

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук і технологій
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Аліна САВЧЕНКО

“ _____ ” _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»
ЗА СПЕЦІАЛЬНІСТЮ 122 «КОМП'ЮТЕРНІ НАУКИ»

Тема: «Інформаційно-пошукова система надання архітектурних послуг»

Виконавець: Погорелов Дмитро Олегович

Керівник: доцент Харченко Олександр Григорович

Нормоконтролер: Олександр ШЕВЧЕНКО
(підпис)

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук і технологій

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь: «Бакалавр»

Галузь знань, спеціальність, освітньо–професійна програма:

12 «Інформаційні технології», 122 «Комп'ютерні науки», «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Аліна САВЧЕНКО

“ _____ ” _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Погорелова Дмитра Олеговича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

- 1. Тема кваліфікаційної роботи:** «Інформаційно-пошукова система надання архітектурних послуг» затверджена наказом ректора №623/ст від 01.05.2023 р.
- 2. Термін виконання роботи:** 15.05.2023 – 25.06.2023
- 3. Вихідні дані до роботи:** дані про веб–сервіс, документація, база даних MySQL, пакет phpMyAdmin.
- 4. Зміст пояснювальної записки:** аналіз існуючих веб-хостингів, системний аналіз та розробка загальної схеми веб–сайту з інформаційною системою, пояснення розробки інформаційної системи, представлення роботи веб–сервісу.
- 5. Перелік обов'язкового графічного (ілюстративного) матеріалу:** слайди презентації Microsoft Power Point.

6. Календарний план–графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Аналіз та дослідження предметної області. Огляд та аналіз доступних технологій та фреймворків.	15.05.2023– 19.05.2023	Виконано
2	Розробка деталізованого змісту розділів кваліфікаційної роботи. Написання вступу.	22.05.2023– 23.05.2023	Виконано
3	Розробка структури інформаційної системи та веб-сервісу. Написання 1 розділу кваліфікаційної роботи.	24.05.2023– 26.05.2023	Виконано
4	Створення веб–сервісу з використанням досліджених технологій. Написання 2 розділу кваліфікаційної роботи.	29.05.2023– 02.06.2023	Виконано
5	Написання 3 розділу кваліфікаційної роботи. Усунення недоліків та завершення створення пояснювальної записки кваліфікаційної роботи.	05.06.2023– 09.06.2023	Виконано
6	Оформлення та друк пояснювальної записки кваліфікаційної роботи.	12.06.2023– 13.06.2023	Виконано
7	Підготовка презентації та доповіді для виступу.	14.06.2023– 16.06.2023	Виконано
8	Захист кваліфікаційної роботи.	19.06.2023– 23.06.2023	Виконано

7. Дата видачі завдання: 15.05.2023 р.

Керівник кваліфікаційної роботи

_____ (підпис керівника)

Олександр ХАРЧЕНКО

(П.І.Б.)

Завдання прийняв до виконання

_____ (підпис випускника)

Дмитро ПОГОРСЛОВ

(П.І.Б.)

РЕФЕРАТ

Кваліфікаційна робота «Інформаційно-пошукова система надання архітектурних послуг» містить 75 сторінок, 33 рисунки, 8 використаних джерел.

Об'єкт дослідження: Інформаційно-пошукова система.

Предмет дослідження: розробка веб-сайту з інформаційно-пошуковою системою.

Мета кваліфікаційної роботи: розробка веб-сервісу для надання архітектурних послуг з використанням різних технологій, визначення вимог, огляд доступних технологій.

Метод дослідження: аналіз та розкриття поняття інформаційної системи, аналіз сучасних Web-технологій.

Результат проекту: розроблений веб-сервіс, що надає функціонал у сфері надання архітектурних послуг.

ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА, CONSTRUCT 2, PHP, HTML, JS.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ВИЗНАЧЕННЯ ЦІЛЕЙ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА ФОРМУЛЮВАННЯ ВИМОГ ДО ФУНКЦІОНАЛЬНОСТІ, ДИЗАЙНУ ТА ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ.	8
1.1. Критерії вибору хостингу для веб-сайту	11
1.1.1. Порівняльна таблиця хостингів	13
1.1.2. Вибір хостингу	13
1.2. Характеристика використаних програмних засобів.....	14
1.2.1. Google Chrome	14
1.2.2. Construct 2	16
1.3. Список необхідних функцій інформаційної системи, інформація про технології, які можна використати для реалізації цих функцій.	20
1.4. Висновки до розділу 1	27
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА РОЗРОБКА ЗАГАЛЬНОЇ СХЕМИ ВЕБ-САЙТУ	28
2.1. Основні завдання СУБД	28
2.2. Порівняльна таблиця різних систем управління базами даних.	32
2.3. Розробка логічної моделі бази даних.....	34
2.4. phpMyAdmin.....	35
2.5. Висновки до розділу 2	36
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-ПОШУКОВОЇ СИСТЕМИ, НАВІГАЦІЙНОГО МЕНЮ ТА ДИЗАЙНУ ВЕБ-САЙТУ	38
3.1. Вибір дизайну веб-сайту.....	38
3.2. Створення навігаційного меню	40
3.3. Додання інформації фірми на сайт.....	44
3.4. Створення бази даних та таблиці в phpMyAdmin	46
3.4.1. Створення бази даних	46

3.4.2. Створення таблиці в phpMyAdmin	47
3.6. Висновок до розділу 3.....	54
ВИСНОВКИ	56
СПИСОК БІБЛОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	58
ДОДАТКИ	59

ВСТУП

В наш час більшість людей шукають інформацію в Інтернеті, тому використання веб-сайтів з інформаційною системою для просування послуг компанії може бути ефективним інструментом для залучення потенційних клієнтів і збільшення впізнаваності бренду. Важливо продумати стратегію і підхід, щоб досягти максимальних результатів.

В даній роботі буде створено веб-сайт з інформаційною системою для надання архітектурних послуг фірми ТОВ «ГУЛСТА».

Сайт надає компанії можливість бути присутньою онлайн і доступною для потенційних клієнтів 24/7. Це дає змогу залучати нових клієнтів і встановлювати контакт із ними навіть поза робочими годинами.

Інформаційна база: сайт надає можливість детально та повно описати послуги компанії, їхні переваги, ціни та іншу важливу інформацію. Потенційні клієнти можуть ознайомитися з цією інформацією у зручний для них час і прийняти інформоване рішення про співпрацю.

Широке охоплення аудиторії: сайт може бути доступний для користувачів з усього світу. Це дає змогу компанії залучати клієнтів із різних регіонів і країн. Можливість міжнародного просування робить сайт ефективним інструментом для розширення ринку і збільшення клієнтської бази.

Візуальне джерело інформації: сайт дає змогу компанії представити свої послуги в привабливому та професійному вигляді. За допомогою якісного дизайну, фотографій, відео та інших медіа-елементів можна викликати довіру у відвідувачів і створити позитивне враження про компанію та її послуги.

РОЗДІЛ 1

**ВИЗНАЧЕННЯ ЦІЛЕЙ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА
ФОРМУЛЮВАННЯ ВИМОГ ДО ФУНКЦІОНАЛЬНОСТІ, ДИЗАЙНУ
ТА ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ.**

Сайт з інформаційною системою можливо ефективно використовувати для просування послуг компанії. Ось деякі способи використання сайту для цієї мети:

Створення привабливого дизайну: Розробити візуально привабливий і професійний дизайн сайту, який відповідає бренду компанії і зручний для користувачів. Чистий і сучасний дизайн допоможе створити позитивне враження у відвідувачів і підкреслить професіоналізм компанії.

Інформація про послуги: Надати детальну інформацію про послуги, які пропонує компанія. Описати їхні особливості, переваги та ціни. Постаратися зробити інформацію зрозумілою, доступною і такою, що легко сприймається відвідувачами.

Контент-маркетинг: Створити корисний та інформативний контент, пов'язаний з послугами компанії. Наприклад, можливо публікувати статті, блоги, керівництва або відео, які пропонують цінні поради та вирішення проблем для вашої цільової аудиторії. Це допоможе поставити компанію як експерта у своїй галузі та залучити більше відвідувачів на ваш сайт.

Онлайн-форми та запити: Створити форми на сайті, які дозволять відвідувачам надсилати запити або замовляти послуги. Потрібно переконатися, що форми прості у використанні та інтуїтивно зрозумілі. Швидка і зручна взаємодія з клієнтами може підвищити кількість переходів і

Кафедра КІТ (47)				НАУ 23.17.60 000 ПЗ			
Виконав	Погорелов Д.О.			ВИЗНАЧЕННЯ ЦІЛЕЙ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА ФОРМУЛЮВАННЯ ВИМОГ ДО ФУНКЦІОНАЛЬНОСТІ, ДИЗАЙНУ ТА ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ.	Літера	Аркуш	Аркушів
Керівник	Харченко О.Г.					8	20
Консультант					<i>УС-411 122</i>		
Н.Контроль	Шевченко О..						

збільшити число запитів на послуги.

Також можливо застосувати основи SEO (оптимізацію для пошукових систем) на сайті, щоб покращити його видимість у пошукових результатах. Використовувати ключові слова, оптимізувати теги, заголовки та описи сторінок, створювати якісний контент і забезпечити хорошу швидкість завантаження сторінок. Це допоможе залучити більше органічного трафіку і підвищити відвідуваність вашого сайту.

Масштабованість і гнучкість: На відміну від традиційних методів просування, сайт з інформаційною системою може бути легко масштабованим і гнучким. Ви можете додавати нові сторінки, функції та контент відповідно до розвитку вашої компанії та змін потреб ринку. Це дає змогу тримати ваше просування в актуальному стані й адаптуватися до змін у бізнес-середовищі.

Веб-сайт - це колекція веб-сторінок, пов'язаних між собою гіперпосиланнями і доступних через інтернет. Вони можуть містити різну інформацію, таку як текст, зображення, відео, аудіо, а також форми для введення та надсилання даних. Веб-сайти можуть бути статичними або динамічними, залежно від того, чи містять вони контент, що динамічно генерується, наприклад, з бази даних. Веб-сайти можуть бути частиною інформаційної системи, являючи собою інтерфейс для користувачів, щоб отримувати доступ до інформації та взаємодіяти з нею.

Інформаційна система (ІС) - це комплексне апаратне забезпечення, програмне забезпечення, дані, процедури та особи, що працюють разом для збирання, оброблення, зберігання та розповсюдження інформації. Кожен компонент інформаційної системи виконує свою унікальну функцію, щоб підтримувати цілісність і ефективність усієї системи.

Складові інформаційної системи можуть бути розділені на різні категорії, що описані нижче.

Апаратні засоби: комп'ютери, сервери, мобільні пристрої, периферійні пристрої та мережеве обладнання. Апаратні засоби забезпечують фізичну інфраструктуру, яка підтримує роботу ІС;

Програмне забезпечення: операційні системи, бази даних, додатки, програми та скрипти. Програмне забезпечення забезпечує функціональність, необхідну для обробки інформації, і взаємодіє з іншими компонентами ІС;

Дані: інформація, яку ІС обробляє, зберігає і передає. Дані можуть бути структурованими або неструктурованими і можуть містити текст, зображення, аудіо та відео;

Процедури: правила і процеси, які використовуються для збирання, оброблення та передавання інформації. Процедури можуть бути автоматизованими або виконуватися вручну, і вони забезпечують правильну роботу ІС;

Люди: користувачі, розробники, системні адміністратори, менеджери та інші люди, які використовують і керують ІС. Люди відіграють важливу роль у забезпеченні правильної роботи ІС і забезпеченні безпеки інформації.

Усі ці компоненти взаємодіють один з одним, щоб забезпечити роботу ІС. Кожен компонент необхідний для правильного функціонування ІС, і будь-яка невідповідність може призвести до порушення роботи всієї системи.

Веб-сайт, що буде розроблений в результаті виконання цієї роботи, є частиною інформаційної системи, яка містить у собі базу даних для зберігання інформації про користувачів і замовлення. Веб-сайт може обробляти замовлення, взаємодіючи з іншими компонентами інформаційної системи, такими як системи управління, щоб забезпечити швидке і точне виконання замовлень.

Тобто, веб-сайт є важливою частиною інформаційної системи, надаючи користувачеві доступ до інформації та можливість взаємодії з іншими компонентами системи.

1.1. Критерії вибору хостингу для веб-сайту

Вибір хорошого хостингу для сайту є важливим кроком, який може суттєво впливати на його продуктивність, доступність і безпеку. Ось кілька факторів, які потрібно враховувати під час вибору хостинг-провайдера:

Один із ключових аспектів хорошого хостингу - це надійність і доступність серверів. Треба переконатися, що хостинг-провайдер має високу відсоткову гарантію доступності (зазвичай вище 99%). Потрібно дослідити репутацію провайдера, ознайомитися з відгуками користувачів і дізнатися про його історію непланових простоїв.

Швидке завантаження сайту є важливим для задоволення користувачів і поліпшення ранжирування в пошукових системах. Треба дізнатися, які технічні характеристики пропонує хостинг-провайдер, як-от обсяг виділеної пам'яті (RAM), процесори, типи дискового простору і наявність кешування. Також, уточнити, чи підтримує хостинг SSD-диски, які забезпечують більш швидку роботу сайту.

У випадку, якщо сайт почне рости і залучати більше відвідувачів. Потрібно переконатися, що хостинг-провайдер пропонує можливості масштабування, такі як можливість переходу на потужніші плани хостингу, збільшення ресурсів або додаткові опції розширення.

Хороша технічна підтримка від хостинг-провайдера може бути важлива, особливо якщо виникнуть проблеми або буде потрібна допомога. Треба дізнатись, який рівень підтримки пропонується (наприклад, 24/7 онлайн-підтримка телефоном, електронною поштою або чатом), як швидко вони відповідають на запити і які канали комунікації доступні.

Безпека: Забезпечення безпеки вебсайту - це важливий аспект хостинг-провайдера. Необхідно дізнатись, яких заходів безпеки вживають, наприклад, регулярні резервні копії даних, захист від DDoS-атак, міжсайтовий захист (XSS), міжсайтове підроблення запиту (CSRF) тощо.

Інтерфейс управління: хороший хостинг-провайдер надає зручний та

інтуїтивно зрозумілий інтерфейс управління, який дасть змогу легко керувати сайтом і його налаштуваннями. Потрібно звернути увагу на те, яка панель управління використовується і які функції доступні.

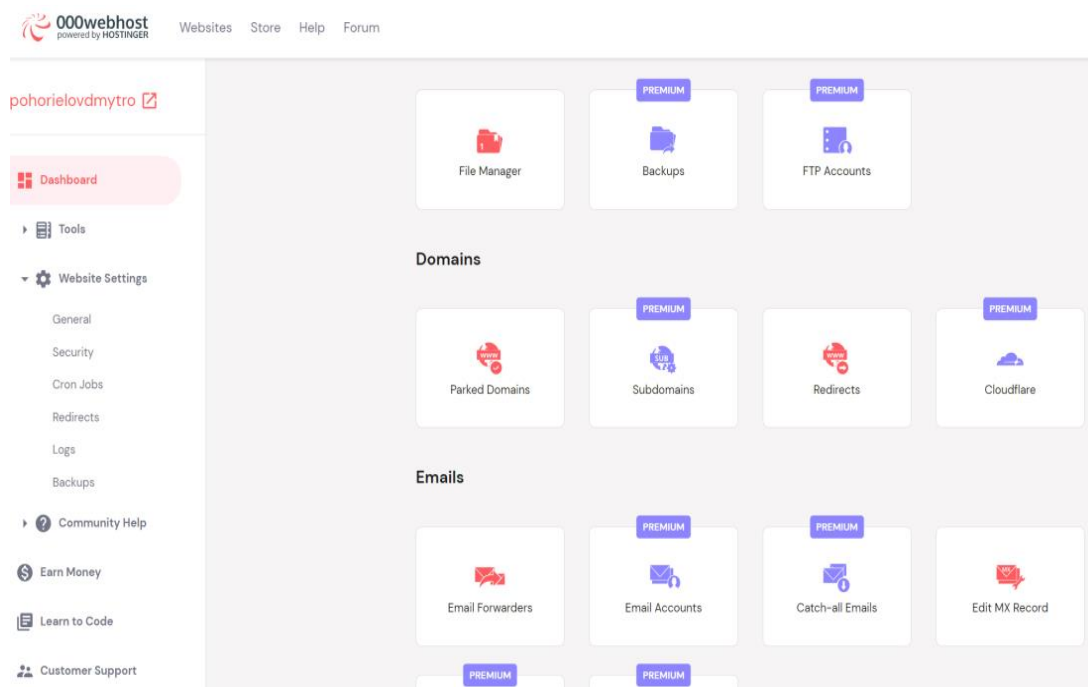


Рис. 1.1. Приклад інтерфейсу управління хостингу 000webhost

Вартість: Вартість хостингу - один із чинників, які потрібно врахувати. Треба порівняти різні плани хостингу та їхню вартість, врахувати не тільки ціну, а й включені функції, ресурси та можливості масштабування. Також звернути увагу на можливість надання знижок або акцій від провайдера.

