href=	
Знайдено посилань: 0%	id: 88
"style.css"	
!-- Підключення CSS стилів -- /head !-- Кінець head -- body !-- Початок видимого вмісту сторінки -- header class=	
Знайдено посилань: 0%	id: 89
"topbar"	
!-- Верхня панель -- a href=	
Знайдено посилань: 0%	id: 90
"museums.html"	
← Назад /a !-- Кнопка повернення до списку музеїв -- h2 TicketWave /h2 !-- Назва сайту в шапці -- /header !-- Кінець шапки -- div id=	
Знайдено посилань: 0%	id: 91
"content"	
/div !-- Контейнер, куди JS вставляє інформацію про музей -- script src=	
Знайдено посилань: 0%	id: 92
"museums.js"	
/script !-- Підключення файлу з даними музеїв -- script !-- Початок JavaScript логіки -- const id = new URLSearchParams(location.search).get(	
Знайдено посилань: 0%	id: 93
"id"	
); !-- Отримання id музею з URL (наприклад ?id=1) -- const m = museums[id]; !-- Пошук музею в об'єкті museums за отриманим id -- if (!m) { !-- Перевірка: якщо музей не знайдено -- document.getElementById(	
Знайдено посилань: 0%	id: 94
"content"	
).innerHTML =	
Знайдено посилань: 0%	id: 95
"<h2>Не знайдено / 2"	
; !-- Вивід повідомлення про помилку -- } else { !-- Якщо музей знайдено -- document.getElementById(	
Знайдено посилань: 0%	id: 96
"content"	
).innerHTML = ` !-- Вставка HTML-коду динамічно -- div class=	
Знайдено посилань: 0%	id: 97
"card big"	
!-- Велика картка музею -- h1 \${m.name} /h1 !-- Назва музею з даних -- p \${m.desc} /p !-- Короткий опис музею -- div class=	
Знайдено посилань: 0%	id: 98
"full-text"	
\${m.fullDesc} /div !-- Опис музею -- img src=	
Знайдено посилань: 0%	id: 99
"\${m.image}"	
class=	
Знайдено посилань: 0%	id: 100
"museum-img"	
a class=	
Знайдено посилань: 0%	id: 101
"big"	
href=	
Знайдено посилань: 0%	id: 102
"prev.html?m=\${id}"	
🔒 Бронювати /a !-- Кнопка бронювання -- /div `; } !-- Кінець умови -- /script !-- Кінець JavaScript -- /body !-- Кінець тіла сторінки -- /html !-- Кінець HTML документа -- Додаток Г.	

Код сторінки бронювання !DOCTYPE html html lang=	Знайдено посилань: 0%	id: 103
"uk"		
head meta charset=	Знайдено посилань: 0%	id: 104
"UTF-8"		
title Бронювання /title link rel=	Знайдено посилань: 0%	id: 105
"stylesheet"		
href=	Знайдено посилань: 0%	id: 106
"style.css"		
/head body header class=	Знайдено посилань: 0%	id: 107
"topbar"		
!-- Верхня панель -- a href=	Знайдено посилань: 0%	id: 108
"museums.htm"		
← Назад /a !-- Кнопка повернення -- h2 Бронювання /h2 !-- Заголовок сторінки -- /header div class=	Знайдено посилань: 0%	id: 109
"card big"		
id=	Знайдено посилань: 0%	id: 110
"app"		
/div !-- Основний контейнер, куди JS вставляє форму -- script src=	Знайдено посилань: 0%	id: 111
"museums.js"		
/script !-- Підключення бази музеїв -- script !-- Початок JavaScript -- const id = new URLSearchParams(location.search).get(	Знайдено посилань: 0,01%	id: 112
"id"); const m = museums[id]		
; let total = 0; const app = document.getElementById(	Знайдено посилань: 0%	id: 113
"app"		
); if (!m) { app.innerHTML =	Знайдено посилань: 0%	id: 114
"Музей не знайдено"		
; } else { app.innerHTML = ` h2 \${m.name} /h2 p \${m.desc} /p div class=	Знайдено посилань: 0%	id: 115
"form"		
label Ім'я: /label input type=	Знайдено посилань: 0%	id: 116
"text"		
id=	Знайдено посилань: 0%	id: 117
"firstName"		
label Прізвище: /label input type=	Знайдено посилань: 0%	id: 118
"text"		
id=	Знайдено посилань: 0%	id: 119
"lastName"		
/div label Оберіть час: /label select id=	Знайдено посилань: 0%	id: 120
"visitTime"		
option value=	Знайдено посилань: 0%	id: 121
""		
