

Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота бакалавра

на тему : «Розроблення інформаційного сайту з продажу малюнків»

Виконав: студент групи ПП320-1

Спеціальність 121 «Інженерія програмного
забезпечення»

Ібрагімов Саядулли Мехманович

(прізвище та ініціали)

Керівник д.т.н., доц. Яковенко В.О.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент Університет митної справи та
фінансів

(місце роботи)

Професор кафедри кібербезпеки та
інформаційних технологій

(посада)

д.е.н., професор Паршина О.А.

(науковий ступінь, вчене звання, прізвище та ініціали)

АНОТАЦІЯ

Ібрагімов С. М. Розроблення інформаційного сайту з продажу малюнків.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». – Університет митної справи та фінансів, Дніпро, 2024.

Ця робота присвячена розробці інформаційного сайту з продажу малюнків, що включає аналіз та вибір програмних засобів для його створення, розробку бази даних для зберігання інформації про малюнки, створення інтерфейсу вебзастосунку, розробку стилю та мережевого інтерфейсу. У розділі аналізу методів розробки розглянуті різні технології, мови програмування та інструменти, що можуть бути використані для реалізації проекту. Особлива увага приділяється вибору програмних засобів та їхньому практичному застосуванню для досягнення мети проекту. Крім того, розглядаються аспекти безпеки та продуктивності, а також можливості масштабування сайту для різних методів розробки. Ця робота важлива для творців та покупців малюнків, які шукають зручний та ефективний спосіб обміну творчістю та придбання унікальних мистецьких робіт. Її результати можуть бути використані для розробки потужного та функціонального інформаційного ресурсу для продажу малюнків.

Інформаційний портал створюється для митців, щоб вони могли представити свої твори для продажу, а також для покупців, які шукають оригінальні картини. Робота включає вибір програмних засобів, від PHP до Bootstrap, з врахуванням функціональності, безпеки та ефективності. Досліджується розробка бази даних, реєстрація користувачів, інтерфейс та зручний вебдизайн. Висновки надають практичну цінність для розробки ефективного майданчика з продажу малюнків, підкреслюючи необхідність збалансованого підходу до технологічних рішень та дизайну.

Ключові слова: інформаційний сайт, мова програмування PHP, JavaScript, OpenServer, MySQL, AJAX, Bootstrap.

ABSTRACT

Ibragimov S.M. Development of an information website for the sale of drawings.

Qualification of work to obtain a bachelor's degree in specialty 121 «Software Engineering» – University of Customs and Finance, Dnipro, 2024.

This work is devoted to the development of an information site for the sale of drawings, which includes the analysis and selection of software tools for its creation, the development of a database for storing information about drawings, the creation of a web application interface, the development of a style and a network interface. In the section of the analysis of development methods, various technologies, programming languages and tools that can be used for the implementation of the project are considered. Special attention is paid to the selection of software tools and their practical application to achieve the goal of the project. In addition, security and performance aspects are considered, as well as the scalability of the site for different development methods. This work is important for creators and buyers of art who are looking for a convenient and efficient way to share their creativity and purchase unique works of art. Its results can be used to develop a powerful and functional information resource for selling drawings.

An information portal is created for artists to present their works for sale, as well as for buyers looking for original paintings. The work involves choosing software tools, from PHP to Bootstrap, with functionality, security and efficiency in mind. Database development, user registration, interface and user-friendly web design are explored. The findings provide practical value for the development of an effective platform for the sale of drawings, emphasizing the need for a balanced approach to technological solutions and design.

Keywords: information site, PHP programming language, JavaScript, OpenServer, MySQL, AJAX, Bootstrap.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ.....	8
1.1 Аналіз існуючих сайтів по продажу малюнків	8
1.2 Аналіз методів розробки інформаційних сайтів.....	15
1.2 Висновки до першого розділу. Постановка завдань дослідження.....	23
РОЗДІЛ 2 АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ	24
2.1 Вибір програмних засобів для реалізації проекту	24
2.2 PHP	24
2.3 JavaScript	28
2.4 OpenServer.....	30
2.5 MySQL.....	32
2.6 AJAX.....	33
2.7 Bootstrap	36
2.8 Висновки до другого розділу.....	37
РОЗДІЛ 3 РОЗРОБКА	39
3.1 Проектування бази даних.....	39
3.2 Види профілей користувачів.....	44
3.3 Реєстрація користувачів. Авторизація.	45
3.4 Користування контентом.....	47
3.4.1 Користування сервісом перегляду інформації	47
3.4.2 Користування сервісом оформлення замовлення	50
3.5 Різниця між профілями	51
3.6 Висновки до третього розділу	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТОК А.....	60
ДОДАТОК Б	77

ВСТУП

Мета роботи по розробці сайту для продажу малюнків полягає в створенні інтернет-платформи, що надає користувачам можливість перегляду, вибору та придбання малюнків онлайн. Головною метою є розроблення зручного та ефективного середовища для покупців, де вони можуть швидко знайти та придбати потрібні малюнки, а також для продавців, щоб вони могли представляти свої твори та здійснювати продажі через цю платформу.

Створення сайту для продажу малюнків має значну актуальність в сучасному світі з огляду на кілька ключових аспектів.

Популярність онлайн-торгівлі мистецтвом. За останні роки спостерігається зростання популярності онлайн-торгівлі мистецтвом, включаючи малюнки. Користувачі все частіше шукають можливості купувати твори мистецтва безпосередньо через Інтернет, що робить проект по створенню такого сайту актуальним.

Доступність та різноманітність контенту. Інтернет надає можливість легкого доступу до різноманітних малюнків з усього світу. Створення платформи для їх продажу дозволить об'єднати як художників, так і покупців у велику спільноту.

Зручність та ефективність. Онлайн-магазин малюнків дозволить покупцям зручно та ефективно шукати, переглядати та придбавати малюнки, використовуючи різноманітні фільтри та опції пошуку.