Відгуки та рекомендації: Необхідно дослідити відгуки та рекомендації від інших користувачів або експертів. Це може допомогти отримати уявлення про якість і надійність хостинг-провайдера.

Тип хостингу: Існує кілька типів хостингу, таких як загальний хостинг, віртуальний приватний сервер (VPS), виділений сервер, хмарний хостинг та інші. Потрібно обрати тип хостингу, який відповідає потребам сайту і бюджету.

Розташування серверів: Якщо аудиторія зосереджена в певному регіоні, треба врахувати розташування серверів хостинг-провайдера.

Близькість серверів до цільової аудиторії може поліпшити швидкість завантаження сайту.



Рис. 1.2. Приклад часу відповіді від серверів в різних місцях планети

Взагалі, вибір хостинг-провайдера залежить від конкретних потреб і бюджету. Тому потрібне дослідження, щоб порівняти різні варіанти та обрати того, хто найкраще відповідає вимогам.

1.1.1. Порівняльна таблиця хостингів

Для порівняння я обрав 3 раніше використовуваних мною веб-хостинги Beget.com, 000webhost.com та zzz.com.ua

Таблиця 1.1

Порівняльна веб-хостингів

Характеристика	Beget	000webhost	zzz.com.ua
Ціна	Платний	Платний / безкоштовний	Платний / безкоштовний
Доступ	гарантовано	99.9% гарантовано	99.9% гарантовано
Місце на диску	15GB	200GB	13GB
Трафік	Різні плани	Різні плани	Різні плани
Домени	Різна кількість	Різна кількість	Різна кількість
Підтримка	Онлайн	Онлайн	Форум
SSL сертифікат	Безкоштовний	Безкоштовний при плані	Платний

1.1.2. Вибір хостингу

Узгодивши ціну веб-хостингу з фірмою, було вибрано 000webhost. Основною перевагою 000webhost є його легкість використання. 000webhost надає простий та інтуїтивно зрозумілий користувацький інтерфейс.

Також можливо розмістити свій сайт без необхідності платити за хостинг. Це особливо корисно для тестування та експериментів, тому що при виникненні проблем можливо змінити хостинг.

000webhost обіцяє хорошу доступність серверів і достатню швидкість завантаження. Також 000webhost надає деякі корисні функції, наприклад, доступ до баз даних MySQL, підтримка PHP, FTP-доступ, безкоштовні SSL-сертифікати Let's Encrypt та інші. Провайдер прагне забезпечити надійність і хорошу продуктивність для користувачів.

1.2. Характеристика використаних програмних засобів

Використані мною програмні засоби у даній роботі це Construct 2, та браузер Google Chrome. Construct 2 - це середовище розробки, яке дає змогу створювати проекти без написання коду. Воно використовує візуальний скриптинг, де ви можете перетягувати і з'єднувати готові блоки для створення різних механік і логіки.

1.2.1. Google Chrome



Рис. 1.3. Логотип браузера Google Chrome

Google Chrome є одним із найпопулярніших веб-браузерів, і він пропонує низку потужних інструментів, які можуть бути корисними під час налагодження та створення сайту. Ось деякі з них:

Інструменти розробника: Google Chrome містить вбудовані інструменти розробника, які дозволяють аналізувати та налагоджувати веб-сторінки. Ви можете відкрити ці інструменти, натиснувши правою кнопкою миші на будь-якому місці сторінки і вибравши "Дослідити елемент" або "Переглянути код сторінки". Інструменти розробника пропонують безліч можливостей, таких як зміна вихідного коду, налагодження JavaScript, аналіз мережевого трафіку та багато іншого.

Режим мобільних пристроїв: За допомогою інструментів розробника в Chrome можна емулювати різні розміри екранів мобільних пристроїв. Це дає змогу побачити, який вигляд має сайт і як він працює на різних пристроях, що важливо для створення адаптивного дизайну. Вибравши режим мобільних пристроїв, можливо також емулювати медіа-запити, щоб перевірити, як сайт реагує на різні роздільні здатності екрана.

Інструменти для аналізу продуктивності: Chrome пропонує інструменти для аналізу продуктивності сайту. За допомогою вкладки "Аудит" в інструментах розробника можливо виконати перевірку продуктивності та отримати рекомендації щодо оптимізації вашого сайту. Це включає аналіз швидкості завантаження сторінки, оптимізацію ресурсів та інші фактори, які можуть вплинути на користувацький досвід.

Налагодження JavaScript: Інструменти розробника Chrome пропонують потужні можливості для налагодження JavaScript. Ви можете встановити точки зупинки у вихідному коді, відстежувати значення змінних, виконувати кроки виконання коду та багато іншого. Це дає змогу вам ідентифікувати та виправляти помилки у вашому JavaScript-коді та покращувати його продуктивність.

1.2.2. Construct 2

Незважаючи на те, що Construct 2 не був оригінально створений для цієї мети, його можна використовувати для створення веб-сайтів. Наприклад, можливо використовувати Construct 2 для створення анімацій, інтерактивних графіків, тестів або форм для заповнення.



Рис. 1.4. Логотипи програми Construct 2

Для використання Construct 2 на веб-сайті треба вбудувати готові проекти або створені за допомогою нього елементи у веб-сторінку за допомогою `iframe` або скриптів. Також можливо експортувати готові проекти в HTML5, що дає змогу запускати їх у веб-браузері без додаткових плагінів або розширень.

Construct 2 має низку переваг, які роблять його корисним інструментом для створення інтерактивних елементів на веб-сайті:

Він має простий інтерфейс, який легко освоїти навіть без знання програмування.

За допомогою Construct 2 ви можете швидко створювати сторінки та інтерактивні елементи, що робить його корисним інструментом для швидкої розробки прототипів.

Безліч готових елементів: Construct 2 постачається з бібліотекою готових елементів і графіки, які можна використовувати для створення проектів та інтерактивних елементів.

Експорт у HTML5 дає змогу запускати проекти та інтерактивні елементи у веб-браузері без необхідності встановлювати додаткові плагіни або розширення.

Візуальний скриптинг - це метод створення програмних інструкцій за допомогою графічного інтерфейсу замість написання коду мовою програмування. Нижче я перелічу деякі переваги та недоліки візуального скриптингу.

4	Mouse	Cursor is over SettingsIcon	SettingsIcon	Set Rotate speed to 45 degrees per second
5	Mouse	Cursor is over SettingsIcon	SettingsIcon	Set Rotate speed to 0 degrees per second
6	Mouse	On Left button Clicked on SettingsIcon	System	Wait 0.05 seconds
	System	settingsOpened = 1	System	Set settingsOpened to 0
7	Mouse	On Left button Clicked on SettingsIcon	System	Set settingsOpened to 1
8	System	settingsOpened = 0	SettingsBackground	Set Invisible
			DisableEffectsButton	Set Invisible
			FullscreenButton	Set Invisible
			DndIcon	Set Invisible
9	System	settingsOpened = 1	SettingsBackground	Set Visible
			DisableEffectsButton	Set Visible
			FullscreenButton	Set Visible
			DndIcon	Set Visible
0	FullscreenButton	On clicked	Browser	Request fullscreen - Stretch (scale inner)

Рис. 1.5. Приклад меню візуального скриптингу у Construct 2

Переваги візуального скриптингу:

1) Зручність використання;

Візуальний скриптинг може бути простішим для розуміння і використання для тих, хто не має досвіду в програмуванні. Це може бути особливо корисно для початківців, які тільки починають вивчати програмування.

2) Прискорення розробки;

Візуальний скриптинг може допомогти прискорити процес розробки, оскільки не вимагає написання коду вручну. Це може дозволити

розробникам швидше створювати прототипи, перевіряти концепції та доопрацьовувати проекти.

3) Візуалізація процесу;

Візуальний скриптинг дає змогу візуалізувати процес виконання програми, що може допомогти розробникам розуміти, як працює код і які зміни впливають на його поведінку.

4) Надійність.

Візуальний скриптинг може бути надійнішим, оскільки не допускає синтаксичних помилок, як-от друкарських помилок, які можуть виникнути під час написання коду вручну.

Недоліки візуального скриптингу:

1) Обмеження;

Візуальний скриптинг може бути обмежений у тому, що ви можете зробити з вашим кодом. Він може бути не настільки гнучким і потужним, як написання коду вручну, особливо для складніших завдань.

2) Низька продуктивність:

Візуальний скриптинг може бути менш ефективним, ніж написання коду вручну, особливо для великих проектів або завдань, які вимагають високої продуктивності.

3) Обмежена підтримка:

Деякі мови візуального скриптингу можуть мати обмежену підтримку спільноти, що може ускладнити знаходження рішень для проблем або помилок.

4) Складність налагодження:

Налагодження візуального скриптингу може бути складнішим, ніж написання коду вручну, оскільки, можливо, складно визначити, які саме блоки або вузли спричиняють помилку у вашій програмі, особливо якщо ви працюєте з великою кількістю блоків.

5) Нестача гнучкості:

Іноді візуальний скриптинг може бути менш гнучким, ніж написання коду вручну, оскільки деякі функції та можливості можуть бути недоступні або потребують додаткової роботи для їхньої реалізації.

б) Складність масштабування:

Візуальний скриптинг може зіткнутися з проблемами масштабування під час роботи з великими проектами, які потребують великої кількості блоків і вузлів. У цьому випадку може знадобитися перемикання на більш традиційний підхід програмування.

Загалом, візуальний скриптинг може бути корисним інструментом для розробки програмного забезпечення, особливо для розробників-початківців або для створення прототипів. Однак, під час роботи зі складнішими завданнями або великими проектами, може знадобитися більш традиційний підхід програмування для досягнення найкращих результатів.

Підтримувані платформи Construct 2

Основні експортні платформи побудовано на HTML5. Це означає підтримку через Google Chrome, Firefox, Internet Explorer 9+, Safari 6+ і Opera 15+ серед настільних браузерів, а також підтримкою Safari в iOS 6+, Chrome та Firefox для Android, Windows Phone 8+, BlackBerry 10+ і Tizen.

Крім того, Construct 2 здатен експортувати в деякі інтернет-майданчики та магазини, серед яких Facebook, інтернет-магазин Chrome, Amazon Appstore, власний майданчик Scirra та Kongregate.

Настільні операційні системи

Construct 2 забезпечує прямий експорт до Windows, OS X, 32-бітні та 64-бітні версії Linux, за допомогою Node-Webkit. Експорт робиться задля того, щоб користуватися деякими особливостями, які не підтримуються HTML5. 23 жовтня 2012 року Scirra також оголосила про повну підтримку нового інтерфейсу Windows 8 під назвою Metro, в тому числі покупки в додатку, сенсорний ввід, керування плитками, акселерометр тощо.

1.3. Список необхідних функцій інформаційної системи, інформація про технології, які можна використати для реалізації цих функцій.

Замовлення послуг: на сайті повинна бути форма замовлення послуг, у якій клієнти можуть залишати заявки на конкретні проекти або звертатися за консультацією. Для цього потрібно створити базу даних: вибрати базу даних, яку використовуватиме система, і створити необхідні таблиці та поля для зберігання інформації. Можна використовувати, наприклад, MySQL або PostgreSQL. Перед цим треба розробити інтерфейс для користувача, щоб він міг взаємодіяти з нашою системою. Для цього можна використовувати HTML, CSS і JavaScript.

Також на сайті повинна бути панель адміністратора, потрібно написати код на PHP: використати мову програмування PHP для створення логіки системи. У PHP можна використовувати з'єднання з базою даних для доступу до даних.

Потім потрібно протестувати систему на відповідність вимогам і виправити помилки та недоліки, після цього розгорнути систему на сервері, щоб вона була доступна користувачам.

В майбутньому на сайт можливо буде додати відгуки клієнтів, які раніше користувалися послугами архітекторів. Це може допомогти новим клієнтам оцінити якість роботи компанії.

1.3.1. Технології, що будуть використовуватися для реалізації

MySQL

MySQL - це одна з найпопулярніших реляційних баз даних, яка широко використовується для зберігання та управління структурованими даними.

Ось деяка інформація про MySQL:

Основні характеристики:

- Реляційна модель даних: MySQL використовує реляційну модель, що базується на таблицях з певними полями та зв'язками між ними;
- Багатокористувацький доступ: MySQL підтримує одночасний доступ декількох користувачів до бази даних і забезпечує контроль доступу та безпеку даних;
- Мова запитів: для роботи з даними в MySQL використовується SQL (Structured Query Language), стандартна мова запитів баз даних;
- Типи даних: MySQL підтримує різні типи даних, включно з числами, рядками, датами, часами, бінарними даними та іншими. Типи даних можуть бути визначені при створенні таблиці і використовуються для забезпечення цілісності та відповідності даним;
- Індеси: MySQL дає змогу створювати індеси для прискорення пошуку та сортування даних. Індеси створюються на одному або декількох полях таблиці і допомагають значно підвищити продуктивність запитів до бази даних;
- MySQL підтримує транзакції для забезпечення цілісності даних. Транзакції дають змогу виконувати групу операцій як єдине ціле, гарантуючи, що або всі операції успішно завершаться, або жодна не буде виконана;
- Також MySQL надає можливості для масштабування бази даних, даючи змогу керувати великим обсягом даних і забезпечувати високу продуктивність. Він підтримує реплікацію даних, поділ даних на кілька серверів та інші техніки масштабування.

MySQL є відкритим вихідним кодом і має велику спільноту розробників. Це забезпечує доступність великої документації, форумів підтримки, навчальних матеріалів і розширень, що сприяють розробці та підтримці баз даних MySQL.

PHP

PHP - це популярна серверна мова програмування, яку широко використовують для розроблення динамічних веб-додатків. Ось деяка інформація про PHP:

- PHP виконується на стороні сервера, що означає, що код PHP обробляється на веб-сервері перед надсиланням результату на клієнтський браузер;
- PHP може бути вбудований в HTML-код, що дає змогу створювати динамічні веб-сторінки, генерувати контент і взаємодіяти з базами даних;
- PHP підтримує безліч баз даних, включно з MySQL, PostgreSQL, Oracle, SQLite та інші, що робить його хорошим вибором для роботи з даними.

Синтаксис і мовні особливості:

Створення динамічних веб-сторінок: PHP дає змогу впроваджувати динамічний контент у HTML-сторінки, наприклад, виводити дані з бази даних або обробляти форми.

Розробка веб-додатків: PHP надає функціональність для створення повноцінних веб-додатків, таких як системи управління контентом (CMS), інтернет-магазини, форуми, блоги та інші.

Робота з базами даних: PHP має широку підтримку для роботи з різними базами даних, даючи змогу виконувати запити, вставляти, оновлювати і витягувати дані з бази даних.

HTML

HTML - це мова розмітки, яка використовується для створення структури та представлення веб-сторінок. Ось деяка інформація про HTML:

Розмітка вмісту: HTML дозволяє описувати структуру веб-сторінки, визначати заголовки, параграфи, списки, таблиці, зображення та інші елементи вмісту.

HTML дозволяє створювати гіперпосилання (посилання), які дозволяють користувачам переходити різними сторінками та ресурсами в Інтернеті.

Також HTML має різні версії, де HTML5 є останньою та найпоширенішою версією, саме її ми і будемо використовувати.

Основні елементи HTML:

HTML-елементи полягають у теги, які визначають тип елемента та його властивості. Наприклад, <p> використовується для позначення параграфа, <h1>-<h6> - для заголовків різного рівня, - для вставки зображень тощо.

HTML-елементи можуть мати атрибути, які надають додаткову інформацію про елемент. Наприклад, src та alt - атрибути елемента , href - атрибут елемента <a> (гіперпосилання).

Елементи HTML можуть бути вкладені один в одного, утворюючи ієрархічну структуру. Це допомагає визначити відносини та організацію вмісту на веб-сторінці.

Семантична розмітка:

HTML5 надає семантичні елементи, які дозволяють давати більш точне значення та мету елементам сторінки. Наприклад, <header>, <nav>, <article>, <section>, <footer> та інші елементи допомагають визначити структуру та смислове навантаження вмісту.

HTML дозволяє створювати форми, які дозволяють користувачам надсилати дані на сервер. Елементи форми, такі як <input>, <select>, <textarea> і <button>, використовуються для введення даних та надсилання їх на сервер для обробки.

JavaScript

JavaScript (JS) - це високорівнева мова програмування, яка широко використовується для розроблення інтерактивних веб-сторінок і додатків. Вона була створена для додавання динамічної поведінки на веб-сторінки і працює в браузері користувача.