-- Оберіть час -- /option option value=	Знайдено посилань: 0%	id: 122
"10:00"		
10:00 /option option value=	Знайдено посилань: 0%	id: 123
"13:00"		
13:00 /option option value=	Знайдено посилань: 0%	id: 124
"16:00"		
16:00 /option /select label Оберіть дату: /label input type=	Знайдено посилань: 0%	id: 125
"date"		
id=	Знайдено посилань: 0%	id: 126
"visitDate"		

div id=	Знайдено посилань: 0%	id: 127
"tickets"		
/div h3 id=	Знайдено посилань: 0%	id: 128
"sum"		
Сума: 0 /h3 button class=	Знайдено посилань: 0%	id: 129
"btn"		
onclick=	Знайдено посилань: 0%	id: 130
"goToPayment()"		
Перейти до оплати /button`; const box = document.getElementById(	Знайдено посилань: 0%	id: 131
"tickets"		
); m.tickets.forEach(t => { box.innerHTML += `div class=	Знайдено посилань: 0%	id: 132
"ticket"		
span \${t.name} (\${t.price})/span input type=	Знайдено посилань: 0%	id: 133
"number"		
min=	Знайдено посилань: 0%	id: 134
"0"		
value=	Знайдено посилань: 0%	id: 135
"0"		
data-price=	Знайдено посилань: 0%	id: 136
"\${t.price}"		
data-name=	Знайдено посилань: 0%	id: 137
"\${t.name}"		
/div`; }); document.addEventListener(	Знайдено посилань: 0%	id: 138
"input",		
() => { total = 0; let count = 0; document.querySelectorAll(	Знайдено посилань: 0%	id: 139
"input[type='number']"		
).forEach(i => { const val = Number(i.value) 0; count += val; total += val *		
Number(i.dataset.price); }); if (count > 15) { alert(	Знайдено посилань: 0,01%	id: 140
"Максимум 15 квитків в одній групі"		
); document.activeElement.value = 0; return; } document.getElementById(	Знайдено посилань: 0%	id: 141
"sum"		
).innerText = "Сума: " + total + "	Знайдено посилань: 0,01%	id: 142
"; }); // ГЛАВНАЯ ФУНКЦИЯ function goToPayment() { const firstName =		
document.getElementById("		
firstName	Знайдено посилань: 0,01%	id: 143
").value.trim(); const lastName = document.getElementById("		
lastName	Знайдено посилань: 0,01%	id: 144
").value.trim(); let time = document.getElementById("		
visitTime	Знайдено посилань: 0,01%	id: 145
").value; if (time && time.length === 5) { time += ":		
00	Знайдено посилань: 0,01%	id: 146
"; } const date = document.getElementById("		
visitDate	Знайдено посилань: 0,01%	id: 147
").value; if (!firstName !lastName) { alert("Введіть ім'я та прізвище"	Знайдено посилань: 0,01%	id: 148
"); return; } if (!date) { alert("Введіть дату"		
Оберіть дату	Знайдено посилань: 0,02%	id: 149
"); return; } const selectedDate = new Date(date); if (selectedDate.getDay() === 0) { alert("Неділю музеї не працює"		
"); return; } let tickets = []; let count = 0; document.querySelectorAll("input[type='number']")	Знайдено посилань: 0,05%	id: 150
).forEach(i => { if (Number(i.value) > 0) { tickets.push({ name: i.dataset.name, count:		

Number (.value))); count += Number (.value); } }); if (count 15) { alert ("	
Максимум 15 квитків на групу	
Знайдено посилань: 0,01%	id: 151
"); return; } const assignedGroup = Number (document.getElementById("	
visitTime	
Знайдено посилань: 0,02%	id: 152
".selectedindex); location.href = `payment.html?museum=\${encodeURIComponent(m.name)}&total=\${encodeURIComponent(total)}&tickets=\${encodeURIComponent(tickets)}&time=\${encodeURIComponent(time)}&date=\${encodeURIComponent(date)}&group=\${encodeURIComponent(assignedGroup)}`; = `\${.name} x \${.count}`.join(",	
Знайдено посилань: 0,03%	id: 153
"))&time=\${encodeURIComponent(time)}&date=\${encodeURIComponent(date)}&group=\${encodeURIComponent(assignedGroup)}`; `/script /css /html Додатковий код стилів і скриптів!DOCTYPE html html lang="	