Можливість для художників. Для художників такий сайт відкриває можливості для просування своїх творів та збільшення їхньої аудиторії. Вони можуть легко представляти свої роботи на платформі та залучати нових клієнтів.

Отже, розробка сайту для продажу малюнків є актуальним завданням, яке відповідає потребам сучасного ринку мистецтва та відкриває нові можливості для художників та покупців.

Створення сайту для продажу малюнків має велике значення для

предметної області мистецтва та онлайн-торгівлі. Декілька ключових аспектів:

- створення такого сайту дозволить розвивати художню галерею в онлайн-середовищі, надаючи художникам можливість представляти свої твори широкій аудиторії;

- онлайн-платформа для продажу малюнків робить мистецтво доступнішим для широкого кола людей, незалежно від їх місця проживання або фінансових можливостей;

- створення такого сайту надасть підтримку художникам, дозволяючи їм заробляти на своїх творах та залучати нових клієнтів;

- онлайн-платформа дозволить привернути аудиторію з різних країн та культур, розширюючи можливості для художників та збільшуючи ринок мистецтва.

Отже, розв'язання цього завдання відіграє важливу роль у розвитку сучасної мистецької сфери, сприяючи її розширенню, доступності та підтримці художників.

Сучасний рівень розв'язання завдання створення сайту для продажу малюнків відображає використання передових технологій та підходів у веброботці.

Адаптивний дизайн. Сучасні сайти для продажу малюнків мають адаптивний дизайн, що забезпечує їх коректне відображення на різних пристроях, включаючи комп'ютери, планшети та смартфони.

Інтерактивність. Застосування JavaScript та AJAX дозволяє створювати інтерактивні елементи на сайті, такі як живі фільтри, швидка завантаження контенту без перезавантаження сторінки тощо.

Безпека даних. Сучасні сайти використовують захист від SQL-ін'єкцій, перевірку на введення користувача, шифрування паролів та інші заходи для забезпечення безпеки даних користувачів.

Управління контентом. Використання систем управління контентом (CMS) або власних адміністративних панелей дозволяє легко додавати, редагувати та видаляти малюнки на сайті без необхідності втручання в роботу

програміста.

Швидкість завантаження. Використання кешування, стиснення ресурсів та інші оптимізації допомагають забезпечити швидке завантаження сторінок сайту.

SEO-оптимізація. Модульна структура сайту, дружні URL-адреси, метатеги та інші SEO-техніки сприяють підвищенню видимості сайту у пошукових системах.

Узагальнено, сучасні рішення для сайтів з продажу малюнків поєднують в собі передові технології, безпеку та зручність в управлінні контентом, що робить їх привабливими як для художників, так і для покупців мистецьких творів.

Основні технічні характеристики продукту, що розробляється (вебсайту для продажу малюнків), можуть включати:

- сайт має бути адаптивним до різних пристроїв, забезпечуючи зручний перегляд на комп'ютерах, планшетах та смартфонах;
- розробка адміністративної панелі або інтеграція готової CMS для додавання, редагування та видалення малюнків, керування замовленнями та іншим контентом;
- система реєстрації та авторизації для користувачів, можливо з використанням соціальних медіа акаунтів;
- можливість пошуку малюнків за різними критеріями (тегами, категоріями, художниками тощо) і фільтрація результатів;
- оптимізація завантаження сторінок, захист від атак, шифрування даних та інші заходи для забезпечення швидкої та безпечної роботи сайту.

Ці технічні характеристики спрямовані на створення функціонального, зручного та безпечного вебсайту для продажу малюнків, який задовольнятиме потреби як художників, так і покупців мистецьких творів.

Кваліфікаційна робота складається зі вступу, 3-х розділів, висновків, використаних джерел з 15 найменувань, 2 додатків.

Обсяг роботи 90 сторінок, містить 22 рисунки.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Аналіз існуючих сайтів по продажу малюнків

Ось кілька прикладів відомих українських та зарубіжних вебсайтів, спеціалізованих на продажу малюнків:

1) Saatchi Art (рис. 1.1)

Saatchi Art – це онлайн-галерея, яка спеціалізується на продажі сучасного мистецтва від художників з усього світу. Можливості, які надає Saatchi Art:

- широкий вибір різноманітних творів мистецтва, включаючи картини, малюнки, скульптури, фотографії, твори різних стилів, технік і напрямків, що відповідають різним уподобанням і інтересам;
- відкритий для талановитих художників з усього світу, що дозволяє відкривати нові таланти та різноманітність мистецьких виразів;
- персоналізовані рекомендації на основі різних інтересів і попередніх переглядів, що допомагає знайти твори мистецтва, які зацікавлять;
- консультації від професійних мистецьких радників, щоб допомогти вам зробити правильний вибір;
- взаємодію безпосередньо з художниками через платформу, можливість задавати їм питання про їхні роботи, запитувати про їх творчий процес або навіть замовляти індивідуальні роботи;
- повний сервіс покупки, включаючи доставку та упаковку мистецьких робіт, щоб зробити процес покупки максимально зручним для клієнтів.

Saatchi Art – це платформа, яка пропонує широкий вибір сучасного мистецтва, сприяє відкритості для художників з усього світу і забезпечує повний сервіс для покупців.



Рисунок 1.1 – Інтерфейс Saatchi Art

2) Etsy (рис. 1.2)

Etsy – це онлайн-торгова платформа, яка спеціалізується на ручних роботах, виробах ремісників, унікальних товарах та виробах мистецтва. Ось кілька ключових можливостей Etsy:

- широкий вибір товарів, включаючи ручні роботи, унікальні вироби ремісників, мистецтво, вироби з натуральних матеріалів, прикраси, предмети декору та багато іншого;

- багато товарів на Etsy є унікальними та неповторними, оскільки вони створюються ручним способом або в обмежених кількостях ремісниками та художниками;

- надає можливість місцевим та незалежним ремісникам та художникам продавати свої вироби онлайн, що сприяє підтримці місцевих спільнот та культурних традицій;

- використовує алгоритми рекомендацій, щоб пропонувати користувачам товари, які відповідають їхнім інтересам та уподобанням;

- створює спільноту ремісників та покупців, де користувачі можуть обмінюватися досвідом, порадами та ідеями, а також підтримувати та спілкуватися з творцями;

– надає можливість купувати товари з усього світу, а також сприяє глобальній доставці, щоб покупці могли отримати доступ до унікальних товарів навіть з інших країн.