Ось деякі ключові особливості JavaScript:

- **Інтерактивність на стороні клієнта:** JavaScript дає змогу створювати інтерактивні веб-сторінки, які можуть реагувати на дії користувача без необхідності звернення до сервера. Він може змінювати вміст сторінки, обробляти події, валідувати форми тощо.
- **Об'єктно-орієнтоване програмування:** JavaScript підтримує об'єктно-орієнтовану парадигму програмування. Ви можете створювати об'єкти, визначати їхні властивості та методи, успадковувати функціональність тощо.
- **Динамічна типізація:** У JavaScript змінні не потрібно оголошувати з явною вказівкою типу даних. Тип змінної може змінюватися динамічно під час виконання програми.
- **Функції вищого порядку:** JavaScript підтримує функції вищого порядку, що означає, що функції можуть приймати інші функції як аргументи і повертати їх як результат. Це дає змогу використовувати функції для абстракції та повторного використання коду.

Потужні можливості роботи з DOM: Document Object Model (DOM) представляє структуру HTML-документа у вигляді дерева об'єктів, якими можна маніпулювати за допомогою JavaScript. Ви можете змінювати вміст сторінки, додавати і видаляти елементи, застосовувати стилі тощо.

Асинхронне програмування: JavaScript підтримує асинхронні операції з використанням колбеків, промісів і асинхронних функцій (`async/await`). Це дає змогу виконувати завдання без блокування основного потоку виконання, що особливо корисно під час виконання мережевих запитів і обробки подій.

Багатофункціональність: JavaScript може використовуватися не тільки в браузері, а й на сервері з використанням платформи Node.js. Це дає змогу розробляти повноцінні веб-додатки як на стороні клієнта, так і на стороні сервера, використовуючи одну мову програмування.

З плином часу JavaScript став однією з найпопулярніших мов програмування, і її екосистема постійно розвивається. Існує безліч фреймворків і бібліотек, як-от React, Angular і Vue.js, які полегшують розробку складних веб-додатків. JavaScript також використовують для розроблення мобільних застосунків (з використанням фреймворків, таких як React Native і NativeScript) і навіть настільних застосунків (з використанням фреймворка Electron).

JavaScript має багато плюсів і мінусів у веб-розробці. Нижче будуть перераховані основні з них.

Плюси JavaScript:

- JavaScript є однією з найбільш широко підтримуваних мов програмування у веб-розробці. Практично всі сучасні браузери підтримують виконання JavaScript, що дає змогу створювати крос-платформні веб-додатки;
- Інтерактивність і динамічність: JavaScript дає змогу додавати інтерактивність і динамічну поведінку на веб-сторінки. Ви можете створювати анімації, реагувати на дії користувача, валідувати форми, оновлювати вміст сторінки без перезавантаження і багато іншого;
- JavaScript має потужні засоби для маніпуляції DOM. Ви можете легко знаходити, створювати, змінювати і видаляти елементи, а також застосовувати стилі і обробляти події на веб-сторінці;
- Велика екосистема: JavaScript має величезну екосистему різних фреймворків, бібліотек та інструментів, які полегшують розробку. Фреймворки, такі як React, Angular і Vue.js, пропонують потужні інструменти для створення складних веб-додатків;

- Універсальність: JavaScript може бути використаний як на стороні клієнта (у браузері), так і на стороні сервера з використанням платформи Node.js. Це дає змогу розробляти повноцінні веб-додатки, що працюють на різних рівнях стека.

Мінуси JavaScript:

- Відмінності в підтримці браузерами: Різні браузери можуть мати різну підтримку JavaScript і його функціональності. Це може створювати складнощі в підтримці та забезпеченні сумісності з різними браузерами;
- Безпека: Через можливість виконання коду на стороні клієнта, JavaScript вразливий до вразливостей безпеки, таких як міжсайтовий скриптинг (XSS) та ін'єкції коду;
- Продуктивність: JavaScript, що виконується на стороні клієнта, може бути менш продуктивним, ніж мови програмування, що компілюються. Некоректне використання JavaScript може призвести до повільного завантаження сторінки та уповільнення роботи веб-додатка;
- Обмежений доступ до ресурсів пристрою: JavaScript у браузері має обмежений доступ до ресурсів пристрою, таких як файлова система або операційна система. Це обмеження накладається з міркувань безпеки;
- Проблеми масштабованості: Під час розроблення великих веб-додатків на чистому JavaScript може виникати складність у підтримці та масштабуванні кодової бази. Однак використання фреймворків може впоратися з цією проблемою.

Загалом, JavaScript є потужною і широко застосовуваною мовою програмування у веб-розробці, але вона має свої обмеження і проблеми, з якими розробники мають бути ознайомлені та враховувати під час створення веб-додатків.

1.4. Висновки до Розділу 1

Висновок першого розділу підкреслює важливість веб-сайтів у сучасному світі та їхню роль у просуванні послуг компанії з надання архітектурних послуг. У першому розділі розділу глави було показано, що веб-сайт є не тільки інформаційним ресурсом, а й потужним інструментом для залучення нових клієнтів і зміцнення іміджу компанії.

Потім були представлені основні аспекти програмного забезпечення, необхідного для створення веб-сайту. Це включало в себе опис різних програмних інструментів і платформ, які дають змогу розробникам створювати функціональні та естетично привабливі веб-сайти. Цей розділ підкреслює важливість вибору правильного інструментарію для розробки сайту, з урахуванням вимог і цілей компанії.

У заключній частині глави було розглянуто технології, що використовуються під час розроблення веб-сайтів, такі як PHP, MySQL і JavaScript. PHP є однією з найпоширеніших мов програмування для розробки динамічних веб-сайтів. MySQL, своєю чергою, є потужною системою управління базами даних, яка забезпечує ефективне зберігання та управління даними на веб-сайті. JavaScript, у свою чергу, дає змогу створювати інтерактивність на веб-сторінках, що значно підвищує користувацький досвід.

У висновку наголошено на основних перевагах і недоліках JavaScript під час розроблення сайтів. JavaScript забезпечує динамічну та інтерактивну взаємодію з користувачами, що робить сайт більш привабливим і функціональним. Однак, некоректне використання JavaScript може призвести до проблем із безпекою та продуктивністю, тому важливо ретельно планувати та тестувати його використання.

Загалом, перший розділ підкреслює важливість веб-сайтів у сучасному бізнесі та представляє основні технології та інструменти, необхідні для їхнього розроблення.

РОЗДІЛ 2

АНАЛІЗ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА РОЗРОБКА ЗАГАЛЬНОЇ СХЕМИ ВЕБ-САЙТУ

Для роботи з базами даних потрібні програмні засоби, які забезпечуватимуть доступ до необхідної інформації, внесення певних змін до БД та інші дії з даними. Для вирішення цього завдання використовується Система Управління Базами Даних (СУБД)

Система управління базами даних - це програмне забезпечення, що дає змогу організувати зберігання, управління та обробку даних у базі даних. Вона забезпечує доступ до даних, їх додавання, зміну та видалення, а також виконання запитів та аналітичних операцій.

2.1. Основні завдання СУБД

Основні завдання СУБД включають у себе:

- Створення та визначення структури бази даних: СУБД дає змогу визначити схему бази даних, включно з таблицями, полями, зв'язками між таблицями та іншими елементами, необхідними для організації даних;
- Зберігання даних: СУБД забезпечує механізми для ефективного зберігання даних на диску або в інших сховищах. Вона керує способом фізичного розподілу даних і забезпечує доступ до них за потреби;
- Маніпулювання даними: СУБД надає можливість додавати, змінювати та видаляти дані в базі даних. Вона виконує операції

Кафедра КІТ (47)				НАУ 23 17 60 000 ПЗ			
Виконав	Погорелов Д.О.			АНАЛІЗ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА РОЗРОБКА ЗАГАЛЬНОЇ СХЕМИ ВЕБ- САЙТУ	Літера	Аркуш	Аркушів
Керівник	Харченко О.Г.					28	10
Консультант					<i>УС-411 122</i>		
Н.Контроль	Шевченко О..						

вставки, оновлення та видалення даних згідно із заданими правилами та обмеженнями;

- Запити та обробка даних: СУБД дає змогу виконувати складні запити до даних. Вона надає мову запитів, за допомогою якої можна витягувати інформацію з бази даних, а також проводити агрегацію, сортування, фільтрацію та інші операції обробки даних;
- Безпека даних: СУБД забезпечує механізми захисту даних від несанкціонованого доступу та пошкодження. Вона дає змогу встановлювати права доступу до даних для різних користувачів і ролей, а також забезпечує механізми резервного копіювання та відновлення даних;
- Забезпечення цілісності даних: СУБД підтримує цілісність даних шляхом застосування обмежень, перевірок і зв'язків між таблицями. Вона гарантує дотримання заданих правил і умов під час додавання, зміни та видалення даних.

СУБД є ключовим компонентом для управління даними в різних організаціях і додатках. Вона полегшує роботу з даними, підвищує ефективність і надійність та надає потужні інструменти для обробки інформації.

Існує кілька видів систем управління базами даних, кожна з яких має свої особливості та застосовується в різних сценаріях. Нижче наведено деякі поширені види СУБД:

- Реляційні СУБД (Relational DBMS): Це найпоширеніший тип СУБД. Реляційні СУБД організують дані у вигляді таблиць, що складаються з рядків (кортежів) і стовпців (атрибутів). Вони підтримують SQL (Structured Query Language) для виконання запитів до даних. Приклади реляційних СУБД включають Oracle Database, MySQL, PostgreSQL, Microsoft SQL Server;
- Ієрархічні СУБД (Hierarchical DBMS): Ієрархічні СУБД організують дані у вигляді ієрархічної структури, що складається з набору

пов'язаних записів. Кожен запис може мати одного або кількох нащадків. Цей тип СУБД широко використовували в минулому, але нині він не такий поширений. Приклади ієрархічних СУБД включають IBM's Information Management System (IMS) і Windows Registry;

- **Мережеві СУБД (Network DBMS):** Мережеві СУБД організують дані у вигляді мережі, що складається з набору пов'язаних записів. Кожен запис може мати кілька зв'язків з іншими записами. Мережеві СУБД також не так поширені, як реляційні СУБД. Приклади мережевих СУБД включають Integrated Data Store (IDS) та Integrated Database Management System (IDMS);
- **Об'єктно-орієнтовані СУБД (Object-Oriented DBMS):** Об'єктно-орієнтовані СУБД зберігають дані у вигляді об'єктів, що можуть містити дані та методи для їх обробки. Вони надають розширені можливості для зберігання і роботи зі складними структурами даних. Приклади об'єктно-орієнтованих СУБД включають MongoDB, Cassandra, ObjectStore;
- **NoSQL СУБД (Not Only SQL DBMS):** NoSQL СУБД призначені для роботи з великими обсягами неструктурованих даних, які не відповідають традиційній реляційній моделі. Вони пропонують гнучку схему даних, масштабованість і високу продуктивність. Приклади NoSQL СУБД включають MongoDB, Cassandra, Redis;
- **Інтернет-орієнтовані СУБД (Internet-oriented DBMS):** Цей тип СУБД розроблено для опрацювання даних, пов'язаних з Інтернетом, наприклад вебсторінки, журнали серверів, користувацькі дані тощо. Вони пропонують масштабованість, високу продуктивність і можливість роботи з розподіленими системами. Приклади інтернет-орієнтованих СУБД включають Google Bigtable, Apache Cassandra, Amazon DynamoDB.

Крім того, існують й інші спеціалізовані види СУБД, наприклад тимчасові СУБД, просторові СУБД, аналітичні СУБД тощо, які призначено

для конкретних завдань і сфер застосування. Вибір конкретного типу СУБД залежить від вимог вашого проєкту і характеру даних, які необхідно зберігати й обробляти.

У сучасних системах управління базами даних використовуються різні мови опису та маніпуляції даними. Нижче наведено деякі з найпоширеніших мов, які використовуються в сучасних СУБД:

- SQL (Structured Query Language): SQL є стандартною мовою запитів і маніпуляції даними в реляційних СУБД. Вона дає змогу створювати таблиці, визначати структуру бази даних, вставляти, оновлювати та видаляти дані, а також виконувати складні запити для вилучення інформації. SQL є декларативною мовою, де ви описуєте, що потрібно зробити, а не як це зробити.
- PL/SQL (Procedural Language/SQL): PL/SQL є розширенням SQL, що надає можливості для написання процедур, функцій, тригерів і пакетів у базі даних Oracle. Він дає змогу створювати складніші програмні логіки на основі SQL операцій.
- T-SQL (Transact-SQL): T-SQL є розширенням SQL, використовуваним у СУБД Microsoft SQL Server. Він надає додаткові можливості, як-от керування транзакціями, обробка помилок, визначення збережених процедур і функцій.
- PL/pgSQL (Procedural Language/PostgreSQL): PL/pgSQL є мовою програмування, яку підтримують у СУБД PostgreSQL. Вона надає засоби для створення збережених процедур, функцій і тригерів у базі даних.
- NoSQL Query Languages: Для деяких типів NoSQL СУБД, таких як MongoDB і Cassandra, використовуються спеціалізовані мови запитів. Наприклад, MongoDB використовує мову запитів MongoDB Query Language (MQL), а Cassandra надає мову запитів CQL (Cassandra Query Language).

Також існують й інші мови та стандарти, які використовуються для опису та маніпуляції даними, такі як XQuery (для роботи з XML даними), SPARQL (для запитів до RDF даних), та інші.

Вибір мови залежить від конкретної СУБД і її підтримуваних можливостей, а також від вимог і цілей проекту.

2.2. Порівняльна таблиця різних систем управління базами даних.

Порівняльна таблиця різних систем управління базами даних (СУБД) за деякими основними критеріями:

- Тип;
- Мова запитів;
- Модель даних;
- Поширеність;
- Операційні системи.

Таблиця 2.1

Порівняльна характеристика СУБД

Назва	Тип	Мова запитів	Модель даних	Поширеність	Операційні системи
MySQL	Реляційна	SQL	Реляційна	Широко використовувана	Крос-платформенна
PostgreSQL	Реляційна	SQL	Реляційна	Широко використовувана	Крос-платформенна
Oracle Database	Реляційна	SQL	Реляційна	Широко використовувана	Крос-платформенна

Назва	Тип	Мова запитів	Модель даних	Поширеність	Операційні системи
Microsoft SQL Server	Реляційна	SQL	Реляційна	Широко використовувана	Windows
MongoDB	Документна	MongoDB	Документна	Зростаюча популярність	Крос-платформенна
Redis	Ключ-значення	Redis	Ключ-значення	Зростаюча популярність	Крос-платформенна

Для інформаційної системи, що розробляється, у якості СУБД була обрана MySQL.

2.2.1. Основні переваги MySQL:

Простота використання: MySQL має простий та інтуїтивно зрозумілий синтаксис SQL, що полегшує розробку та адміністрування баз даних. Вона має широку підтримку і велику спільноту користувачів, що полегшує пошук допомоги та вирішення проблем.

Хороша продуктивність: MySQL має хорошу продуктивність і масштабованість. Вона здатна обробляти великі обсяги даних і підтримує реплікацію та шардинг для розподілу навантаження.

Широка підтримка та інтеграція: MySQL є однією з найпопулярніших СУБД, тому вона має широку підтримку з боку інструментів, бібліотек і платформ розробки. MySQL також підтримує безліч драйверів для різних мов програмування.

Гнучкість: MySQL пропонує різні рушії зберігання даних, наприклад InnoDB, MyISAM та інші, даючи змогу вибрати відповідний рушій залежно

від вимог проєкту. Крім того, ця СУБД підтримує різні операційні системи і платформи.

Велика спільнота і ресурси: MySQL має величезне активне співтовариство користувачів і розробників, що забезпечує доступність навчальних матеріалів, документації та рішень проблем. Також є безліч сторонніх інструментів і розширень, які полегшують роботу з MySQL.

2.3. Розробка логічної моделі бази даних

Для розробки логічної моделі даних необхідно визначити: які дані треба зберігати у нашій базі даних. До таких даних входять:

- Ім'я користувача;
- Номер телефону користувача;
- Обраний тип об'єкту;
- Площа об'єкту;
- Точний час відправки форми.

Таблиця 2.2

Перелік сутностей логічної моделі даних

Назва	Опис
Користувач	Повне ім'я користувача
Обраний тип об'єкту	id обраного чекбокса
Час відправки форми	Час відправки форми у текстовому форматі. Наприклад (Tue May 23 2023 12:36:32 GMT 0300)
Площа об'єкту	Площа обраного об'єкту, що вказана користувачем (у м ²)
Номер телефону користувача	Номер телефону вказаний користувачем веб-сайту

2.4 phpMyAdmin

phpMyAdmin - це безкоштовний веб-інтерфейс для керування базами даних MySQL. Він надає зручний спосіб взаємодії з базою даних MySQL через веб-браузер, що полегшує адміністрування баз даних для розробників і адміністраторів.