uk	
Знайдено посилань: 0%	id: 154
"head meta charset="	
UTF-8" title Оплата /title link rel="stylesheet" href="	
Знайдено посилань: 0%	id: 155
"style.css"	
/head body div class="	
Знайдено посилань: 0%	id: 156
"card big"	
id="	
Знайдено посилань: 0%	id: 157
"box"	
/div script const p = new URLSearchParams(location.search); const museum = p.get(	
Знайдено посилань: 0%	id: 158
"museum"	
); const total = p.get(	
Знайдено посилань: 0%	id: 159
"total"	
); const tickets = p.get(	
Знайдено посилань: 0%	id: 160
"tickets"	
); const time = p.get(	
Знайдено посилань: 0%	id: 161
"time"	
); const date = p.get(	
Знайдено посилань: 0%	id: 162
"date"	
); const group = p.get(	
Знайдено посилань: 0%	id: 163
"group"	
); const id = p.get(	
Знайдено посилань: 0%	id: 164
"id"	
); const firstName = p.get(	
Знайдено посилань: 0%	id: 165
"firstName"	
); const lastName = p.get(	
Знайдено посилань: 0%	id: 166
"lastName"	
); if (!museum) { document.body.innerHTML =	
Знайдено посилань: 0%	id: 167
"Немає даних"	
; } else { document.getElementById(	
Знайдено посилань: 0%	id: 168
"box"	
).innerHTML = `h2 Бронювання /h2 p b \${museum} /p p \${firstName} \${lastName} /p p \${time} /p p \${date} /p p Група № \${group} /p p \${tickets} /p h3 \${total} зл /h3 button class="	
Знайдено посилань: 0%	id: 169
"btn"	
onclick="	
Знайдено посилань: 0%	id: 170
"pay()"	
Бронювати /button`; } function pay() { const p = new URLSearchParams(location.search); const firstName = p.get(	
Знайдено посилань: 0%	id: 171
"firstName"	
); const lastName = p.get(	
Знайдено посилань: 0%	id: 172
"lastName"	
); const date = p.get(	
Знайдено посилань: 0%	id: 173
"date"	

```
); const time = p.get(
Знайдено посилань: 0% id: 174
"time"
); const group = p.get(
Знайдено посилань: 0% id: 175
"group"
); const id = p.get(
Знайдено посилань: 0% id: 176
"id"
); const museum = p.get(
Знайдено посилань: 0% id: 177
"museum"
); const total = p.get(
Знайдено посилань: 0% id: 178
"total"
); const tickets = p.get(
Знайдено посилань: 0% id: 179
"tickets"
); console.log(
Знайдено посилань: 0% id: 180
"PAY CLICKED"
); if (!firstName || !lastName || !date || !id) { alert(
Знайдено посилань: 0% id: 181
"Missing data"
); return; } let ticketData = []; if (tickets) { tickets.split(",").forEach(t => { const parts =
t.trim().split(" x
Знайдено посилань: 0,04% id: 182
"); ticketData.push({ name: parts[0], count: Number(parts[1] || 1) }); }); } const payLoad = {
name: firstName, surname: lastName, date: date, time: time, group: Number(group), id_place:
Number(id), tickets: ticketData }; console.log("
SENDING:
Знайдено посилань: 0,01% id: 183
", payLoad); console.log(time); letId("
save.php
Знайдено посилань: 0% id: 184
", { method: "
POST
Знайдено посилань: 0% id: 185
", headers: { "
Content-Type": "application/json" }, body: JSON.stringify(payload) }) .then(res => res.json())
.then(data => { console.log(
Знайдено посилань: 0% id: 186
"SERVER: ",
data); if (data.success) { location.href =
`success.html?museum=${encodeURIComponent(museum)}&total=${encodeURIComponent(total)}&tickets=${encodeURIComponenter
} else { alert(
Знайдено посилань: 0% id: 187
"DB ERROR: "
+ data.error); }) } .catch(err => { console.log(err); alert(
Знайдено посилань: 0% id: 188
"Server error"
); }); } /script /body /html Додаток Д Код сторінки для отримання квитка !DOCTYPE html html
lang=
Знайдено посилань: 0% id: 189
"uk"
head meta charset=
Знайдено посилань: 0% id: 190
"UTF-8"
title Успіх /title script src=