Загалом, Etsy – це платформа, яка збирає разом ремісників, художників та покупців з усього світу, сприяючи обміну унікальними та ручними творами мистецтва та ремесла.

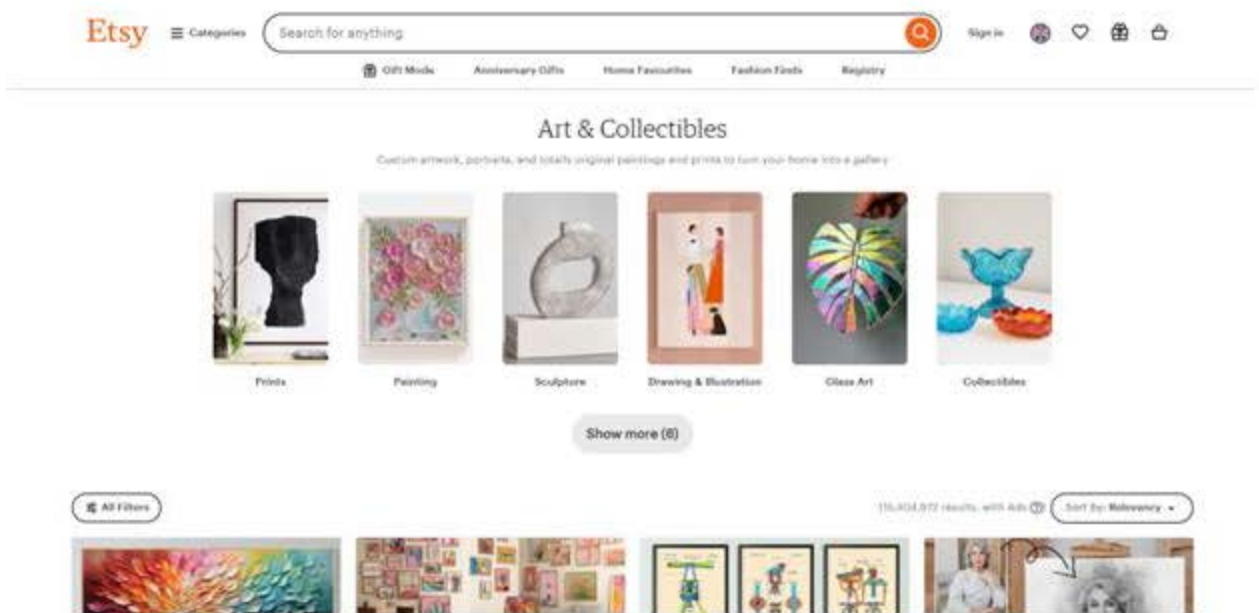


Рисунок 1.2 – Інтерфейс Etsy

3) DeviantArt (рис. 1.3)

DeviantArt – це онлайн-спільнота для художників та фотографів, де вони можуть виставляти свої твори для перегляду, обговорення та продажу. Ось кілька ключових можливостей DeviantArt:

– користувачі можуть публікувати свої малюнки, фотографії, графічні дизайни та інші твори мистецтва для показу спільноті та отримання відгуків;

– створює сприятливу спільноту, де користувачі можуть обговорювати роботи один з одним, ділитися порадами та враженнями, а також взаємодіяти з іншими художниками;

– має можливість продажу творів, що дозволяє художникам продавати свої роботи спільноті або приватним покупцям;

- користувачі можуть оцінювати та залишати відгуки про роботи інших учасників спільноти, що допомагає збирати враження про їхній талант і стиль;
- надає можливості для організації та управління особистим портфоліо, де художники можуть показати свої найкращі роботи та демонструвати свій творчий шлях;
- проводить різноманітні конкурси, виставки та інші заходи, в яких користувачі можуть брати участь та вигравати визнання та нагороди.

DeviantArt – це місце, де художники можуть публікувати, спілкуватися, продавати та отримувати відгуки на свої твори, а також долучатися до широкої та різноманітної мистецької спільноти.