Нижче перераховані деякі особливості phpMyAdmin:

- phpMyAdmin дає змогу створювати, видаляти та змінювати бази даних. Ви можете створювати таблиці, змінювати їхню структуру, додавати і видаляти стовпці, а також керувати індексами, обмеженнями та іншими аспектами бази даних;
- phpMyAdmin надає вбудований редактор SQL, що дозволяє виконувати SQL-запити безпосередньо з веб-інтерфейсу. Ви можете виконувати SELECT, INSERT, UPDATE, DELETE та інші типи запитів, а також переглядати результати запитів.
- Ви можете імпортувати й експортувати дані в різних форматах, включно з SQL, CSV, XML та іншими. Це корисно під час перенесення даних з однієї бази даних в іншу або під час створення резервних копій даних.
- phpMyAdmin дає змогу створювати і керувати користувачами бази даних MySQL. Ви можете встановлювати привілеї доступу до баз даних, таблиць та інших об'єктів бази даних.
- phpMyAdmin надає інтуїтивний спосіб перегляду і редагування даних у таблицях. Ви можете переглядати вміст таблиці, редагувати значення, сортувати дані та застосовувати фільтри.

phpMyAdmin також пропонує інші функції, такі як створення звітів, графічне представлення структури бази даних, аналіз продуктивності запитів та інші інструменти для допомоги в адмініструванні баз даних. phpMyAdmin є дуже популярним інструментом для керування базами даних MySQL, і він

широко використовується веб-розробниками та адміністраторами для роботи з MySQL.

2.5. Висновки до Розділу 2

Висновок другого розділу охоплює основні аспекти, пов'язані з системами керування базами даних (СУБД). На початку розділу було описано основні завдання, які розв'язує СУБД, як-от зберігання й керування даними, забезпечення безпеки, забезпечення цілісності даних і ефективне виконання запитів. Це підкреслює важливість використання відповідної СУБД для створюваної інформаційної системи.

Після цього було представлено порівняльну таблицю різних СУБД, де було наведено їхні основні характеристики, переваги та недоліки. Це допомогло читачеві порівняти різні варіанти і вибрати найбільш підходящу СУБД для свого проекту.

Далі було детально описано одне з найпопулярніших рішень - MySQL. Було представлено його переваги, такі як широка підтримка, висока продуктивність, надійність і простота використання. Описані переваги MySQL підтверджують його ефективність у створенні та управлінні базами даних для інформаційних систем.

У розділі також було представлено маленьку логічну модель майбутньої бази даних для створюваної інформаційної системи. Це дає уявлення про те, як дані будуть організовані та пов'язані між собою, що є важливою частиною процесу розроблення бази даних.

У заключній частині розділу було представлено опис інтерфейсу і можливостей phpMyAdmin - популярного інструменту для адміністрування баз даних MySQL. Опис phpMyAdmin дає змогу читачеві зрозуміти, як керувати базою даних, виконувати запити, створювати таблиці та багато іншого за допомогою цього інтуїтивно зрозумілого інтерфейсу.

Загалом, другий розділ надає огляд основних завдань, пов'язаних із СУБД, пропонує порівняльний аналіз різних варіантів СУБД, підкреслює переваги MySQL, презентує логічну модель бази даних і описує можливості phpMyAdmin. Це створює основу для подальшого розроблення інформаційної системи та роботи з базою даних.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-ПОШУКОВОЇ СИСТЕМИ, НАВІГАЦІЙНОГО МЕНЮ ТА ДИЗАЙНУ ВЕБ-САЙТУ

3.1. Вибір дизайну веб-сайту

При створенні сайтів існує безліч різних стилів дизайну, які можуть бути використані для надання унікальності та естетичного вигляду сайту. Нижче перераховані деякі з популярних стилів дизайну, які часто зустрічаються у веб-розробці:

- Мінімалізм;
- Плоский дизайн;
- Матеріальний дизайн;
- Типографічний дизайн;
- Ілюстраційний дизайн;
- Темний дизайн;
- Градієнтний дизайн;
- Ретро-дизайн.

Мінімалізм: Основна ідея цього стилю полягає у створенні простого і чистого дизайну, з акцентом на простір і типографіку. Використовуються нейтральні кольори, прості форми і мінімум декоративних елементів.

Плоский дизайн: Характеризується застосуванням простих елементів, без використання об'ємних або реалістичних ефектів. Використовуються яскраві кольори, чіткі межі та прості іконки.

Кафедра КІТ (47)				НАУ 23 17 60 000 ПЗ			
Виконав	Погорелов Д.О.			РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-ПОШУКОВОЇ СИСТЕМИ, НАВІГАЦІЙНОГО МЕНЮ ТА ДИЗАЙНУ ВЕБ-САЙТУ	Літера	Аркуш	Аркушів
Керівник	Харченко О.Г.					38	18
Консультант					<i>УС-411 122</i>		
Н.Контроль	Шевченко О..						

Матеріальний дизайн: Розроблений компанією Google, цей стиль об'єднує простоту плоского дизайну з деякими реалістичними елементами, як-от тіней і градієнтів. Він прагне створити відчуття матеріальності та глибини.

Типографічний дизайн: Основна мета цього стилю - використання друкарні як головного елемента дизайну. Використовуються великі, виразні шрифти та цікаві комбінації типографічних елементів.

Ілюстраційний дизайн: Включає використання ілюстрацій і малюнків для створення унікального і привабливого зовнішнього вигляду сайту. Ілюстрації можуть бути абстрактними, реалістичними або стилізованими.

Темний дизайн: Передбачає використання темних колірних схем і фонових відтінків. Цей стиль може створювати елегантний і сучасний вигляд сайту.

Градієнтний дизайн: Використовує плавні переходи між двома або більше кольорами для створення градієнтних ефектів. Градієнти можуть бути використані в різних елементах дизайну, включно з тлом, кнопками і заголовками.

Ретро-дизайн: Натхненний стилем і естетикою минулих десятиліть. Включає використання яскравих кольорів, вінтажних шрифтів, піксельної графіки та інших елементів, що відсилають до старих часів.

Після обговорення з фірмою було обрано монохромний дизайн сайту. Стиль дизайну, в якому переважають сірі тони і суворе оформлення елементів, часто називають "монохромним дизайном" або "сірим дизайном". Цей стиль зазвичай використовує різні відтінки сірого кольору в поєднанні з білим або чорним для створення чистого і мінімалістичного зовнішнього вигляду. Монохромний дизайн може бути дуже елегантним і стриманим. Він зазвичай підкреслює простоту і акцентує увагу на контенті, роблячи його основним фокусом візуального сприйняття. Суворе оформлення елементів доповнює цей стиль, надаючи чіткі межі, прості форми й акуратне розміщення елементів на сторінці.

У монохромному дизайні може використовуватися тільки сірий колір, але також можливе додавання невеликих акцентних кольорів для виділення певних елементів або створення контрасту. Це може бути, наприклад, один яскравий колір, який виділяється на сірому тлі.

Монохромний дизайн може бути дуже ефективним, особливо для сайтів, де важливо передати строгість, професіоналізм або елегантність.

3.2. Створення навігаційного меню

Навігаційне меню створюємо у Construct 2. Для початку створимо фон сірого кольору та 6 кнопок для майбутніх сторінок, які будуть потрібні фірмі та додамо 2 панелі для текстів:

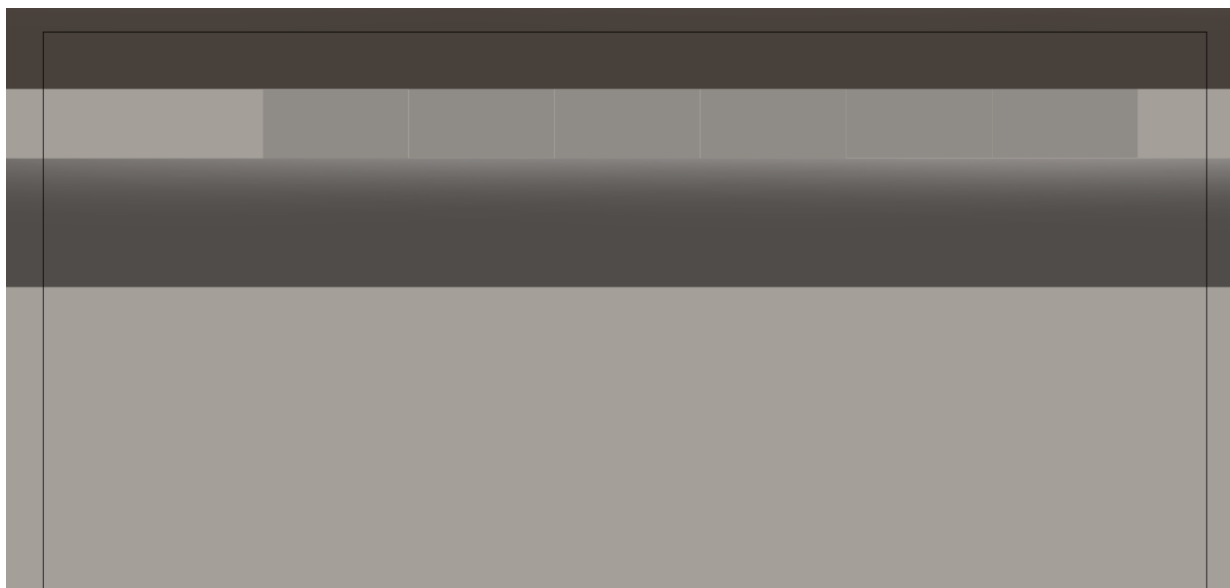


Рис. 3.1. Створення навігаційного меню

Додамо необхідні назви сторінок на кнопки:

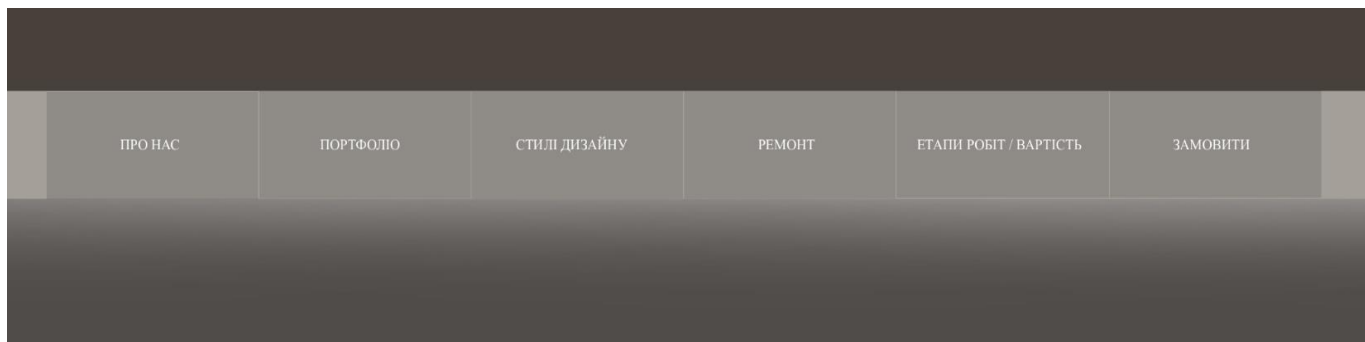


Рис. 3.2. Створення навігаційного меню (додання текстів)

Виставимо шрифт тексту, що пропонує фірма:

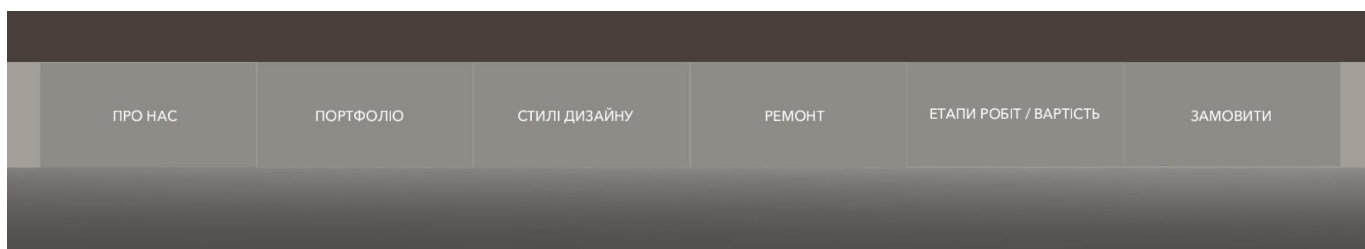


Рис. 3.3. Створення навігаційного меню (оновлені шрифти)

Додамо назву відкритої сторінки та номери телефонів для зв'язку:

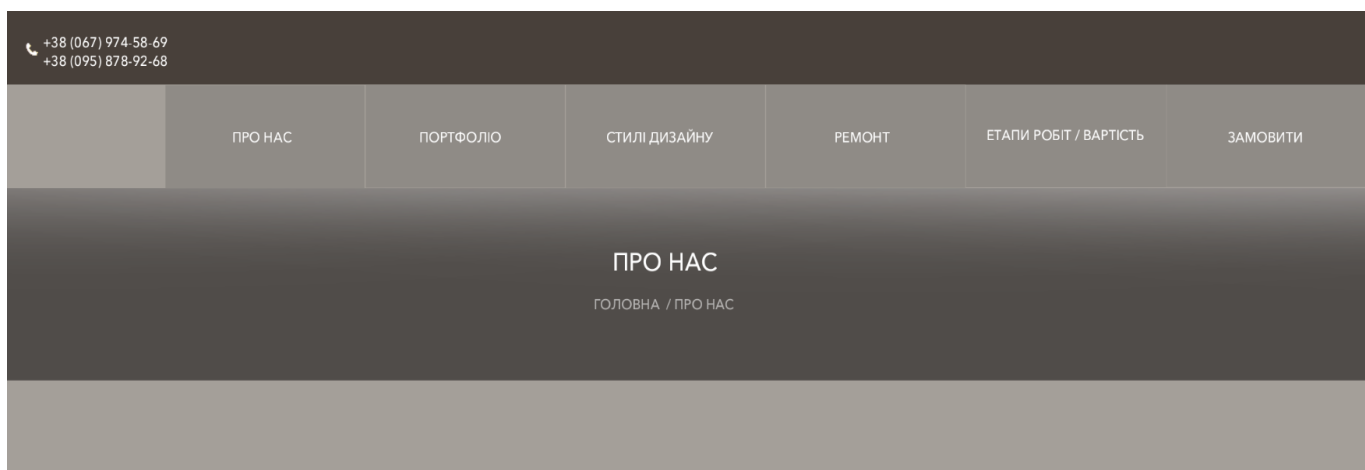


Рис. 3.4. Створення навігаційного меню (додання назви сторінок)

Також додамо логотип фірми:



Рис. 3.5. Логотип ТОВ «ГУЛСТА»

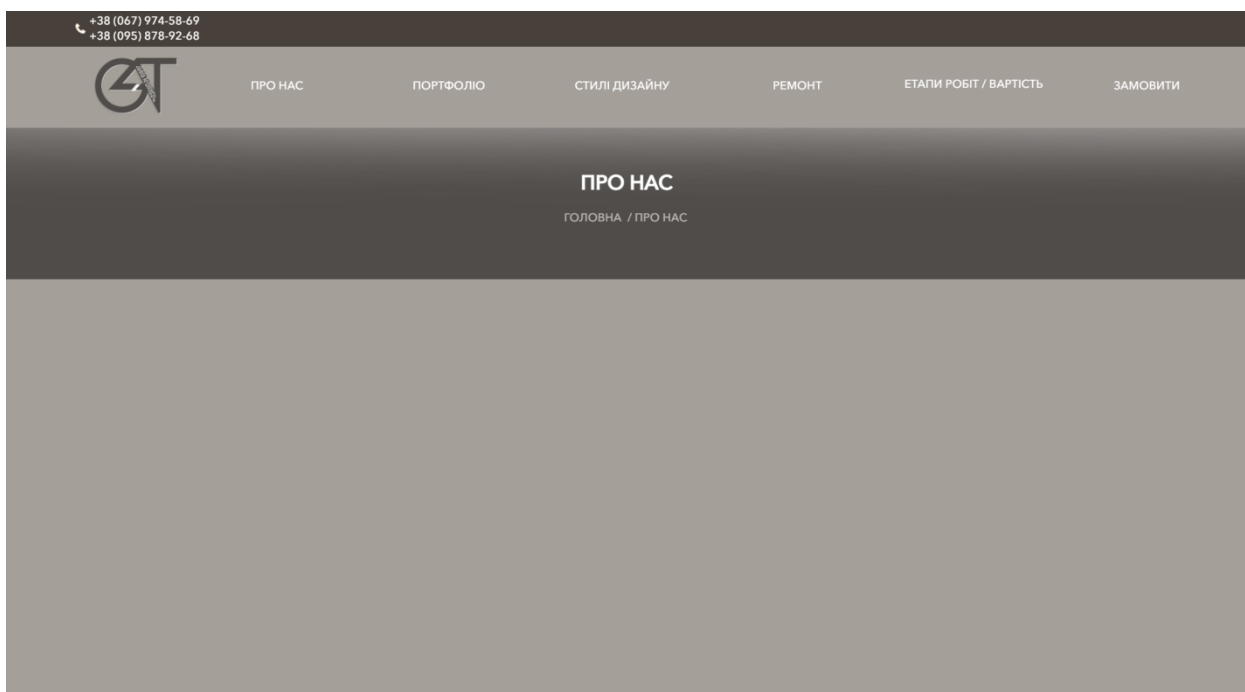


Рис. 3.6. Створення навігаційного меню (додання логотипу)