Знайдено посилань: 0,01% id: 191
"http://dnr.cloudflare.com/ajax/libs/pdf/2.5.1/pdf.min.js"
/script link rel=
Знайдено посилань: 0% id: 192
"stylesheet"
href=
Знайдено посилань: 0% id: 193
"style.css"
/head body div class=
Знайдено посилань: 0% id: 194
"card big"
id=
Знайдено посилань: 0% id: 195
"ok"
/div script const p = new URLSearchParams(location.search); const museum = p.get(
Знайдено посилань: 0% id: 196
"museum"
```

<pre>); const total = p.get(</pre>	Знайдено посилань: 0%	id: 197
<pre>"total"</pre>		
<pre>); const tickets = p.get(</pre>	Знайдено посилань: 0%	id: 198
<pre>"ticket"</pre>		
<pre>); const time = p.get(</pre>	Знайдено посилань: 0%	id: 199
<pre>"time"</pre>		
<pre>); const date = p.get(</pre>	Знайдено посилань: 0%	id: 200
<pre>"date"</pre>		
<pre>); const group = p.get(</pre>	Знайдено посилань: 0%	id: 201
<pre>"group"</pre>		
<pre>); const firstName = p.get(</pre>	Знайдено посилань: 0%	id: 202
<pre>"firstName"</pre>		
<pre>); const lastName = p.get(</pre>	Знайдено посилань: 0%	id: 203
<pre>"lastName"</pre>		
<pre>); if (!museum) {document.body.innerHTML =</pre>	Знайдено посилань: 0%	id: 204
<pre>"Д н н зн йд н"</pre>		
<pre>}; else { document.getElementById(</pre>	Знайдено посилань: 0%	id: 205
<pre>"out"</pre>		
<pre>).innerHTML = `h1 Заброньовано /h1 p \${museum} /p p \${firstName} \${lastName} /p p \${time} /p p \${date} /p p Група N:\${group} /p p \${tickets} /p h3 \${total} z /h3 button class =</pre>	Знайдено посилань: 0%	id: 206
<pre>"btn"</pre>		
<pre>onclick =</pre>	Знайдено посилань: 0%	id: 207
<pre>"pdf()"</pre>		
<pre>Скачати PDF /button`; } function pdf() { const { jsPDF } = window.jspdf; const doc = new jsPDF(); doc.setFont(</pre>	Знайдено посилань: 0%	id: 208
<pre>"helvetica",</pre>	Знайдено посилань: 0%	id: 209
<pre>"normal"</pre>		
<pre>); const safeMuseum = museum; const ticketLines = tickets .split(</pre>	Знайдено посилань: 0%	id: 210
<pre>"",</pre>		
<pre>).map(t = { const line = t.trim(); return line .replace(/Дорслий/g,</pre>	Знайдено посилань: 0%	id: 211
<pre>"Adult"</pre>		
<pre>).replace(/Дитячий/g,</pre>	Знайдено посилань: 0%	id: 212
<pre>"Child"</pre>		
<pre>).replace(/Пльг_вий/g,</pre>	Знайдено посилань: 0%	id: 213
<pre>"Discount"</pre>		
<pre>).replace(/Дти\Молодь \((7-26 рок\в)/g,</pre>	Знайдено посилань: 0%	id: 214
<pre>"У адт (7-26)"</pre>		
<pre>).replace(/Дти до 7 рок\в/g,</pre>	Знайдено посилань: 0%	id: 215
<pre>"Children under 7"</pre>		
<pre>); }); doc.setFontSize(24); doc.text(</pre>	Знайдено посилань: 0%	id: 216
<pre>"MUSEUM TICKET",</pre>		
<pre>20, 25); doc.line(20, 32, 190, 32); doc.setFontSize(15); doc.text("Museum:", 20, 50); doc.text(safeMuseum, 70, 50); doc.text("Visitor:", 20, 65); doc.text(firstName + " " + lastName, 70, 65); doc.text(</pre>	Знайдено посилань: 0%	id: 217
<pre>"Date:",</pre>		
<pre>20, 80); doc.text(date, 70, 80); doc.text(</pre>	Знайдено посилань: 0%	id: 218
<pre>"Time:",</pre>		
<pre>20, 95); doc.text(time, 70, 95); doc.text(</pre>	Знайдено посилань: 0%	id: 219
<pre>"Group:",</pre>		
<pre>20, 110); doc.text(</pre>	Знайдено посилань: 0%	id: 220

"No "	
+ group, 70, 110); doc.text(	
Знайдено посилань: 0%	id: 221
"Tickets":,	
20, 125); let yTickets = 125; ticketLines.forEach(line = { doc.text(line, 70, yTickets); yTickets += 8; }); const y = yTickets + 10; doc.text("Total:", 20, y); doc.text(total + " zł	
Знайдено посилань: 0,01%	id: 222
", 70, y); doc.setFontSize(11); doc.text("Please arrive 15 minutes before the visit.	