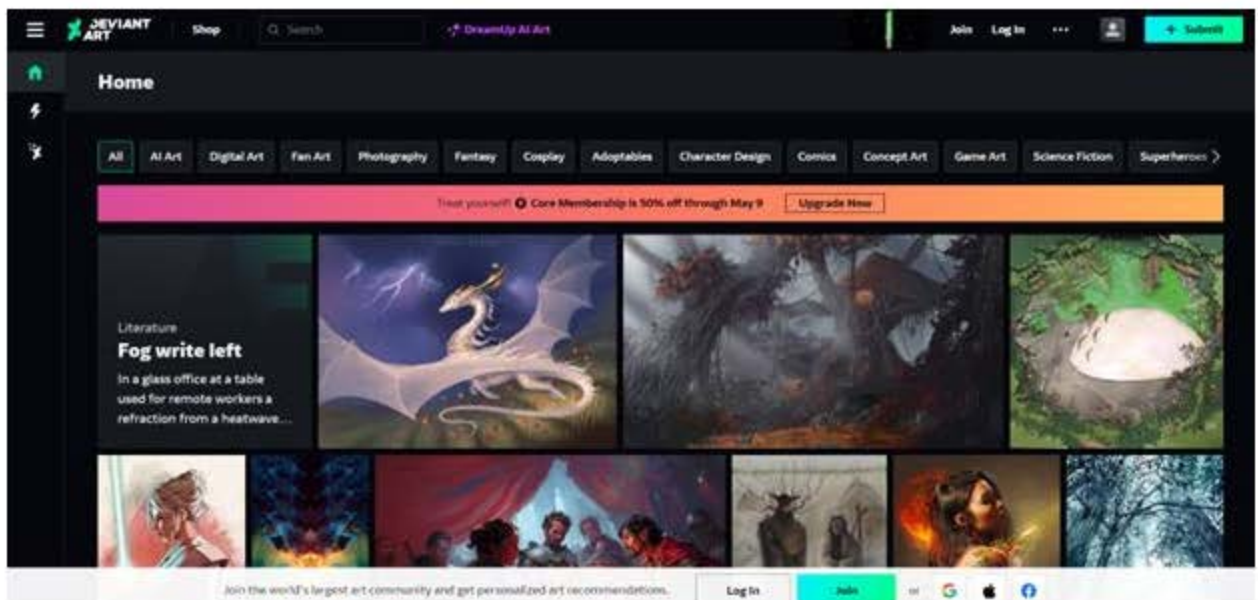


Рисунок 1.3 – Інтерфейс DeviantArt

4) Society6 (рис. 1.4)

Порівняльний огляд можливостей Society6 з Etsy та DeviantArt.

Як і в інших платформах, Society6 дозволяє художникам продавати свої твори, але вони пропонують різні типи продукції, такі як футболки, подушки, постери та інші товари, які можна надрукувати з їхніми дизайнами. Це відрізняється від Etsy, де переважно продаються ручні роботи та вироби мистецтва, і DeviantArt, де продаж робіт здійснюється як цифрові копії або

фізичні екземпляри.

Society6 відкритий для нових та талановитих художників, але вони спеціалізуються на дизайні для друку на різних продуктах, що може зацікавити художників, які шукають альтернативні способи монетизації своїх творів.

Society6 надає покупцям можливість придбати мистецькі твори у вигляді різних предметів, що може бути зручно для тих, хто шукає унікальні та оригінальні товари для дому чи подарунки. У порівнянні з DeviantArt, де продаж обмежується в основному цифровими копіями або фізичними друками, а Society6 пропонує більшу різноманітність продукції;

Хоча Society6 має певні спільнотні функції, такі як коментарі та рейтинги, вони можуть бути менш вираженими порівняно з DeviantArt, де спілкування з іншими художниками та отримання відгуків є важливою частиною платформи.

Отже, Society6 відрізняється від Etsy та DeviantArt шляхом спеціалізації на друку мистецьких творів на різних товарах, що може відкривати нові можливості для художників у монетизації їхніх дизайнів.

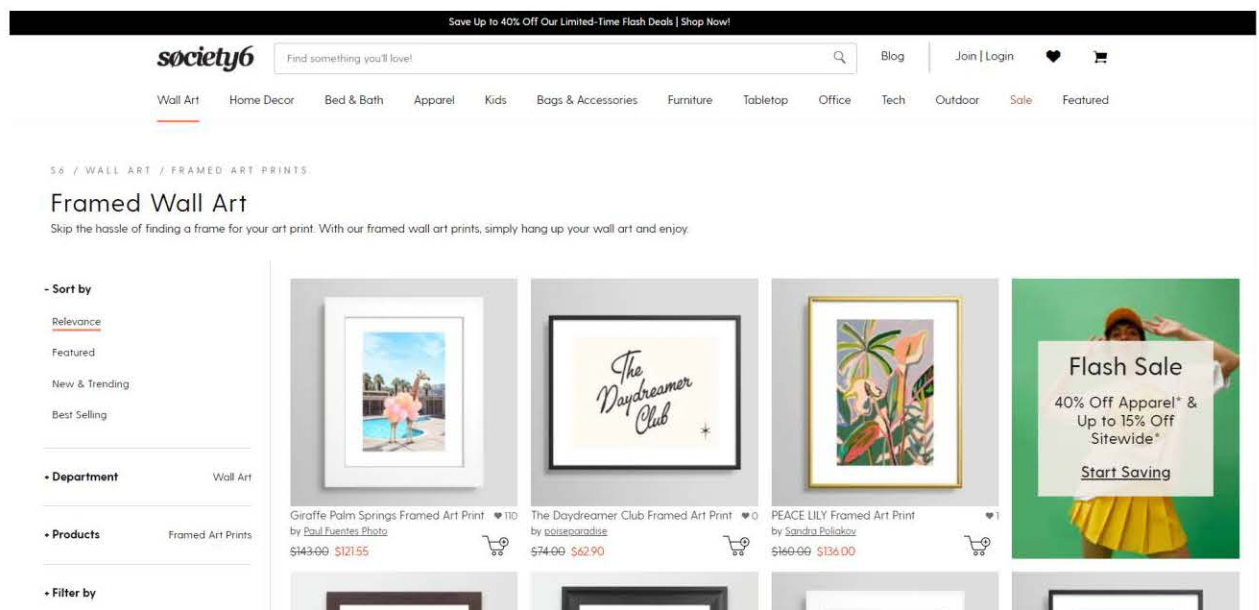


Рисунок 1.4 – Інтерфейс Society6

5) Ukrainian Art (рис. 1.5)

Ukrainian Art – це онлайн-платформа, яка спеціалізується на представленні українського мистецтва та творів українських художників. Ось деякі ключові можливості Ukrainian Art:

- надає майданчик для представлення творів українських художників та розвитку української мистецької спільноти;
- пропонує широкий вибір мистецьких творів, таких як картини, малюнки, скульптури, фотографії та інші види мистецтва, що дозволяє покупцям вибирати те, що вони бажають;
- сприяє популяризації та підтримці українських художників, допомагаючи їм збільшувати свою аудиторію та збут своїх творів;
- може організовувати виставки та інші мистецькі заходи для презентації творів українських художників та сприяння спілкуванню українських митців;
- надає можливість клієнтам купувати твори мистецтва онлайн та отримувати їх через доставку, що дозволяє зручно придбати твори для свого дому або колекції;
- сприяє збереженню та популяризації української культурної спадщини та творчої спадщини художників шляхом сприяння розвитку української мистецької сцени.