Додамо прозорий логотип на фоні та анімації при наведенні миші на елементи інтерфейсу:

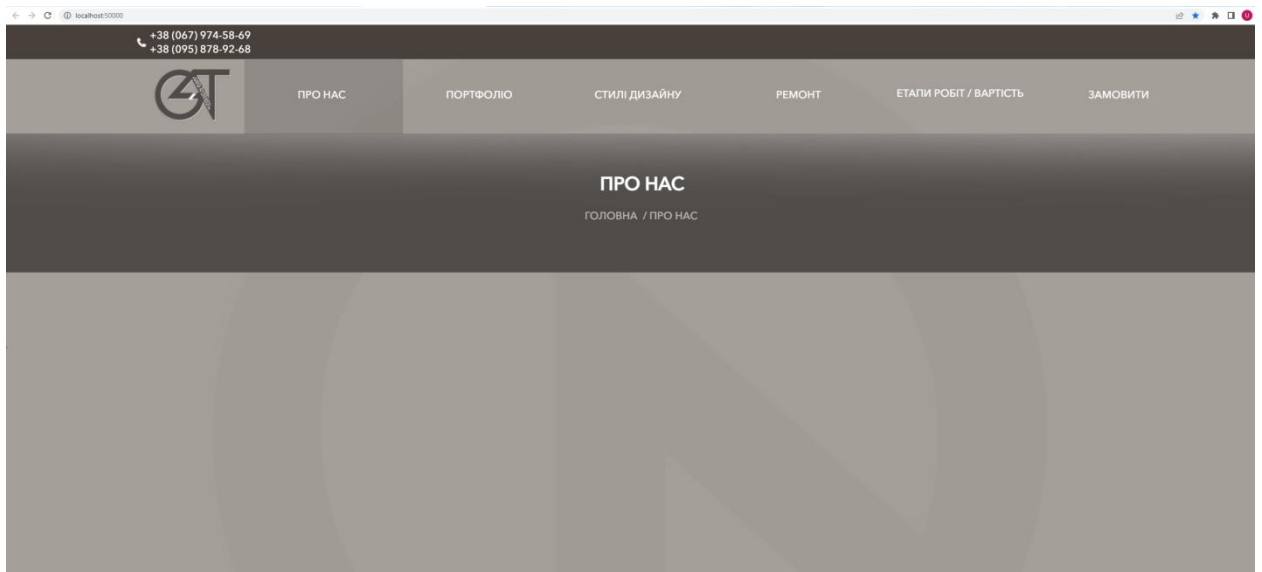


Рис. 3.7. Створення навігаційного меню (додання логотипу на фоні та анімацій меню)

Для кожної кнопки навігаційного меню призначимо унікальний ідентифікатор id для відкриття різних сторінок при натиску.

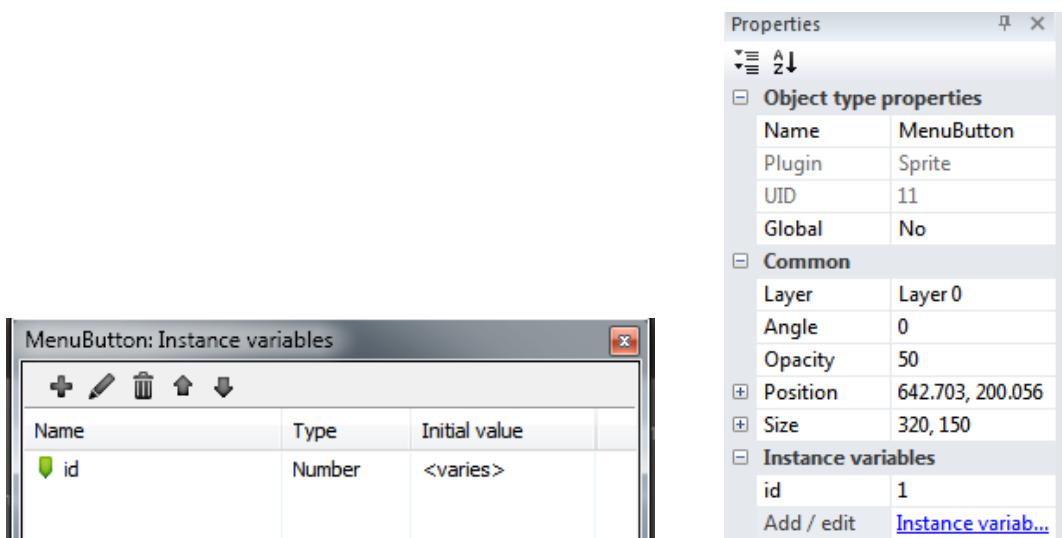


Рис. 3.8. Створення навігаційного меню (призначення id елементам навігаційного меню)

Після цього створимо інші сторінки (шість сторінок для навігаційного меню, та одну зарезервуємо для панелі адміністратора.) :

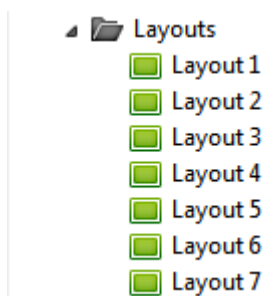


Рис. 3.9. Створення інших сторінок

Додамо події (events) при натиску на елементи меню.

10	Mouse MenuButton	On Left button Clicked on MenuButton id = 6	System Add action	Go to Layout 2
11	Mouse MenuButton	On Left button Clicked on MenuButton id = 2	System Add action	Go to Layout 5
12	Mouse MenuButton	On Left button Clicked on MenuButton id = 3	System Add action	Go to Layout 6
13	Mouse MenuButton	On Left button Clicked on MenuButton id = 4	System Add action	Go to Layout 7
14	Mouse MenuButton	On Left button Clicked on MenuButton id = 1	System Add action	Go to Layout 4
15	Mouse MenuButton	On Left button Clicked on MenuButton id = 5	System Add action	Go to Layout 1

Рис. 3.10. Події (events) у Constuct 2 при натиску на елементи меню.

3.3. Додання інформації фірми на сайт

Станом на зараз фірма передала розділи «про нас», «етапи робіт/вартість» та назви і типи форм для відправки користувачем для замовлення проекту. Розділи «портфоліо», «стилі дизайну» та «ремонт» будуть заповнені коли фірма передасть інформацію.

Розмістимо тексти на сторінках:



Рис. 3.11. Доданий текст на сторінці «ПРО НАС»

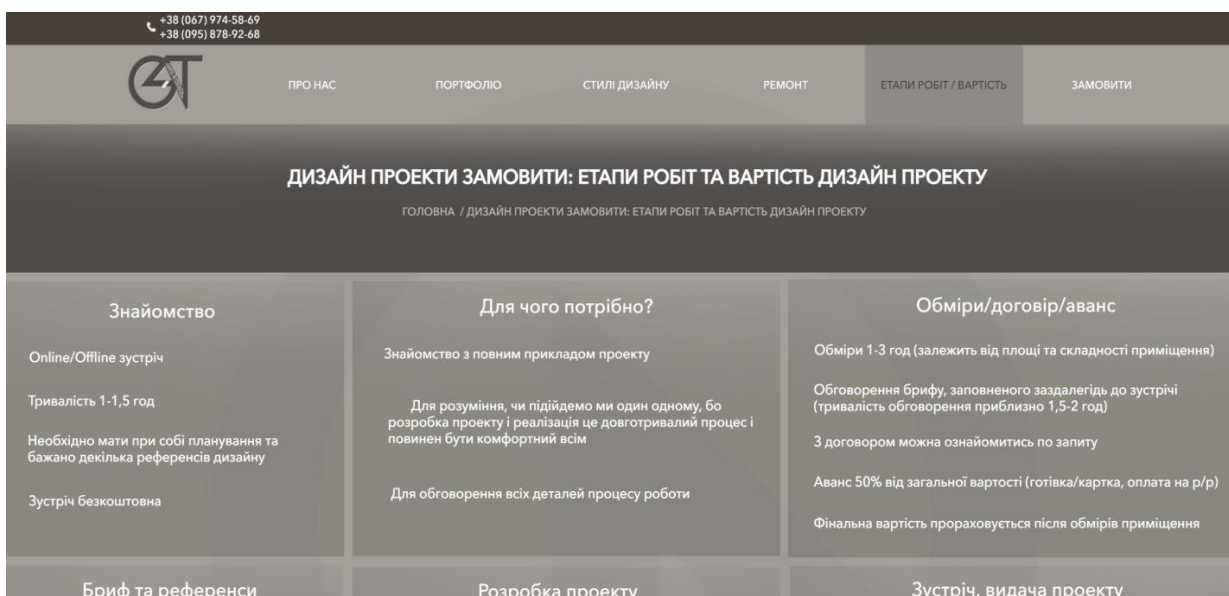


Рис. 3.12. Доданий текст на сторінці «ЕТАПИ РОБІТ/ВАРТІСТЬ»

Після розміщення текстів створимо форму замовлення:

The image shows a web form for ordering a design project. The form is centered on a dark gray background. At the top, there is a dark gray header with the text 'ЗАМОВИТИ ДИЗАЙН-ПРОЕКТ' in white. Below the header, the text 'ГОЛОВНА / ЗАМОВИТИ ДИЗАЙН ПРОЕКТ' is displayed in a smaller font. The form itself is a light gray box with a white border. It contains the following elements: a label 'П.І.Б.' above a text input field with 'Name' as a placeholder; a label 'Телефон' above a text input field with 'Phone number' as a placeholder; a label 'Вид об'єкту' above five radio button options: 'Квартира', 'Дім', 'Офіс', 'Магазин', and 'Ресторан / Кафе'; a label 'Площа' above a text input field with 'm²' as a placeholder; and a 'Відправити' button at the bottom center.

Рис. 3.13. Вигляд форми замовлення дизайн-проекту

3.4. Створення бази даних та таблиці в phpMyAdmin

3.4.1. Створення бази даних

Для створення бази даних перейдемо в панель управління обраного раніше веб хостингу і натиснемо create database.

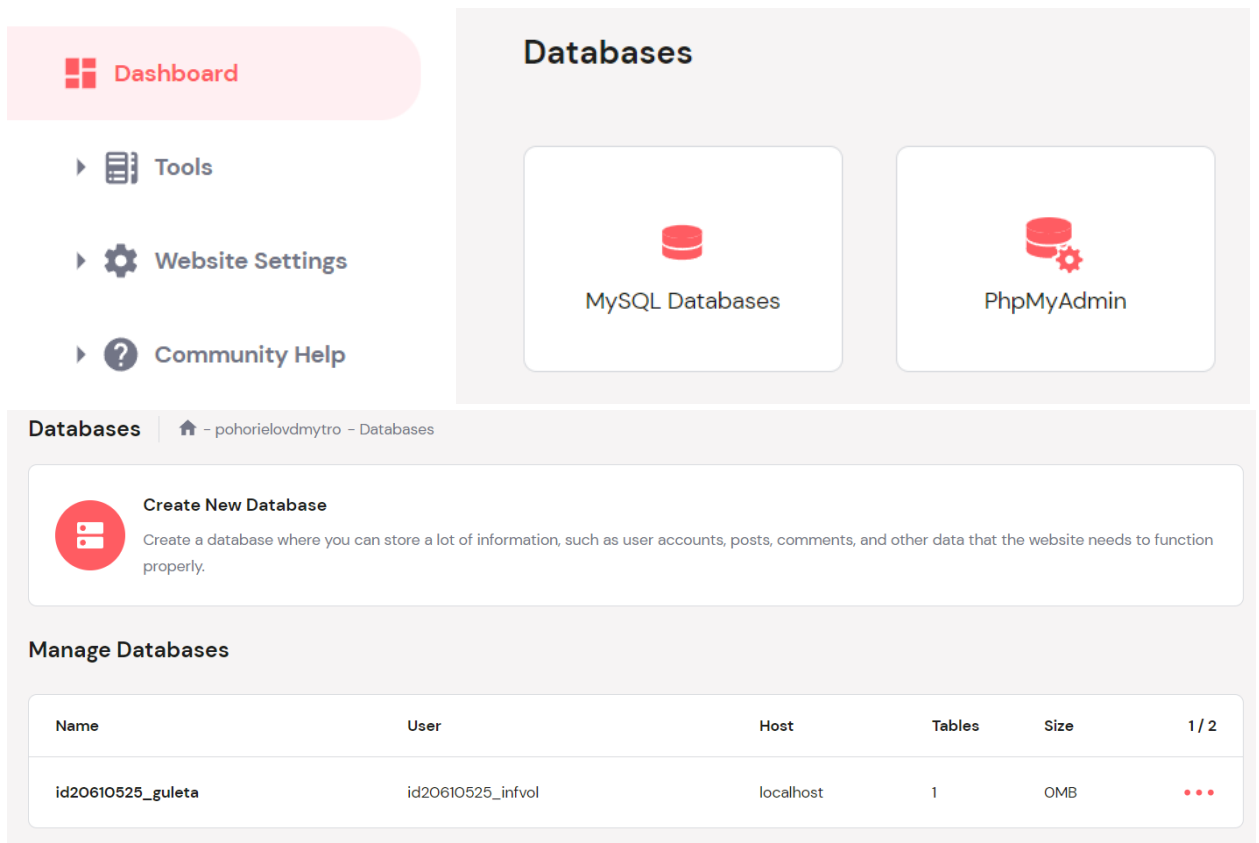


Рис. 3.14. Панель управління 000webhost

3.4.2. Створення таблиці в phpMyAdmin

Дані поля таблиці відображають відповідні дані з полів форми замовлення:

- username - Ім'я користувача;
- typechosen - Вид об'єкту;
- submittime - Час відправки форми;
- sqare - Площа об'єкту;
- data - Номер телефону.

Після створення бази даних можемо перейти у phpMyAdmin та створити таблицю з назвами полів що відповідають формі.