Знайдено посилань: 0,01%	id: 223
", 20, y + 25); doc.rect(10, 10, 190, y + 35); doc.save("ticket.pdf	
Знайдено посилань: 0,02%	id: 224
"); } /script /body /html Д д т к Е К д ст р нки для приедн ння д б з и д н и x ? php \$ host =	
localhost	
Знайдено посилань: 0%	id: 225
"; \$user = "	
root	
Знайдено посилань: 0%	id: 226
", \$password = ""; \$database = "booking_system"; \$conn = new mysqli(\$host, \$user, \$password, \$database); if (\$conn-connect_error) { die("DB ERROR: " . \$conn-connect_error); } \$conn-set_charset(	
Знайдено посилань: 0%	id: 227
"utf8"	
); ? Додаток Є Код сторінки для зберігання даних в базі даних ? php error_reporting(E_ALL); ini_set('display_errors', 1); require	
Знайдено посилань: 0%	id: 228
"connect.php"	
; header(	
Знайдено посилань: 0%	id: 229
"Content-Type: application/json"	
); \$data = json_decode(file_get_contents(	
Знайдено посилань: 0%	id: 230
"php://input"	
), true); if (!\$data) { echo json_encode([	
Знайдено посилань: 0%	id: 231
"success"	
= false,	
Знайдено посилань: 0%	id: 232
"error"	
=	
Знайдено посилань: 0%	id: 233
"No data received"	
)); exit;} \$name = \$data[	
Знайдено посилань: 0%	id: 234
"name"]	
; \$surname = \$data[	
Знайдено посилань: 0,01%	id: 235
"surname"; \$date = \$data["date"]; \$time = \$data["time"]	
; \$groupId = \$data[	
Знайдено посилань: 0%	id: 236
"group"]	
; \$id_place = \$data[	
Знайдено посилань: 0%	id: 237
"id_place"]	
; \$tickets = \$data[	
Знайдено посилань: 0%	id: 238
"tickets"	
]; try { foreach (\$tickets as \$ticket) { \$ticketName = \$ticket[	
Знайдено посилань: 0%	id: 239
"name"	
]; \$ticketCount = \$ticket["count"]; \$stmt = \$conn-prepare(" SELECT id_ticket FROM ticket_type WHERE id_place = ? AND ticket_name = ? LIMIT 1	
Знайдено посилань: 0%	id: 240
"); \$stmt-bind_param("is	
Знайдено посилань: 0,02%	id: 241
", \$id_place, \$ticketName); \$stmt-execute(); \$res = \$stmt-get_result(); \$ticketData = \$res-fetch_assoc(); if (!\$ticketData) { continue; } \$id_ticket = \$ticketData["	
id_ticket	
Знайдено посилань: 0%	id: 242
"); \$stmt2 = \$conn-prepare("SELECT id_group FROM excursion_group WHERE id_place = ? ORDER BY id_group LIMIT 1 OFFSET ?	
Знайдено посилань: 0,01%	id: 243

```

"); $offset = $groupindex - 1; $stmt2- bind_param("
id: 244
Знайдено посилань: 0,02%
", $id_place, $offset); $stmt2- execute(); $res2 = $stmt2- get_result(); $groupData = $res2-
fetch_assoc(); if (!$groupData) { continue; } $id_group = $groupData["
id_group
Знайдено посилань: 0%
"]; $check = $conn- prepare("
SELECT SUM(ticket_count) as total FROM booking WHERE id_group = ?
id: 246
Знайдено посилань: 0%
"); $check- bind_param("
id: 247
Знайдено посилань: 0,01%
", $id_group); $check- execute(); $res = $check- get_result(); $row = $res- fetch_assoc();
$current = $row["
total
Знайдено посилань: 0,01%
"] ?? 0; $newTotal = $current + array_sum(array_column($tickets, "
count
Знайдено посилань: 0,01%
")); $groupTimes = [ 1 = "
10:00
Знайдено посилань: 0%
", 2 = "
13:00", 3 = "16:00" ]; $timeLabel = $groupTimes[$groupindex] ??