Загалом, Ukrainian Art – це платформа, яка сприяє розвитку та підтримці українського мистецтва та українських художників, пропонуючи широкий вибір творів та сприяючи їхній популяризації та продажам.

6) ArtUkraine (рис. 1.6)

Сайт "ArtUkraine" присвячений українському мистецтву та художникам. Ось кілька можливостей, які доступні на цьому сайті:

- пропонує художникам виставляти свої твори для перегляду та продажу на платформі;
- містить різноманітні види мистецтва, такі як картини, малюнки, скульптури, фотографії тощо;

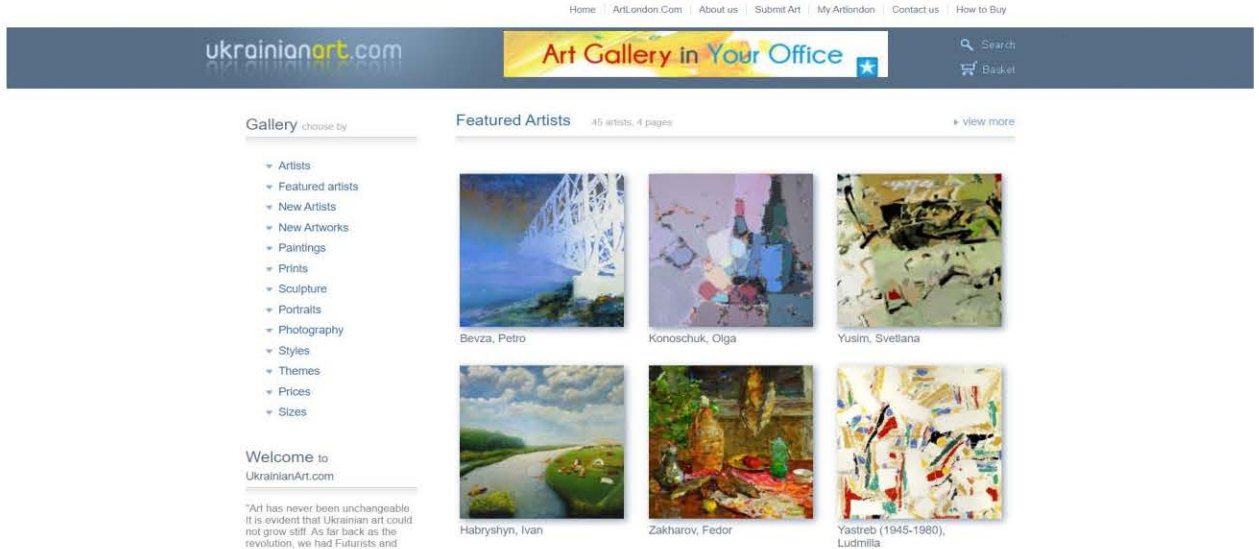


Рисунок 1.5 – Інтерфейс Ukrainian Art

- є функція магазину, де користувачі можуть придбати твори мистецтва українських художників;
- є профілі різних художників, де можна знайти інформацію про них та їхні твори;
- надає інформацію про мистецькі події, виставки, анонси нових робіт художників тощо;
- є контактна інформація, яка дозволяє зв'язатися з адміністрацією або художниками.

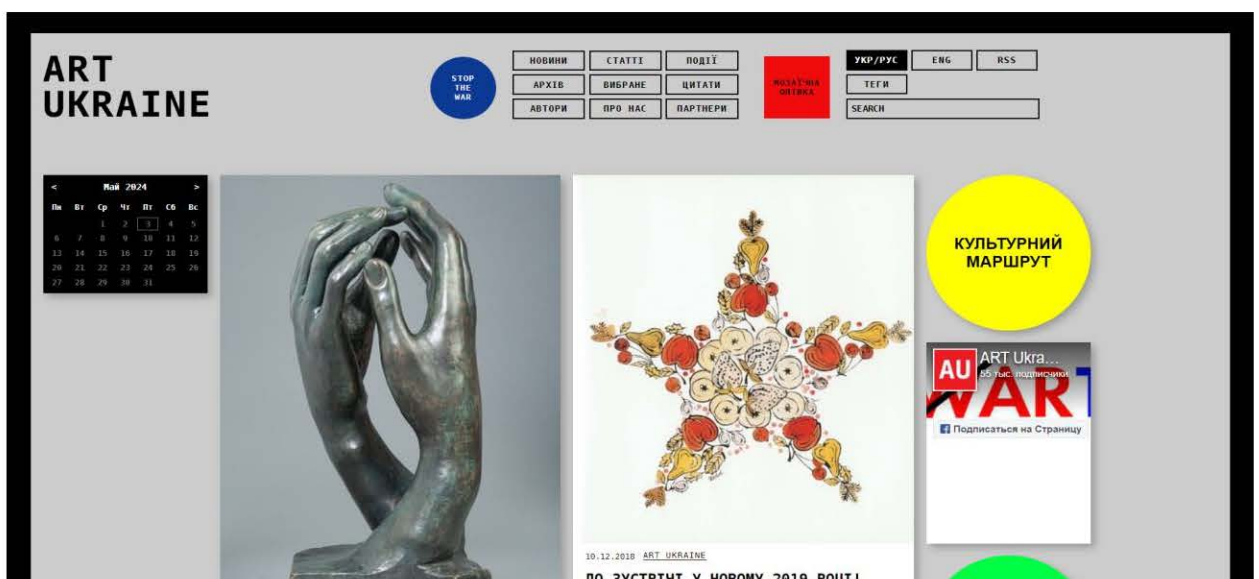


Рисунок 1.6 – Інтерфейс ArtUkraine

1.2 Аналіз методів розробки інформаційних сайтів

Аналіз методів розробки інформаційних сайтів може бути корисним для визначення найбільш відповідного способу створення вебпроєкту залежно від його мети, обсягу та потреб користувачів. До основних аспектів аналізу методів розробки можна віднести:

- розгляд і аналіз вимог до проєкту, таких як функціональність, вимоги до дизайну, потреби в безпеці та інші;
- аналіз різних технологій, мов програмування, фреймворків, бібліотек та інструментів, які можуть бути використані для реалізації функціональності сайту;
- аналіз продуктивності різних технологій та підходів до розробки, включаючи швидкість завантаження сторінок, реакцію інтерфейсу;
- розгляд і аналіз потенційних загроз безпеці для сайту та методів їх запобігання, таких як захист від атак, обробка введених даних користувача та інші;
- розгляд можливостей масштабування вебсайту для різних методів розробки, включаючи можливості розширення функціональності та обробки великого обсягу трафіку.