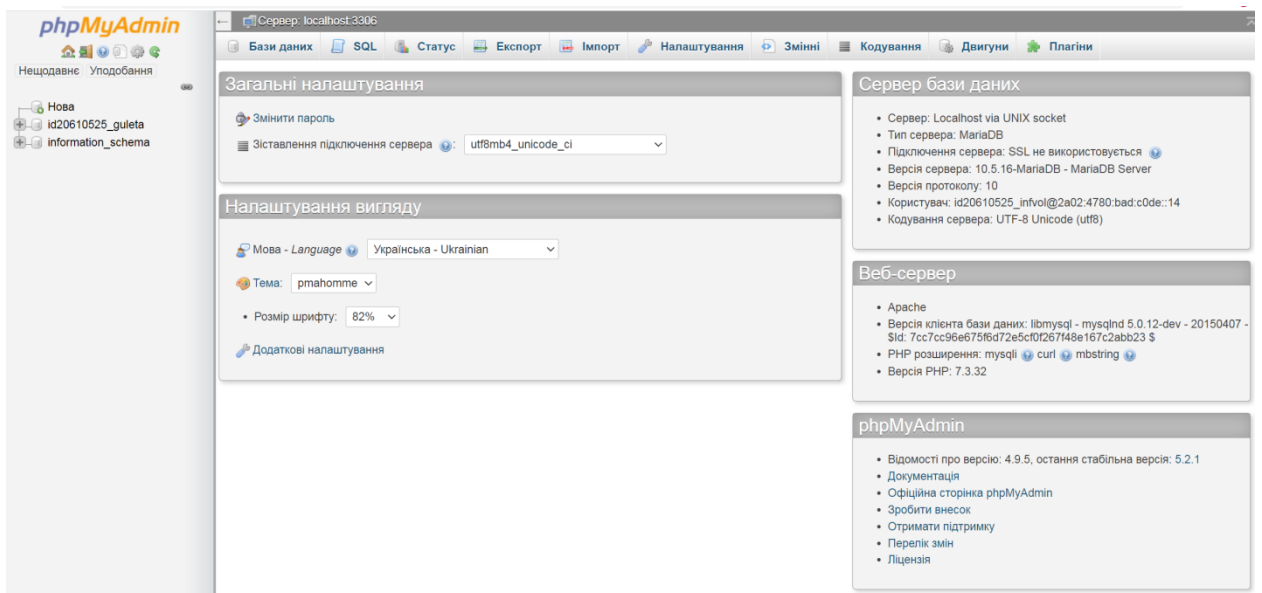


Рис. 3.15. Інтерфейс phpMyAdmin

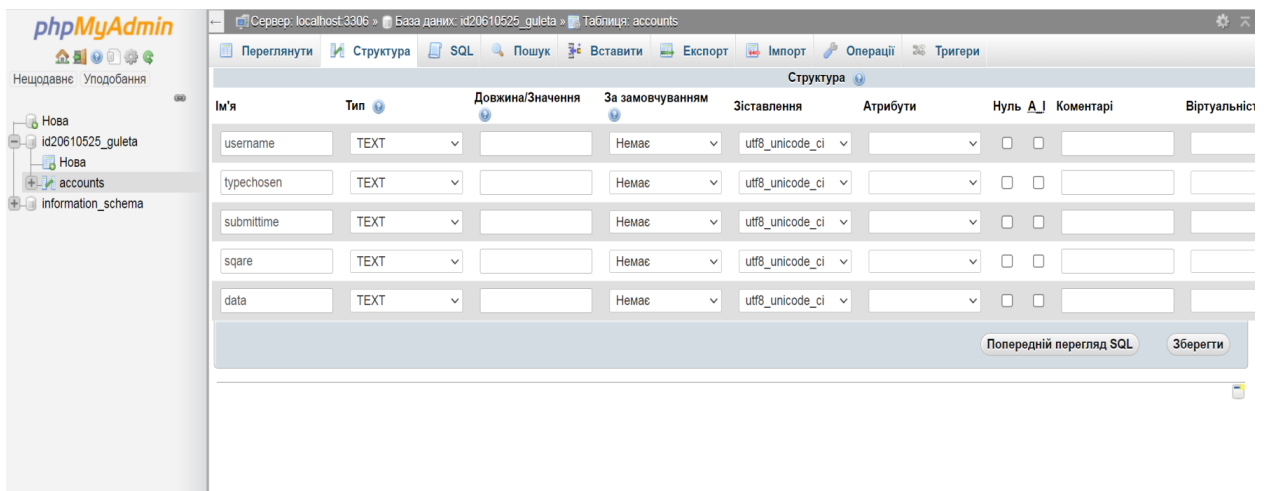


Рис. 3.16. Створення таблиці у phpMyAdmin

3.5. Створення панелі адміністратора

Зробимо відкриття раніше створеної сторінки за нажиманням клавіші F3.

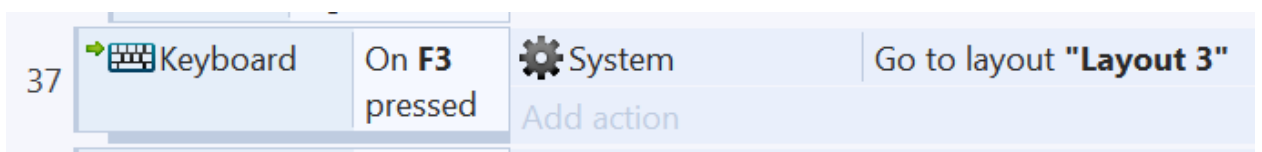


Рис. 3.19. Подія натиску клавіші у Construct 2

Створимо поле для вводу паролю адміністратора:

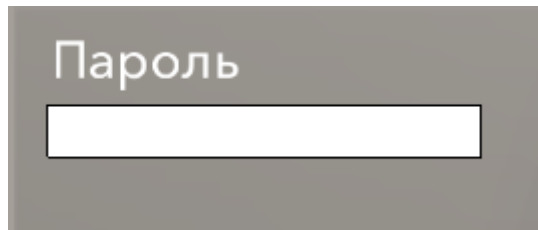


Рис. 3.17. Форма вводу паролю адміністратора

Також додамо інші кнопки управління:

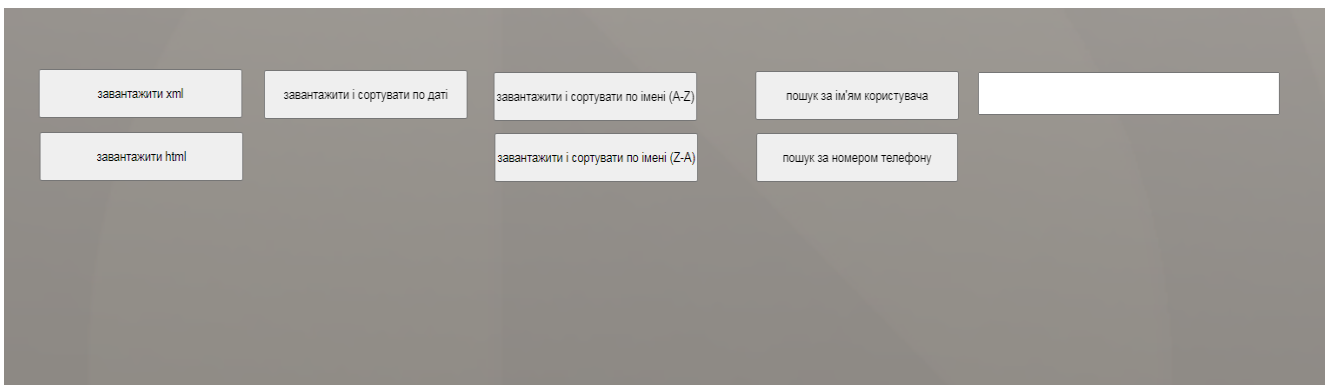


Рис. 3.18. Кнопки управління панелі адміністратора

27	loaddate	On clicked	AJAX	Send "password2" & AdminPass.Text to URL "https://pohorielovdmytro.000webhostapp.com/authsortbydate.php" (method "POST", tag "authsortbydate")
			XML	Load XML document from string AJAX.LastData
			System	Set xml to d.Text&AJAX.LastData&d.Text
			Text5	Set text to xml
			Browser	Execute javascript :JStextcreate.Text
			Text3	Set text to AJAX.LastData
			Add action	
28	loadnameAZ	On clicked	AJAX	Send "password2" & AdminPass.Text to URL "https://pohorielovdmytro.000webhostapp.com/sortbynameAZ.php" (method "POST", tag "authsortbynameAZ")
			XML	Load XML document from string AJAX.LastData
			System	Set xml to d.Text&AJAX.LastData&d.Text
			Text5	Set text to xml
			Browser	Execute javascript :JStextcreate.Text
			Text3	Set text to AJAX.LastData
			Add action	
29	loadnameZA	On clicked	AJAX	Send "password2" & AdminPass.Text to URL "https://pohorielovdmytro.000webhostapp.com/sortbynameZA.php" (method "POST", tag "authsortbynameZA")
			XML	Load XML document from string AJAX.LastData
			System	Set xml to d.Text&AJAX.LastData&d.Text
			Text5	Set text to xml
			Browser	Execute javascript :JStextcreate.Text
			Text3	Set text to AJAX.LastData
			Add action	
30	searchDBbyname	On clicked	AJAX	Send "password2" & AdminPass.Text& &username=" &DatabaseSearch.text to URL "https://pohorielovdmytro.000webhostapp.com/authsearch.php" (method "POST", tag "authsearchdb")
			XML	Load XML document from string AJAX.LastData
			System	Set xml to d.Text&AJAX.LastData&d.Text
			Text5	Set text to xml
			Browser	Execute javascript :JStextcreate.Text
			Text3	Set text to AJAX.LastData
			Add action	
31	loadhtml	On clicked	AJAX	Send "password2" & AdminPass.Text to URL "https://pohorielovdmytro.000webhostapp.com/authhtml.php" (method "POST", tag "authsearchdb")
			XML	Load XML document from string AJAX.LastData
			System	Set xml to d.Text&AJAX.LastData&d.Text
			Text5	Set text to xml
			Text3	Set text to AJAX.LastData
			Browser	Execute javascript :function saveXML(xmlString, fileName) { blob, fileName); appendChild(link);
			Add action	

Рис.3.19. Події (events) у Construct 2 при натиску кнопок управління панелі адміністратора

Тепер потрібно завантажити сайт на хостинг, та написати php файли для перевірки паролю адміністратора та пошуку/сортування, а потім експорту даних з бази.

За замовчуванням в файловому менеджері ми побачимо папку `public_html` та файл `.htaccess` у ній.

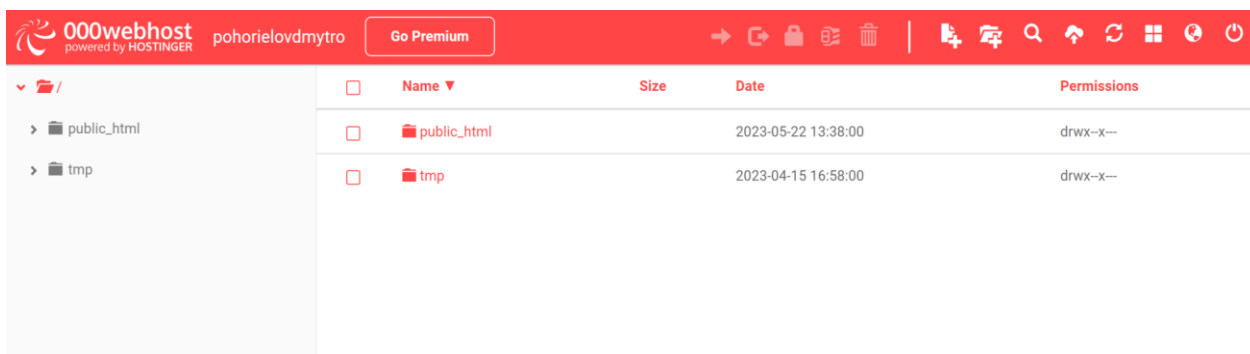


Рис. 3.20. Вигляд файлового менеджера 000webhost

Файл `.htaccess` є конфігураційним файлом, який використовується на сервері веб-сайту, що працює на веб-сервері Apache. Він відповідає за встановлення та перевизначення налаштувань, що стосуються поведінки сервера, а також за керування доступом до файлів і директорій на веб-сайті. Ось деякі основні області, за які відповідає файл `.htaccess`:

Встановлення налаштувань сервера: Файл `.htaccess` дає змогу визначати і змінювати конфігураційні параметри сервера Apache для конкретного веб-сайту або директорії. Наприклад, ви можете використовувати `.htaccess` для ввімкнення або вимкнення певних модулів Apache, зміни параметрів PHP, встановлення користувачьких налаштувань часу виконання тощо

За допомогою `.htaccess` можливо керувати перенаправленнями URL на веб-сайті. Наприклад, ви можете налаштувати переспрямування з однієї URL на іншу, встановити постійні переспрямування (301 Redirects) для оновлення URL-адрес, встановити тимчасові переспрямування (302 Redirects) тощо.

Файл `.htaccess` дає змогу керувати доступом до файлів і директорій на веб-сайті. Ви можете налаштувати різні рівні автентифікації та авторизації,

встановити паролі для захисту певних областей сайту, обмежити доступ за IP-адресами або діапазонами IP тощо.

За допомогою `.htaccess` можна налаштувати сторінки обробки помилок для різних типів помилок HTTP. Наприклад, ви можете налаштувати користувацькі сторінки помилок 404 (сторінка не знайдена), 500 (внутрішня помилка сервера) та інших помилок.

Також файл `.htaccess` дає змогу керувати налаштуваннями кешування і стиснення на сервері. Ви можете налаштувати правила кешування для певних типів файлів, налаштувати стиснення даних для зменшення розміру переданих файлів тощо.

Для завантаження сайту на веб-хостинг спочатку його потрібно експортувати з Construct 2. Для цього натиснемо `Export Project` у меню програми. Оберемо шлях до папки з сайтом, назви підпапок для зображень та файлів, ступінь стискання зображень та виставимо мініфікацію JS скриптів, що зробить завантаження більш швидким

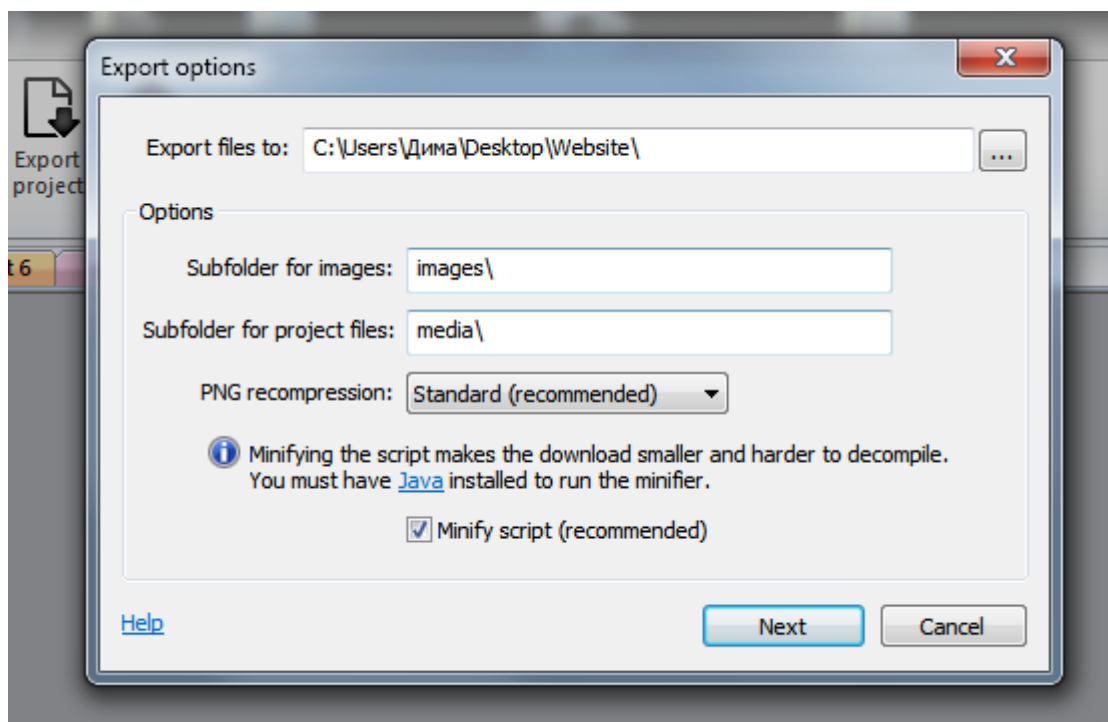


Рис. 3.34. Меню екпорту проекту у Construct 2

images	20.05.2023 0:03	Папка с файлами	
appmanifest.json	23.05.2023 12:42	JSON File	1 КБ
c2runtime.js	23.05.2023 12:42	Файл сценария JS...	185 КБ
data.js	23.05.2023 12:42	Файл сценария JS...	42 КБ
icon-16.png	23.05.2023 6:41	Рисунок PNG	1 КБ
icon-32.png	23.05.2023 6:41	Рисунок PNG	2 КБ
icon-114.png	23.05.2023 6:41	Рисунок PNG	7 КБ
icon-128.png	23.05.2023 6:41	Рисунок PNG	9 КБ
icon-256.png	23.05.2023 6:41	Рисунок PNG	25 КБ
index.html	25.05.2023 9:25	Файл "HTML"	5 КБ
indexfake.html	25.05.2023 9:29	Файл "HTML"	15 КБ
jquery-3.4.1.min.js	21.10.2019 12:47	Файл сценария JS...	87 КБ
loading-logo.png	23.05.2023 6:41	Рисунок PNG	25 КБ
offline.js	23.05.2023 12:42	Файл сценария JS...	1 КБ
offlineClient.js	29.09.2016 17:16	Файл сценария JS...	2 КБ
sw.js	15.05.2018 13:53	Файл сценария JS...	14 КБ

Рис. 3.21. Кінцевий вигляд експортованих файлів

Після завантаження усіх експортованих файлів створимо файли reg.php, auth.php, authsearch, authhtml.php, authsortbydate.php, authsortbynameAZ.php, authsortbynameZA.php, назви котрих ми раніше вписували в Construct 2.