id: 251
Знайдено посилань: 0%
"unknown"
; if ($newTotal > 15) { echo json_encode([
id: 252
Знайдено посилань: 0%
"success"
= false,
id: 253
Знайдено посилань: 0%
"error"
=
id: 254
Знайдено посилань: 0,01%
"Н ц й ч с к л ь к с т ь к в и т к в щ в м п т р б н н д ступн ",
id: 255
Знайдено посилань: 0%
"details"
= [
id: 256
Знайдено посилань: 0%
"group_time"
= $timeLabel,
id: 257
Знайдено посилань: 0%
"current_tickets"
= $current,
id: 258
Знайдено посилань: 0%
"requested_tickets"
= array_sum(array_column($tickets,
id: 259
Знайдено посилань: 0%
"count"
)), "max" = 15 ] ); exit; } $stmt3 = $conn- prepare(" INSERT INTO booking ( customer_name,
customer_surname, id_group, id_ticket, ticket_count, visit_time, booking_date ) VALUES (?, ?, ?, ?,
?, ?, ?)
id: 260
Знайдено посилань: 0%
"); $stmt3- bind_param( "
sssiiss", $name, $surname, $id_group, $id_ticket, $ticketCount, $time, $date ); if (!$stmt3-
execute()) { echo json_encode([ "success" = false,
id: 261
Знайдено посилань: 0%
"error"
= $stmt3- error ]; exit; } } echo json_encode([
id: 262
Знайдено посилань: 0%
"success"
= true ]; ); catch (Exception $e) { echo json_encode([
id: 263
Знайдено посилань: 0%
"success"
= false,
id: 264
Знайдено посилань: 0%
"error"
= $e- getMessage() ); } Додаток Ж Код сторінки стилів сайту body { margin: 0; font-family:
Arial, sans-serif; background: linear-gradient(135deg, #007bff, #0006ff); color: white; } .topbar {
display: flex; justify-content: space-between; align-items: center; padding: 15px 20px;
background: #1f1f1f; color: white; } .back { color: white; text-decoration: none; font-size: 14px;
opacity: 0.9; } .back:hover { opacity: 1; } .hero { text-align: center; padding: 60px 20px 30px; }
.hero h1 { font-size: 42px; margin-bottom: 10px; } .hero p { font-size: 18px; opacity: 0.9; } .menu
{ display: flex; justify-content: center; gap: 20px; padding: 40px; flex-wrap: wrap; } .menu_card {
background: white; color: black; width: 260px; padding: 25px; border-radius: 16px; text-align:
center; text-decoration: none; transition: 0.3s; box-shadow: 0 10px 25px rgba(0,0,0,0.2); } .menu

```

<pre>.card:hover { transform: translateY(-10px); } .icon { font-size: 40px; margin-bottom: 10px; } .disabled { opacity: 0.6; cursor: not-allowed; } .container { max-width: 1200px; margin: 40px auto; padding: 0 20px; } .grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(260px, 1fr)); gap: 20px; } .card { background: white; color: black; padding: 18px; border-radius: 14px; box-shadow: 0 8px 20px rgba(0,0,0,0.12); transition: 0.25s; } .card:hover { transform: translateY(-6px); } .card h3 { margin-top: 0; } .card p { color: #555; font-size: 14px; } /* big card (museum page) */ .card.big { max-width: 500px; margin: 40px auto; text-align: center; } .btn { display: inline-block; margin-top: 10px; padding: 10px 14px; background: #007bff; color: white; border-radius: 8px; text-decoration: none; transition: 0.2s; } .btn:hover { background: #0056b3; } .ticket { display: flex; justify-content: space-between; align-items: center; margin: 10px 0; background: #f5f5f5; padding: 10px; border-radius: 8px; color: black; } .ticket input { width: 60px; } .input { padding: 6px; border-radius: 6px; border: 1px solid #ccc; } .footer { text-align: center; padding: 20px; color: white; opacity: 0.8; margin-top: 40px; } h1, h2, h3 { margin: 10px 0; } .museum-img { width: 100%; height: 300px; object-fit: cover; border-radius: 15px; margin-bottom: 20px; } .full-text { margin-top: 15px; line-height: 1.7; color: #333; font-size: 18px; white-space: pre-line; } Додаток 3 Файл реалізації JavaScript const museums = {</pre>	id: 265