Вимоги до інформаційного сайту з продажу малюнків можна описати наступним чином.

Функціональні вимоги:

- сайт повинен дозволяти користувачам переглядати малюнки, які доступні для продажу; кожен малюнок повинен мати окрему сторінку з детальним описом та можливістю збільшення зображення (показ малюнків);
- користувачі повинні мати можливість шукати малюнки за категоріями, тегами, авторами тощо; також важливо мати функцію фільтрації за ціною, рейтингом тощо (пошук та фільтрація);
- сайт повинен мати можливість реєстрації нових користувачів та авторизації існуючих для здійснення покупок та відстеження замовлень

(авторизація та реєстрація);

– користувачі повинні мати можливість додавати малюнки до кошика та оформляти замовлення на покупку; після цього повинна бути можливість оплати замовлення та відстеження статусу доставки. (кошик та оформлення замовлення).

Дизайн та інтерфейс:

– дизайн сайту повинен бути сучасним та привабливим;
– сайт повинен бути інтерактивним, з можливістю плавного прокручування, анімації та швидкої відповіді на дії користувачів.

Технічні вимоги:

– система повинна бути масштабованою для підтримки великої кількості користувачів та малюнків (масштабованість);
– необхідно забезпечити високий рівень безпеки, зокрема захист від атак на базу даних та перехоплення даних користувачів (безпека);
– сайт повинен працювати швидко та ефективно, з мінімальними часами завантаження сторінок (швидкодія).

Управління контентом:

– потрібна можливість для адміністраторів додавати, редагувати та видаляти малюнки, категорії, теги, а також відстежувати замовлення та користувачів (адміністративний доступ);
– важливо мати можливість відстежувати аналітику сайту, таку як кількість переглядів, продажів, поведінку користувачів тощо (аналітика).

Під час розробки конкретні деталі можуть бути уточнені відповідно до потреб бізнесу та користувачів.

Аналіз різних мов програмування, таких як HTML, CSS, JavaScript, PHP, Python тощо, та їх використання для розроблення фронтенду та бекенду вебсайту включає розгляд різних аспектів кожної мови та їх потенційного впливу на розробку та функціональність вебпроекту.

HTML (HyperText Markup Language):

– використовується для створення структури та вмісту вебсторінок;

- не має можливостей програмування, але дозволяє створювати різні елементи інтерфейсу, такі як тексти, зображення, посилання тощо;

- використовується як основна мова для розмітки сторінок.

CSS (Cascading Style Sheets):

- використовується для оформлення вигляду та візуального представлення вебсторінок;

- дозволяє задавати стилі, кольори, розміри, відступи та інші властивості для HTML-елементів;

- допомагає забезпечити єдність в оформленні сторінок та адаптивність до різних пристроїв.

JavaScript:

- використовується для динамічного взаємодії з вебсторінкою та створення інтерактивності;

- дозволяє додавати реакції на події користувача, змінювати вміст сторінки без перезавантаження та виконувати обчислення на клієнтському боці.

PHP (Hypertext Preprocessor):

- використовується для розробки серверної частини вебдодатків (бекенду);

- дозволяє обробляти дані форм, доступ до баз даних, створення сесій та інші серверні функції;

- забезпечує динамічну генерацію вмісту на основі запитів користувача та взаємодії з базами даних.

Python:

- використовується як альтернатива PHP для розробки бекенду;

- має простий синтаксис та широкі можливості для розробки вебдодатків, включаючи фреймворки, такі як Django та Flask;

- забезпечує високу продуктивність та широкі можливості інтеграції з іншими технологіями.

Кожна з цих мов має свої переваги та обмеження, і вибір між ними

залежить від потреб та вимог конкретного вебпроекту.

Фреймворки для розроблення вебсайтів використовуються для спрощення та прискорення процесу створення вебдодатків шляхом надання заздалегідь підготовлених структур, компонентів та інструментів. Основні цілі використання:

- фреймворки надають готові структури, моделі даних та компоненти, що дозволяють розробникам швидше створювати функціональність вебдодатків (швидша розробка);

- використання фреймворків допомагає зберігати консистентність у структурі та архітектурі вебдодатків, оскільки вони пропонують заздалегідь визначені правила та стандарти (стандартизація);

- багато фреймворків мають вбудовані механізми безпеки, такі як захист від SQL-ін'єкцій, XSS-атак та інших загроз безпеці (безпека);

- фреймворки часто мають велику кількість розширень та плагінів, які дозволяють розробникам швидко додавати нові функціональності до своїх вебдодатків (розширення);

- багато фреймворків мають активні спільноти розробників, які надають допомогу, оновлення та рекомендації щодо використання фреймворку (підтримка спільноти);

- використання фреймворків дозволяє розробникам фокусуватися на бізнес-логіці вебдодатків, а не на деталях інфраструктури та рутинних завданнях (покращення продуктивності).

Загалом, фреймворки допомагають розробникам створювати високоякісні, безпечні та швидкі вебдодатки з меншими зусиллями та витратами часу.

Ось огляд популярних фреймворків для розробки вебсайтів та їх переваги та недоліки.

Переваги Bootstrap:

- легкий у використанні та швидкий у розробці;

- має велику кількість готових компонентів, які допомагають швидко

створити адаптивний та зручний дизайн;

- добре документований та підтримується активною спільнотою.