<input type="checkbox"/>	Name ▼	Size	Date	Permissions
<input type="checkbox"/>	images		2023-05-19 21:16:00	drwxr-xr-x
<input type="checkbox"/>	.htaccess	0.2 kB	2023-04-15 16:58:00	-rw-r--r--
<input type="checkbox"/>	appmanifest.json	0.5 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	auth.php	2.2 kB	2023-05-23 09:02:00	-rw-r--r--
<input type="checkbox"/>	authhtml.php	1.2 kB	2023-05-25 13:24:00	-rw-r--r--
<input type="checkbox"/>	authsearch.php	2.8 kB	2023-05-25 12:45:00	-rw-r--r--
<input type="checkbox"/>	authsortbydate.php	2.1 kB	2023-05-25 08:57:00	-rw-r--r--
<input type="checkbox"/>	authsortbynameAZ.php	2.1 kB	2023-05-25 09:14:00	-rw-r--r--
<input type="checkbox"/>	authsortbynameZA.php	2.1 kB	2023-05-25 09:15:00	-rw-r--r--
<input type="checkbox"/>	c2runtime.js	180.6 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	data.js	38.8 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	icon-16.png	0.7 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	icon-32.png	1.4 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	icon-114.png	6.5 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	icon-128.png	8.5 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	icon-256.png	24.3 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	index.html	4.7 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	jquery-3.4.1.min.js	86.1 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	loading-logo.png	24.3 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	offline.js	0.6 kB	2023-05-22 13:38:00	-rw-r--r--
<input type="checkbox"/>	offlineClient.js	1.4 kB	2023-05-22 13:38:00	-rw-r--r--
<input checked="" type="checkbox"/>	reg.php	2.1 kB	2023-05-23 07:43:00	-rw-r--r--

Рис. 3.22. Кінцевий вигляд створених файлів у файловому менеджері хостингу

Після створення потрібних php файлів вставимо код, що відповідає кнопкам на веб-сайті:

- Файл reg.php - Додаток А;
- Файл auth.php – Додаток Б;
- Файл authsearch.php – Додаток В;
- Файл authhtml.php – Додаток Д;
- Файл authsortbynameAZ.php – Додаток Е;
- Файл authsortbynameZA.php – Додаток Ж;
- Файл authsortbydate.php – Додаток З.

Username	typechosen	submittime	sqare	phone number
Pohorielov Dmytro Olehovich	2	text	200	
Pohorielov Dmytro	4	text	112	
Lena	4	text	63	380753252234
Dmytro	5	Tue May 23 2023 12:36:32 GMT 0300 (Восточная Европа, летнее время)	980	380984944500
Dima	3	Tue May 23 2023 12:38:48 GMT 0300 (Восточная Европа, летнее время)	1426	380984944600
Dmytro Pohorielov	1	Sat May 27 2023 15:26:06 GMT 0300 (Восточная Европа, летнее время)	123	380984944600
name123	5	Sat May 27 2023 15:26:49 GMT 0300 (Восточная Европа, летнее время)	789	098989898

Рис. 3.23. Результат виконання файлу authhtml.php (html файл)

	A	B	C	D	E
1	username	typechosen	submittime	sqare	data
2	Pohorielov Dmytro Olehovich	2	text	200	
3	Pohorielov Dmytro	4	text	112	
4	Lena	4	text	63	380753252234
5	Dmytro	5	Tue May 23 2023 12:36:32 GMT 0300 (Восточная Европа, летнее время)	980	380984944500
6	Dima	3	Tue May 23 2023 12:38:48 GMT 0300 (Восточная Европа, летнее время)	1426	380984944600
7	Dmytro Pohorielov	1	Sat May 27 2023 15:26:06 GMT 0300 (Восточная Европа, летнее время)	123	380984944600
8	name123	5	Sat May 27 2023 15:26:49 GMT 0300 (Восточная Европа, летнее время)	789	098989898

Рис. 3.24. Результат виконання файлу auth.php (xml файл, що відкритий у Excel)

Результат виконання authsortbydate.php збігається з результатом виконання auth.php. Для перевірки роботи пошуку за ім'ям користувача або номером телефону введемо в поле пошуку текст:

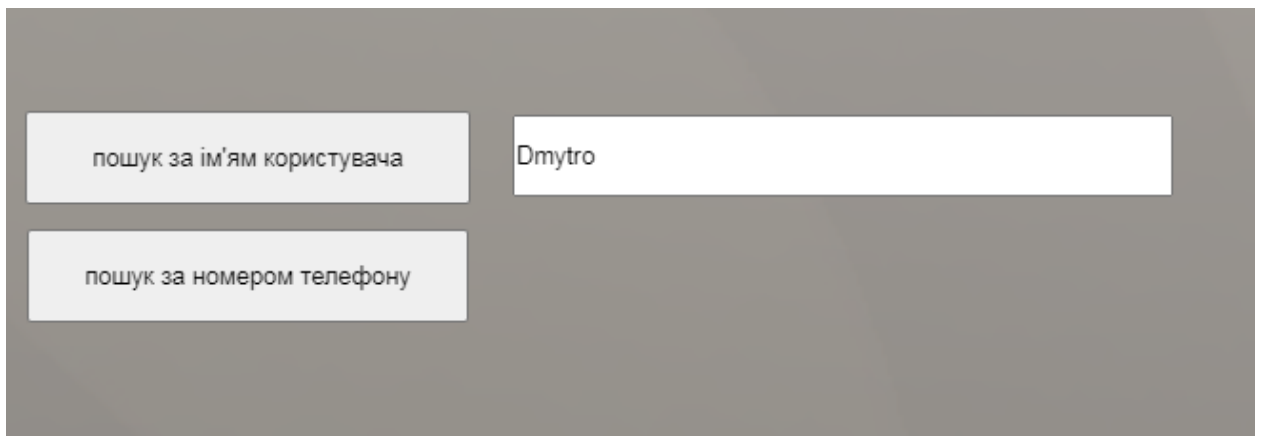


Рис. 3.25. Форма пошуку в базі даних за ім'ям або номером телефону

	A	B	C	D	E
1	username	typechosen	submittime	sqare	data
2	Pohorielov Dmytro Olehovich	2	text	200	
3	Pohorielov Dmytro	4	text	112	
4	Dmytro	5	Tue May 23 2023 12:36:32 GMT 0300 (Восточная Европа, летнее время)	980	380984944500
5	Dmytro Pohorielov	1	Sat May 27 2023 15:26:06 GMT 0300 (Восточная Европа, летнее время)	123	380984944600

Рис. 3.26. Результат виконання пошуку за ім'ям у вигляді XML-файлу в Excel

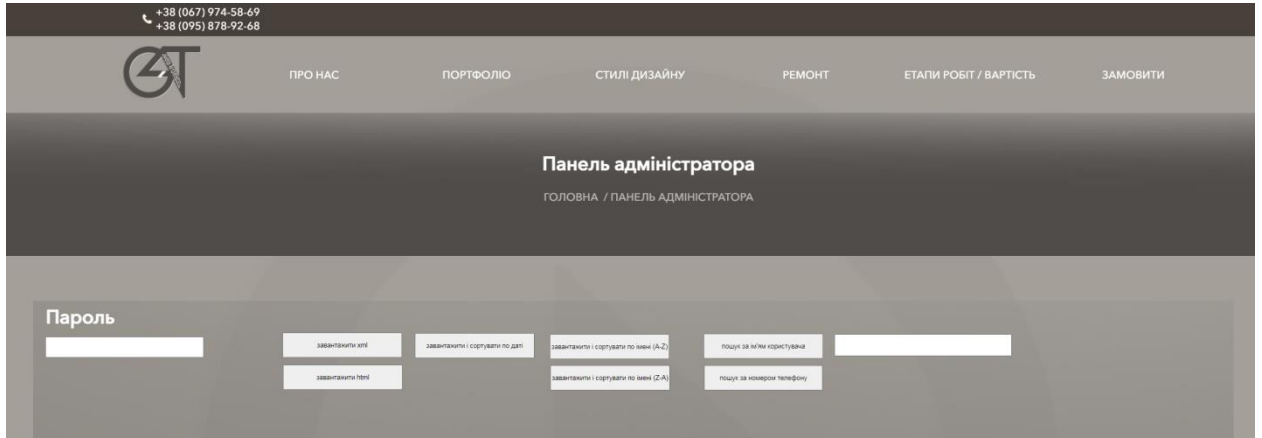


Рис. 3.27. Кінцевий вигляд панелі адміністратора

3.6. Висновок до Розділу 3

У третьому розділі зазначається послідовність дій, пов'язаних із розробкою дизайну та інформаційної системи. Спочатку було обрано дизайн веб-сайту, який відіграє важливу роль у створенні привабливого та зручного

користувачького інтерфейсу. Цей етап підкреслює значущість естетичної складової сайту для задоволення потреб клієнтів.

Далі були представлені скріншоти, супроводжувані коротким описом, які демонструють процес створення навігаційного меню сайту. Навігаційне меню відіграє важливу роль в організації контенту та забезпеченні зручності використання сайту. Описаний процес допомагає читачеві зрозуміти, як створити ефективно та інтуїтивно зрозуміле навігаційне меню.

Потім у розділі було додано текстову інформацію від замовника сайту. Це дає змогу персоналізувати контент сайту і надати інформацію, що відповідає вимогам і очікуванням замовника. Впровадження текстової інформації в веб-сайт є важливим кроком для його контекстуалізації та унікальності.

Наступні етапи включали створення бази даних і таблиці в phpMyAdmin. База даних є основою для зберігання та організації даних, необхідних для функціонування сайту. Створення таблиці в phpMyAdmin дає змогу визначити структуру і зв'язки між даними, що важливо для ефективного зберігання і вилучення інформації.

Нарешті, було представлено код на PHP для сортування, пошуку та відображення інформації у двох варіантах: у вигляді HTML-таблиці та XML-файлу. Це демонструє можливості програмування на PHP для опрацювання даних із бази даних та їхнього представлення в різних форматах. Обидва варіанти надають гнучкість і вибір залежно від вимог і переваг замовника.

Висновок третього розділу наголошує на важливих етапах розробки веб-сайту, починаючи з вибору дизайну та створення навігаційного меню, до додавання текстової інформації замовника, створення бази даних і таблиці, а також написання коду на PHP для обробки та відображення даних. Це забезпечує основу для розробки функціонального та змістовного веб-сайту.

ВИСНОВКИ

У ході цієї роботи було розроблено веб-сайт, що використовує HTML5, JavaScript і PHP, з інформаційно-пошуковою системою для надання архітектурних послуг. Сайт надає користувачеві функціональність пошуку і перегляду інформації про доступні архітектурні послуги, а також можливість зв'язатися з постачальниками послуг.

На початку роботи було визначено цілі та вимоги проекту, а також розроблено план і прототипи інтерфейсу користувача. Потім було обрано відповідні технології та платформи розроблення, включно з HTML5 для розмітки веб-сторінок, JavaScript для створення інтерактивності та PHP для обробки серверної логіки.

Особливу увагу було приділено розробці бази даних, яка зберігає інформацію про архітектурні послуги та замовника. Було створено схему бази даних і реалізовано з використанням відповідних технологій, таких як MySQL.

Важливою частиною процесу розробки було тестування і налагодження сайту. Були проведені тести функціональності, продуктивності, безпеки та сумісності з різними пристроями і браузерами. Виявлені помилки і проблеми були виправлені, щоб забезпечити коректну роботу сайту.

Однак, варто зазначити, що мобільна версія сайту все ще перебуває в розробці та потребує подальшої роботи. Наразі вона не повністю адаптована для мобільних пристроїв і може не забезпечувати оптимальний користувацький досвід на таких платформах. Надалі планується доопрацювання та оптимізація мобільної версії сайту, щоб розширити його доступність і поліпшити користувацький досвід на всіх пристроях.

Крім того, було проведено аналіз різних веб-хостингів для вибору найбільш відповідного провайдера для розміщення сайту. Були розглянуті параметри, такі як ціна, надійність, продуктивність і можливості

масштабування. На основі цього аналізу було обрано оптимальний варіант хостингу для розміщення сайту.

У підсумку, створений вебсайт з інформаційно-пошуковою системою надає зручний та інтуїтивно зрозумілий інтерфейс для користувачів, дає змогу легко знайти й отримати інформацію про доступні архітектурні послуги, а також забезпечує зв'язок із постачальниками послуг. Основні помилки були виправлені, але робота над розробкою мобільної версії сайту триває для поліпшення його адаптивності та користувацького досвіду на мобільних пристроях.

Загалом, цей проект являє собою успішне розроблення вебсайту з інформаційно-пошуковою системою, а також проведення аналізу та вибору оптимального хостингу для його розміщення.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Огляд Construct 2 SDK - Construct 2 Manual [Електронний ресурс]. - Режим доступу: www.scirra.com (дата звернення: 05.06.2023). - Назва з екрану.
2. Advances in dataflow programming languages [Електронний ресурс]. - Режим доступу: <https://dl.acm.org/doi/10.1145/1013208.1013209> (дата звернення 05.06.2023). - Назва з екрану
3. Фрімен Е. Вивчаємо HTML, XHTML и CSS / Е. Фрімен – Київ: Видавництво "Laurus", 2008. – 412 с.
4. Український веб-довідник [Електронний ресурс]. - Режим доступу: <https://css.in.ua/html/tags> (дата звернення: 05.06.2023). - Назва з екрану.
5. Construct 2 [Електронний ресурс]. - Режим доступу: <https://www.construct.net/> (дата звернення: 05.06.2023). - Назва з екрану.
6. Luke Welling, Laura Thomson. PHP & MySQL Web Development. - п'яте видання. / L. Welling, L. Thomson. - "Addison-Wesley", 2017. - С. 687.
7. Роберт Шелдон, Джофрей Мойє. MySQL 5: базовий курс = Beginning MySQL. / Р. Шелдон, Д. Мойє. - М.: "Діалектика", 2007. - 880 с.
8. Пітер Лабберс, Брайан Олберс, Френк Салім. HTML5 для професіоналів: потужні інструменти для розробки сучасних веб-додатків = Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. / П. Лабберс, Б. Олберс, Ф. Салім. - М.: "Вільямс", 2011. - 272 с.