"1":	
{ name:	
Знайдено посилань: 0%	id: 266
"Muzeum Narodowe w Poznaniu",	
desc:	
Знайдено посилань: 0,01%	id: 267
"Г л вний иуд жн й муз й м ст а П зн нь",	
fullDesc: Muzeum Narodowe w Poznaniu є одним із найстарших та найважливіших музеїв Польщі. У музеї знаходяться: • польський живопис; • європейське мистецтво; • античні експонати; • сучасні виставки. Музей розташований у центрі Познані та щороку приймає тисячі туристів.`, image:	
Знайдено посилань: 0,01%	id: 268
"Muzeum Narodowe w Poznaniu.jpg",	
tickets:[{name:	
Знайдено посилань: 0%	id: 269
"Д р слий",	
price:20},{name:	
Знайдено посилань: 0%	id: 270
"П льг вий",	
price:13},{name:	
Знайдено посилань: 0%	id: 271
"Д ти/М л дь (7–26 р к в)",	
price:1}}},	
Знайдено посилань: 0%	id: 272
"2":	
{ name:	
Знайдено посилань: 0%	id: 273
"Muzeum Sztuk Użytkowych",	
desc:	
Знайдено посилань: 0%	id: 274
"Муз й д к р тивного мистецтва",	
fullDesc: Muzeum Sztuk Użytkowych присвячений декоративному мистецтву та дизайну. У музеї представлені: • старовинні меблі; • кераміка; • годинники; • предмети інтер'єру; • вироби XVI–XIX століття. Музей знаходиться у Королівському замку в Познані та має оглядову вежу.`, image:	
Знайдено посилань: 0%	id: 275
"Muzeum Sztuk Użytkowych.jpg",	
tickets:[{name:	
Знайдено посилань: 0%	id: 276
"Д р слий",	
price:20},{name:	
Знайдено посилань: 0%	id: 277
"П льг вий",	
price:13},{name:	
Знайдено посилань: 0%	id: 278
"Д ти/М л дь (7–26 р к в)",	
price:1}}},	
Знайдено посилань: 0%	id: 279
"3":	
{ name:	
Знайдено посилань: 0%	id: 280
"Muzeum Uzbrojenia",	
desc:	
Знайдено посилань: 0,01%	id: 281
"Муз й в йськ в і т н ки та стор і",	
fullDesc: Muzeum Uzbrojenia демонструє військову історію Польщі. У музеї можна побачити: • танки; • гармати; • бронетехніку; • військову форму; • історичну зброю. Музей особливо популярний серед любителів військової техніки.`, image:	
Знайдено посилань: 0%	id: 282
"Muzeum Uzbrojenia.jpg",	
tickets:[{name:	
Знайдено посилань: 0%	id: 283

"Д р слий",	
price:15},{name:	
 Знайдено посилань: 0%	id: 284
"П льг вий",	
price:10},{name:	
 Знайдено посилань: 0%	id: 285
"Д ти д 7 р к в",	
price:0}}},	
 Знайдено посилань: 0%	id: 286
"4":	
{ name:	
 Знайдено посилань: 0%	id: 287
"Muzeum Archeologiczne",	
desc:	
 Знайдено посилань: 0%	id: 288
"Муз й рт ф кт в ст р д в н к цив л з ц й",	
fullDesc: ` Muzeum Archeologiczne представляє археологічні знахідки різних епох. Експозиція містить: • стародавні арт факти; • єгипетські експонати; • монети; • прикраси; • предмети побуту. Музей дозволяє відвідувачам дізнатися більше про розвиток цивілізації.`, image:	
 Знайдено посилань: 0%	id: 289
"Muzeum Archeologiczne.jpg",	
tickets:[{name:	
 Знайдено посилань: 0%	id: 290
"Д р слий",	
price:10},{name:	
 Знайдено посилань: 0%	id: 291
"П льг вий",	
price:6}}},	
 Знайдено посилань: 0%	id: 292
"5":	
{ name:	
 Знайдено посилань: 0%	id: 293
"Muzeum Instrumentów Muzycznych",	
desc:	
 Знайдено посилань: 0%	id: 294
"Муз й музичні нструм нт в св ту",	