Недоліки Bootstrap:

– за замовчуванням виглядає досить загально, існує ризик, що багато вебсайтів можуть виглядати схожими;

- потребує додаткового CSS для індивідуального оформлення.

Переваги Foundation:

– пропонує велику кількість готових компонентів та можливостей для розширення;

– добре пристосований для мобільних пристроїв та адаптивного дизайну;

- легко кастомізується та розширюється за допомогою Sass.

Недоліки Foundation:

– менше популярний порівняно з Bootstrap, тому може бути менше документації та підтримки;

– може здаватися складним для новачків через більші можливості та налаштування.

Переваги React:

– дозволяє створювати динамічні та інтерактивні інтерфейси з використанням компонентів;

– широко використовується в індустрії та має велику спільноту, що розвивається;

– використовує Virtual DOM для оптимізації рендерінгу та покращення продуктивності.

Недоліки React:

– вимагає додаткових знань та навичок у порівнянні з традиційними бібліотеками;

– відсутня повноцінна підтримка для SEO, але можливо використання серверного рендерінгу.

Переваги Vue.js:

- простий у використанні та швидкий у вивченні для початківців;
- має чистий та прозорий синтаксис, що сприяє швидкому розробці;
- добре підходить для створення інтерфейсів користувача та односторінкових додатків.

Недоліки Vue.js:

- менш популярний порівняно з React та Angular, тому може бути менше ресурсів та підтримки;
- менше екосистема, ніж у React або Angular.

Переваги Angular:

- має повний набір інструментів для створення великих та складних додатків;
- добре підтримується Google та має велику активну спільноту;
- вбудований модульний підхід сприяє підтримці чистого та організованого коду.

Недоліки Angular:

- складніше для вивчення та розуміння порівняно з іншими фреймворками;
- велика кількість концепцій та абстракцій може здаватися переповненою для менш складних додатків.

Переваги Django:

- швидка розробка завдяки вбудованим функціональним можливостям та адміністративному інтерфейсу;
- сильна безпека та захист від вразливостей;
- добра документація та активна спільнота.

Недоліки Django:

- менше гнучкості порівняно з іншими фреймворками через більші можливості заздалегідь визначених структур;
- може бути перевантажений для менших або менш складних проектів.

Переваги Laravel:

- елегантний та чистий синтаксис, що полегшує розробку;

- вбудовані інструменти для автентифікації, маршрутизації, кешування та інших завдань;

- активна спільнота та велика кількість розширень та пакетів.

Недоліки Laravel:

- може вимагати додаткового часу на навчання для початківців;

- швидкість виконання може бути повільнішою порівняно з іншими фреймворками.

Кожен фреймворк має свої переваги та недоліки, і вибір залежить від конкретних потреб та вимог проекту, а також рівня знань та досвіду у розробці вебдодатків.

Бази даних грають ключову роль у створенні сучасних вебсайтів з наступних причин:

- дозволяють зберігати великі обсяги даних у структурованому форматі, це може включати інформацію про користувачів, товари, послуги, замовлення, коментарі тощо;

- оптимізовані для ефективного пошуку та отримання даних, що дозволяє швидко відображати потрібну інформацію користувачам вебсайту;

- дозволяють взаємодіяти з користувачами вебсайту, зберігати їхні дані, обробляти замовлення, реєстрації, коментарі тощо;

- за допомогою баз даних можна виконувати різноманітні операції з даними, такі як сортування, фільтрація, групування, обчислення тощо;

- надають можливості для захисту конфіденційної інформації, такої як паролі користувачів, фінансові дані тощо;

- за допомогою баз даних можна розширювати функціональність вебсайту та оброблювати великий потік даних та трафіку.

Ось порівняння найпоширеніших реляційних баз даних (MySQL, PostgreSQL, Microsoft SQL Server) та нереляційної бази даних (MongoDB).

MySQL:

- широко використовується веброзробниками та компаніями будь-якого розміру для зберігання та управління даними вебсайтів та додатків;

- зазвичай відомий своєю високою швидкістю, здатністю до роботи з великими обсягами даних та ефективним використанням пам'яті;
- добре масштабується як вертикально (за допомогою підвищення потужності сервера), так і горизонтально (шляхом розподілу навантаження);
- має велику спільноту користувачів та активну підтримку, включаючи офіційну документацію, форуми та регулярні оновлення.

PostgreSQL:

- також широко використовується для розробки вебсайтів та додатків, особливо там, де потрібна висока надійність, розширюваність та сумісність зі стандартами SQL;
- відомий своєю розширеною функціональністю, включаючи підтримку геоданих, JSON, XML та багато іншого;
- має добре розвинені засоби безпеки, такі як ролі, дозволи та механізми шифрування даних;
- добре масштабується, але у порівнянні з MySQL вимагає більше уваги та налаштувань для роботи з великими навантаженнями.

Microsoft SQL Server:

- широко використовується в корпоративному середовищі та серед веброзробників, які використовують технології Microsoft, такі як .NET;
- добре інтегрується з іншими продуктами та інструментами Microsoft, що робить його привабливим вибором для організацій, які використовують стек технологій Microsoft;
- має розширені можливості для аналітики даних та створення звітів, що робить його популярним серед бізнес-користувачів;
- один з недоліків – це високі витрати на ліцензування, що може бути проблемою для невеликих компаній та стартапів.

MongoDB:

- використовує документ-орієнтовану структуру, що дозволяє зберігати дані у вигляді JSON-подібних документів;
- добре підходить для проектів зі змінюваною структурою даних або

високої швидкодії завдяки використанню вбудованих об'єктів та гнучкої схеми;

– має хорошу горизонтальну масштабованість, що дозволяє легко розширювати базу даних за потреби.

Хоча MongoDB має засоби безпеки, вони можуть бути менш розвиненими порівняно з реляційними базами даних.

Кожна з цих баз даних має свої переваги та недоліки, і вибір залежить від конкретних потреб вашого проекту, вимог щодо продуктивності, безпеки, масштабованості та інтеграції.