ДОДАТКИ

Додаток А

reg.php

```
<?php
header('Access-Control-Allow-Origin: *');
$mysql_host = "localhost";
$mysql_database = "id20610525_guleta";
$mysql_user = "id20610525_infol";
$mysql_password = "*****";

$link = mysql_connect($mysql_host, $mysql_user, $mysql_password) or
die("error MySQL" );
mysql_select_db($mysql_database, $link) or die ('other');

if (isset($_POST['username'])) { $username = $_POST['username']; if
($username == "") { unset($username);} }
if (isset($_POST['submittime'])) { $submittime=$_POST['submittime']; if
($submittime == "") { unset($submittime);} }
if (isset($_POST['square'])) { $square=$_POST['square']; if ($square == "") {
unset($square);} }
if(isset($_POST['typechosen'])) $typechosen = $_POST['typechosen'];
if(isset($_POST['data'])) $data = $_POST['data'];
if (empty($username) or empty($submittime) or empty($square))
{
echo ("188");
}
$username = stripslashes($username);
$username = htmlspecialchars($username);
$typechosen = stripslashes($typechosen);
$typechosen = htmlspecialchars($typechosen);
$submittime = stripslashes($submittime);
$submittime = htmlspecialchars($submittime);
$square = stripslashes($square);
$square = htmlspecialchars($square);
$data = stripslashes($data);
$data = htmlspecialchars($data);
$username = trim($username);
$submittime = trim($submittime);
$square = trim($square);
$sql = mysql_query("SELECT username FROM `accounts` WHERE
`username`='". $username ."");
```

```
$usernameb = mysql_fetch_array($q1);
$usernamebd=$usernameb['username'];
if($usernamebd == $username){
    echo("5");
}
else
    $result3 = mysql_query ("INSERT INTO accounts
(username,typechosen,submittime,sqare,data)
VALUES('$username','$typechosen','$submittime','$sqare','$data)");
    if ($result3=="TRUE")
    {
        echo "6";
    }
else {
    echo "7";
}
?>
```

```
<?php
header('Access-Control-Allow-Origin: *');

$mysql_host = "localhost";
$mysql_database = "id20610525_guleta";
$mysql_user = "id20610525_infvol";
$mysql_password = "*****";

$link = mysqli_connect($mysql_host, $mysql_user, $mysql_password,
$mysql_database) or die("Ошибка при подключении MySQL");

if(isset($_POST['password2'])) $password2 = $_POST['password2'];

$password2 = stripslashes($password2);
$password2 = htmlspecialchars($password2);
if ($password2 == *****)
{
$sql = "SELECT * FROM accounts";
$result = mysqli_query($link, $sql);

header('Content-type: text/xml');
header('Cache-Control: no-cache');

$xmlWriter = xmlwriter_open_memory();
xmlwriter_start_document($xmlWriter, '1.0', 'UTF-8');
xmlwriter_start_element($xmlWriter, 'accounts');

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        xmlwriter_start_element($xmlWriter, 'account');
        xmlwriter_start_element($xmlWriter, 'username');
        xmlwriter_text($xmlWriter, $row["username"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'typechosen');
        xmlwriter_text($xmlWriter, $row["typechosen"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'submittime');
        xmlwriter_text($xmlWriter, $row["submittime"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'square');
        xmlwriter_text($xmlWriter, $row["square"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'data');
        xmlwriter_text($xmlWriter, $row["data"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_end_element($xmlWriter);
    }
}
```

```
}  
  
xmlwriter_end_element($xmlWriter);  
xmlwriter_end_document($xmlWriter);  
  
echo xmlwriter_output_memory($xmlWriter);  
  
mysqli_close($link);  
}  
?>
```

```
<?php
header('Access-Control-Allow-Origin: *');

$mysql_host = "localhost";
$mysql_database = "id20610525_guleta";
$mysql_user = "id20610525_infvol";
$mysql_password = "*****";
$link = mysqli_connect($mysql_host, $mysql_user, $mysql_password,
$mysql_database) or die("Ошибка при подключении MySQL");
if(isset($_POST['password2'])) $password2 = $_POST['password2'];
$password2 = stripslashes($password2);
$password2 = htmlspecialchars($password2);
if ($password2 == *****) {
    if (isset($_POST['username']) && !empty($_POST['username'])) {
        $searchUsername = $_POST['username'];
        $sql = "SELECT * FROM accounts WHERE username LIKE
'%$searchUsername%'";
    } elseif (isset($_POST['data']) && !empty($_POST['data'])) {
        $searchData = $_POST['data'];
        $sql = "SELECT * FROM accounts WHERE data LIKE '%$searchData%'";
    } else {
        $sql = "SELECT * FROM accounts";
    }
}
$result = mysqli_query($link, $sql);
header('Content-type: text/xml');
header('Cache-Control: no-cache');
$xmlWriter = xmlwriter_open_memory();
xmlwriter_start_document($xmlWriter, '1.0', 'UTF-8');
xmlwriter_start_element($xmlWriter, 'accounts');
if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        xmlwriter_start_element($xmlWriter, 'account');
        xmlwriter_start_element($xmlWriter, 'username');
        xmlwriter_text($xmlWriter, $row["username"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'typechosen');
        xmlwriter_text($xmlWriter, $row["typechosen"]);
        xmlwriter_end_element($xmlWriter);

        xmlwriter_start_element($xmlWriter, 'submittime');
        xmlwriter_text($xmlWriter, $row["submittime"]);
        xmlwriter_end_element($xmlWriter);
    }
}
```

```
xmlwriter_start_element($xmlWriter, 'square');  
xmlwriter_text($xmlWriter, $row["square"]);  
xmlwriter_end_element($xmlWriter);  
  
xmlwriter_start_element($xmlWriter, 'data');  
xmlwriter_text($xmlWriter, $row["data"]);  
xmlwriter_end_element($xmlWriter);  
  
xmlwriter_end_element($xmlWriter);  
}  
}  
xmlwriter_end_element($xmlWriter);  
xmlwriter_end_document($xmlWriter);  
echo xmlwriter_output_memory($xmlWriter);  
mysqli_close($link);  
}  
?>
```



```
<?php
header('Access-Control-Allow-Origin: *');
$mysql_host = "localhost";
$mysql_database = "id20610525_guleta";
$mysql_user = "id20610525_infvol";
$mysql_password = "*****";
$link = mysqli_connect($mysql_host, $mysql_user, $mysql_password,
$mysql_database) or die("Ошибка при подключении MySQL");
if(isset($_POST['password2'])) $password2 = $_POST['password2'];
$password2 = stripslashes($password2);
$password2 = htmlspecialchars($password2);
if ($password2 == *****)
{
$sql = "SELECT * FROM accounts";
$result = mysqli_query($link, $sql);
echo "<table>";
echo
"<tr><th>Username</th><th>typechosen</th><th>submittime</th><th>sqare</th>
<th>phone number</th></tr>";
if (mysqli_num_rows($result) > 0) {
while ($row = mysqli_fetch_assoc($result)) {
echo "<tr>";
echo "<td>" . $row["username"] . "</td>";
echo "<td>" . $row["typechosen"] . "</td>";
echo "<td>" . $row["submittime"] . "</td>";
echo "<td>" . $row["sqare"] . "</td>";
echo "<td>" . $row["data"] . "</td>";
echo "</tr>";
}
}
echo "</table>";
mysqli_close($link);
}
?>
```

```
<?php
header('Access-Control-Allow-Origin: *');
$mysql_host = "localhost";
$mysql_database = "id20610525_guleta";
$mysql_user = "id20610525_infvol";
$mysql_password = "*****";
$link = mysqli_connect($mysql_host, $mysql_user, $mysql_password,
$mysql_database) or die("Ошибка при подключении MySQL");
if(isset($_POST['password'])) $password = $_POST['password'];
if ($password == *****)
{
$sql = "SELECT * FROM accounts ORDER BY username ASC";
$result = mysqli_query($link, $sql);
header('Content-type: text/xml');
header('Cache-Control: no-cache');
$xmlWriter = xmlwriter_open_memory();
xmlwriter_start_document($xmlWriter, '1.0', 'UTF-8');
xmlwriter_start_element($xmlWriter, 'accounts');
if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        xmlwriter_start_element($xmlWriter, 'account');
        xmlwriter_start_element($xmlWriter, 'username');
        xmlwriter_text($xmlWriter, $row["username"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'typechosen');
        xmlwriter_text($xmlWriter, $row["typechosen"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'submittime1');
        xmlwriter_text($xmlWriter, $row["submittime"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'sqare');
        xmlwriter_text($xmlWriter, $row["sqare"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'data');
        xmlwriter_text($xmlWriter, $row["data"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_end_element($xmlWriter);
    }
}
xmlwriter_end_element($xmlWriter);
xmlwriter_end_document($xmlWriter);
echo xmlwriter_output_memory($xmlWriter);
```

```
mysqli_close($link);  
?>
```

```
<?php
header('Access-Control-Allow-Origin: *');
$mysql_host = "localhost";
$mysql_database = "id20610525_guleta";
$mysql_user = "id20610525_infvol";
$mysql_password = "*****";
$link = mysqli_connect($mysql_host, $mysql_user, $mysql_password,
$mysql_database) or die("Ошибка при подключении MySQL");
if(isset($_POST['password'])) $password = $_POST['password'];
if ($password == *****)
{
$sql = "SELECT * FROM accounts ORDER BY username DESC";
$result = mysqli_query($link, $sql);
header('Content-type: text/xml');
header('Cache-Control: no-cache');
$xmlWriter = xmlwriter_open_memory();
xmlwriter_start_document($xmlWriter, '1.0', 'UTF-8');
xmlwriter_start_element($xmlWriter, 'accounts');
if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        xmlwriter_start_element($xmlWriter, 'account');
        xmlwriter_start_element($xmlWriter, 'username');
        xmlwriter_text($xmlWriter, $row["username"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'typechosen');
        xmlwriter_text($xmlWriter, $row["typechosen"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'submittime1');
        xmlwriter_text($xmlWriter, $row["submittime"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'square');
        xmlwriter_text($xmlWriter, $row["square"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_start_element($xmlWriter, 'data');
        xmlwriter_text($xmlWriter, $row["data"]);
        xmlwriter_end_element($xmlWriter);
        xmlwriter_end_element($xmlWriter);
    }
}
xmlwriter_end_element($xmlWriter);
xmlwriter_end_document($xmlWriter);
echo xmlwriter_output_memory($xmlWriter);
```

```
mysqli_close($link);  
?>
```

```
<?php
header('Access-Control-Allow-Origin: *');
$mysql_host = "localhost";
$mysql_database = "id20610525_guleta";
$mysql_user = "id20610525_infvol";
$mysql_password = "*****";
$link = mysqli_connect($mysql_host, $mysql_user, $mysql_password,
$mysql_database) or die("Ошибка при подключении MySQL");
if(isset($_POST['password'])) $password = $_POST['password'];
if ($password == *****)
{
$sql = "SELECT * FROM accounts ORDER BY STR_TO_DATE(submittime,
'a %b %d %Y %H:%i:%s GMT %r') DESC";
$result = mysqli_query($link, $sql);
header('Content-type: text/xml');
header('Cache-Control: no-cache');
$xmlWriter = xmlwriter_open_memory();
xmlwriter_start_document($xmlWriter, '1.0', 'UTF-8');
xmlwriter_start_element($xmlWriter, 'accounts');
if (mysqli_num_rows($result) > 0) {
while ($row = mysqli_fetch_assoc($result)) {
xmlwriter_start_element($xmlWriter, 'account');
xmlwriter_start_element($xmlWriter, 'username');
xmlwriter_text($xmlWriter, $row["username"]);
xmlwriter_end_element($xmlWriter);
xmlwriter_start_element($xmlWriter, 'typechosen');
xmlwriter_text($xmlWriter, $row["typechosen"]);
xmlwriter_end_element($xmlWriter);
xmlwriter_start_element($xmlWriter, 'submittime');
xmlwriter_text($xmlWriter, $row["submittime"]);
xmlwriter_end_element($xmlWriter);
xmlwriter_start_element($xmlWriter, 'square');
xmlwriter_text($xmlWriter, $row["square"]);
xmlwriter_end_element($xmlWriter);
xmlwriter_start_element($xmlWriter, 'data');
xmlwriter_text($xmlWriter, $row["data"]);
xmlwriter_end_element($xmlWriter);
xmlwriter_end_element($xmlWriter);
}
}
}
```

```
xmlwriter_end_element($xmlWriter);  
xmlwriter_end_document($xmlWriter);  
echo xmlwriter_output_memory($xmlWriter);  
mysqli_close($link);  
?>
```

```
<!DOCTYPE html>
<html><head>
  <title>Website</title>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

  <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0, user-scalable=no, minimal-ui">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <meta name="HandheldFriendly" content="true">

<style type="text/css">
  * {
    padding: 0;
    margin: 0;
  }
  html, body {
    background: #000;
    color: #fff;
    overflow: hidden;
  }
  canvas {
    touch-action-delay: none;
    touch-action: none;
    -ms-touch-action: none;
  }
  #inspect-outline {
    position: absolute;
    border: 2px dotted red;
    overflow: hidden;
    font-size: 8pt;
    font-family: Sans serif;
    color: #444;
    pointer-events: none;
    -webkit-animation: flash 2s linear 0s infinite;
    -moz-animation: flash 2s linear 0s infinite;
    -o-animation: flash 2s linear 0s infinite;
    animation: flash 2s linear 0s infinite;
    background-image: url(debugger-inspect.png);
  }

  @-webkit-keyframes flash {
```



```

    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
}
@-moz-keyframes flash {
    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
}
@-o-keyframes flash {
    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
}

@keyframes flash {
    0%, 75%, 100% {opacity: 1;}
    87% {opacity: 0.25;}
}
</style>
<script type="text/javascript" src="jquery-3.4.1.min.js"></script>
<script type="text/javascript" src="common_prelude.js"></script>
<script type="text/javascript" src="preview_prelude.js"></script>
<script type="text/javascript" src="shaders.js"></script>
<script type="text/javascript" src="glwrap.js"></script>
<script src="AJAX_common.js"></script>
<script src="AJAX_plugin.js"></script>
<script src="Arr_common.js"></script>
<script src="Arr_plugin.js"></script>
<script src="Browser_common.js"></script>
<script src="Browser_plugin.js"></script>
<script src="Button_common.js"></script>
<script src="Button_plugin.js"></script>
<script src="Function_common.js"></script>
<script src="Function_plugin.js"></script>
<script src="Keyboard_common.js"></script>
<script src="Keyboard_plugin.js"></script>
<script src="Mouse_common.js"></script>
<script src="Mouse_plugin.js"></script>
<script src="Sprite_common.js"></script>
<script src="Sprite_plugin.js"></script>
<script src="Text_common.js"></script>
<script src="Text_plugin.js"></script>
<script src="TextBox_common.js"></script>
<script src="TextBox_plugin.js"></script>
<script src="XML_common.js"></script>

```

```

<script src="XML_plugin.js"></script>
<script src="EightDir_common.js"></script>
<script src="EightDir_behavior.js"></script>
<script src="Fade_common.js"></script>
<script src="Fade_behavior.js"></script>
<script src="scrollto_common.js"></script>
<script src="scrollto_behavior.js"></script>
<script type="text/javascript" src="objrefs.js"></script>
<script type="text/javascript" src="preview.js"></script>
<script type="text/javascript" src="layout.js"></script>
<script type="text/javascript" src="eveng.js"></script>
<script type="text/javascript" src="expressions.js"></script>
<script type="text/javascript" src="system.js"></script>
<script type="text/javascript" src="commonace.js"></script>
<script type="text/javascript">
  // Start the project running on window load
  jQuery(document).ready(function ()
  {
    cr.createRuntime("c2canvas");
  });
  // Pause and resume on page becoming visible/invisible
  function onVisibilityChanged() {
    if (document.hidden || document.mozHidden || document.webkitHidden ||
document.msHidden)
      cr_setSuspended(true);
    else
      cr_setSuspended(false);
  };
  document.addEventListener("visibilitychange", onVisibilityChanged, false);
  document.addEventListener("mozvisibilitychange", onVisibilityChanged, false);
  document.addEventListener("webkitvisibilitychange", onVisibilityChanged,
false);
  document.addEventListener("msvisibilitychange", onVisibilityChanged, false);
</script>
</head>
<body style="cursor: auto;">
<div id="fb-root"></div>
<div id="c2canvasdiv" style="width: 494px; height: 1329px; margin-left: 0px;
margin-top: 0px;">
<canvas id="c2canvas" width="494" height="1329" style="width: 494px;
height: 1329px;">
<div id="canvasNotSupportedBox">

```

```

<h1>This browser does not appear to support HTML5. Try upgrading your
browser to the latest version. <a href="http://www.whatbrowser.org">What is a
browser?</a>
  <br><br><a href="http://www.microsoft.com/windows/internet-
explorer/default.aspx">Microsoft Internet Explorer</a><br>
  <a href="http://www.mozilla.com/firefox/">Mozilla Firefox</a><br>
  <a href="http://www.google.com/chrome/">Google Chrome</a><br>
  <a href="http://www.apple.com/safari/download/">Apple Safari</a>
</h1>
</div>
</canvas>
<input type="button" id="" value="завантажити xml" title="" style="position:
absolute; top: 649px; left: 581px; width: 234px; height: 49px; font-size: 0.8em;
display: none;">
  <input type="password" id="" autocomplete="off" placeholder="" title=""
spellcheck="false" style="position: absolute; top: 657px; left: 117px; width: 306px;
height: 37px; font-size: 0.8em; display: none;">
  <input type="button" id="" value="copy" title="" style="display: none;">
  <input type="text" id="" autocomplete="off" placeholder="" title=""
spellcheck="false" style="display: none;">
  <input type="button" id="" value="завантажити і сортувати по даті"
title="" style="position: absolute; top: 651px; left: 1px; width: 56px; height: 46px;
font-size: 0.722917em; display: none;">
  <input type="button" id="" value="завантажити і сортувати по імені (A-Z)"
title="" style="position: absolute; top: 653px; left: 1px; width: 208px; height:
46px; font-size: 0.722917em;">
  <input type="button" id="" value="завантажити і сортувати по імені (Z-A)"
title="" style="position: absolute; top: 709px; left: 1px; width: 209px; height:
46px; font-size: 0.722917em;">
  <input type="button" id="" value="пошук за ім'ям користувача" title=""
style="position: absolute; top: 652px; left: 271px; width: 216px; height: 46px;
font-size: 0.722917em;">
  <input type="text" id="" autocomplete="off" placeholder="" title=""
spellcheck="false" style="position: absolute; top: 652px; left: 1661px; width:
343px; height: 39px; font-size: 0.8em; display: none;">
  <input type="button" id="" value="пошук за номером телефону" title=""
style="position: absolute; top: 709px; left: 272px; width: 214px; height: 46px;
font-size: 0.722917em;">
  <input type="button" id="" value="завантажити html" title=""
style="position: absolute; top: 712px; left: 582px; width: 234px; height: 49px;
font-size: 0.8em; display: none;"></div>
</body>
</html>

```

```
function saveXML(xmlString, fileName) {
  var blob = new Blob([xmlString], { type: 'text/xml' });
  if (navigator.msSaveOrOpenBlob) {
    navigator.msSaveOrOpenBlob(blob, fileName);
  } else {
    var url = URL.createObjectURL(blob);
    var link = document.createElement('a');
    link.href = url;
    link.download = fileName;
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
    URL.revokeObjectURL(url);
  }
}
var xmlString = "&xml&";
var fileName = 'data.html';
saveXML(xmlString, fileName);
```