fullDesc: ` Muzeum Instrumentów Muzycznych є унікальним музеєм музики. У колекції знаходяться: • піаніно; • скрипки; • духові інструменти; • народні інструменти; • старовинні музичні експонати. Відвідувачі можуть дізнатися про історію музики різних народів світу.`, image:	
 Знайдено посилань: 0%	id: 295
"Muzeum Instrumentów Muzycznych.pdf",	
tickets:[{name:	
 Знайдено посилань: 0%	id: 296
"Д р слий",	
price:15},{name:	
 Знайдено посилань: 0%	id: 297
"П льг вий",	
price:10}}},	
 Знайдено посилань: 0%	id: 298
"6":	
{ name:	
 Знайдено посилань: 0%	id: 299
"Centrum Szyfrów Enigma",	
desc:	
 Знайдено посилань: 0%	id: 300
"Муз й крипт гр ф і т шифр в",	
fullDesc: ` Centrum Szyfrów Enigma присвячений історії криптографії та шифрувальних машин. У центрі можна побачити: • копії машини Enigma; • інтерактивні стенди; • історичні документи; • матеріали про криптографів. Музей розповідає про внесок польських математиків у розшифрування Enigma.`, image:	
 Знайдено посилань: 0%	id: 301
"Centrum Szyfrów Enigma.jpg",	
tickets:[{name:	
 Знайдено посилань: 0%	id: 302
"Д р слий",	
price:25}}},	
 Знайдено посилань: 0%	id: 303
"7":	
{ name:	
 Знайдено посилань: 0%	id: 304
"Rogalowe Muzeum",	
desc:	
 Знайдено посилань: 0,01%	id: 305
"нт р ктивний муз й п зн нські р г л и к в Свят г М р ц н",	
fullDesc: ` Rogalowe Muzeum Poznania — один із найвідоміших туристичних музеїв міста.	

Музей присвячений традиційному роґалику Святого Марціна, який є символом Познані. Під час інтерактивного шоу відвідувачі можуть: • дізнатися історію роґалика; • побачити процес приготування; • взяти участь у майстер-класі; • скуштувати свіжу випічку; • почути легенди та познанський діалект. Музей поєднує гумор, кулінарні традиції та живі вистави, тому особливо популярний серед туристів і сімей.`, image:	
 Знайдено посилань: 0%	id: 306
"Rogalows Museum.jpg",	
tickets:[{name:	
 Знайдено посилань: 0%	id: 307
"Д р слий",	
price:43},{name:	
 Знайдено посилань: 0%	id: 308
"Д ти/М л ьд (7-26 р к в)",	
price:37}}},	
 Знайдено посилань: 0%	id: 309
"8":	
{ name:	
 Знайдено посилань: 0%	id: 310
"Muzeum Broni Pancernej",	
desc:	
 Знайдено посилань: 0%	id: 311
"Музей танк в та брон танки",	
fullDesc: ` Muzeum Broni Pancernej спеціалізується на бронетехніці. У музеї представлені: • танки; • бронемашини; • військові автомобілі; • історична техніка XX століття. Багато експонатів є справжніми зразками військової історії.`, image:	
 Знайдено посилань: 0%	id: 312
"Muzeum Broni Pancernej.jpg",	
tickets:[{name:	
 Знайдено посилань: 0%	id: 313
"Д р слий",	
price:32},{name:	
 Знайдено посилань: 0%	id: 314
"П льг вий",	
price:21}}},	
 Знайдено посилань: 0%	id: 315
"9":	
{ name:	
 Знайдено посилань: 0%	id: 316
"Muzeum Pu y",	
desc:	
 Знайдено посилань: 0%	id: 317
"Музей картпл",	
fullDesc: ` Muzeum Pu y присвячений історії та культурі картоплі у Польщі. У музеї можна: • дізнатися історію картоплі; • побачити тематичні експозиції; • взяти участь у кулінарних шоу; • скуштувати традиційні страви. Музей має інтерактивний формат та підходить для всієї родини.`, image:	
 Знайдено посилань: 0%	id: 318
"Muzeum Pu y.jpg",	
tickets:[{name:	
 Знайдено посилань: 0%	id: 319
"Д р слий",	
price:25}}}};	

Відмова від відповідальності:

Цей звіт повинен бути правильно інтерпретований та проаналізований кваліфікованою особою, яка несе відповідальність за оцінку!

Будь-яка інформація, наведена у цьому звіті, не є остаточною та підлягає ручному огляду та аналізу. Дотримуйтесь вказівок Рекомендації з оцінки