1.2 Висновки до першого розділу. Постановка завдань дослідження

Після аналізу існуючих сайтів по продажу малюнків та методів їх розробки можна зробити кілька висновків.

Потреба в розвитку інтернет-платформ для продажу малюнків є актуальною та зростає. Існуючі сайти демонструють різноманітні підходи до організації контенту, функціональності та дизайну.

Розробка вебсайту для продажу малюнків вимагає досить широкого спектру технологій та інструментів. Від вибору мов програмування та фреймворків до вибору бази даних та методів безпеки, кожен аспект потребує ретельного аналізу та вибору оптимального рішення.

Продуктивність та безпека є важливими аспектами для вебсайтів, що працюють з великим обсягом контенту та особистих даних користувачів. Важливо обрати технології та методи, які забезпечать високу швидкість та захист від потенційних загроз.

Після аналізу можна зробити висновок щодо найбільш підходящого напрямку розробки вебсайту для продажу малюнків, вибрати найефективніші технології та підходи для його реалізації та успішного функціонування.

РОЗДІЛ 2 АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Вибір програмних засобів для реалізації проекту

Під час розробки програмного забезпечення можуть бути використані наступні програмні засоби:

- 1) PHP;
- 2) JavaScript;
- 3) OpenServer;
- 4) MySQL;
- 5) AJAX;
- 6) Bootstrap.

2.2 PHP

PHP (Hypertext Preprocessor) почав розвиватися в 1994 році як особистий проект Расмуса Лердорфа, який був інженером зі зв'язку. Початково це був набір скриптів, які він використовував для керування його особистим сайтом і для вирішення певних завдань на сервері. Перша версія PHP, випущена в 1995 році, мала назву «Personal Home Page/Forms Interpreter» (PHFI). Це був набір Perl-скриптів для обробки даних форм на вебсторінках. У 1995 році Лердорф переробив PHFI і перейменував його на PHP/FI (Personal Home Page/Forms Interpreter). Нова версія мала ряд значних поліпшень, включаючи підтримку роботи з базами даних. У 1995 році PHP/FI став доступним як вільне програмне забезпечення, що робить його відкритим для розробників з усього світу.

У 1997 році вийшла перша стабільна версія PHP (PHP 3.0), яка вже мала підтримку для об'єктно-орієнтованого програмування та велику кількість вбудованих функцій. Після PHP 3.0 були випущені PHP 4.0 у 2000 році, PHP 5.0 у 2004 році та PHP 7.0 у 2015 році, які вносили нові функції, поліпшення

продуктивності та безпеки, а також підтримку сучасних стандартів програмування.

PHP 8.0 – остання версія, яка вийшла наприкінці 2020 року. У цій версії мови з'явилася підтримка union-типів (структур даних або значень, які можуть мати кілька різних уявлень), анотацій (атрибути) та JIT-компіляція – процес перетворення байт-коду в машинний код під час роботи програми. Microsoft вирішила припинити підтримку версії 8.0 для Windows, тому цю роботу тепер виконують учасники спільноти, які цікавляться розробкою PHP для цієї операційної системи.

Основне використання PHP – це розробка скриптів, які виконуються на стороні сервера. Крім того, можна створювати скрипти для командного рядка та GUI-програми (додатки з графічним інтерфейсом користувача) [1–4].

PHP використовується в різноманітних проектах: від браузерних ігор та соціальних мереж до великих вебзастосунків в інтернеті, таких як Facebook, Google, W3C, Yahoo, NASA. Багато систем управління контентом (CMS), таких як Drupal, WordPress, Joomla, написані на PHP. Крім того, PHP використовується для розробки популярних фреймворків для створення сайтів, таких як Yii2, Laravel, Symfony.

PHP – мова з динамічною типізацією. Це означає, що змінна не має жорсткої та заздалегідь визначеної типізації (якщо число, то int, якщо рядок - string). Динамічна типізація дозволяє одній і тій же змінній мати кілька значень: вона може бути масивом, рядком, числом, об'єктом і т. д. Це забезпечує гнучкість в програмуванні, але може призвести до більшого використання оперативної пам'яті та зменшення швидкості виконання програми.

Крім того, PHP – мова, яка інтерпретується. Це означає, що код, написаний на PHP, не компілюється (перетворюється на машинний код), а виконується рядок за рядком за допомогою програми-інтерпретатора, яка читає кожен рядок та виконує його відповідно до вказівок.

Звичайний вебсайт, який просто відображає статичний контент (тобто

такий, який не змінюється часто), може бути створений з використанням простих HTML-сторінок. Проте, коли ми маємо справу з сайтом, на якому інформація постійно оновлюється, наприклад, щодня додаються нові малюнки для продажу або змінюється інформація про ці малюнки, простого HTML вже не вистачить.

Саме тут PHP стає корисним. Він дозволяє розробникам створювати сторінки, які можуть генерувати контент на льоту, в залежності від запитів користувачів або даних, що знаходяться у базі даних. Тобто, замість того, щоб кожен раз редагувати HTML-сторінки вручну, можна написати PHP-скрипти, які автоматично генерують HTML-код на основі поточних даних.

Це означає, що коли, наприклад, додається новий малюнок до каталогу на сайті, можна просто оновити базу даних, а потім PHP-скрипт автоматично забезпечить відображення цього малюнка на вебсторінці. Це дозволяє ефективно управляти великим обсягом інформації на сайті і швидко вносити зміни без необхідності редагувати кожен сторінку окремо.

PHP добре поєднується з різними системами управління базами даних, такими як MySQL, PostgreSQL, що дозволяє зберігати і керувати великою кількістю даних про малюнки, замовлення, користувачів тощо.

Для інформаційного сайту з продажу малюнків база даних може використовуватися для зберігання різноманітної інформації, такої як дані про малюнки (назва, опис, ціна, категорія тощо), інформація про користувачів (якщо є система реєстрації або облікових записів), інформація про замовлення та інше.

PHP має різноманітні функції для з'єднання з базами даних, виконання запитів SQL та обробки результатів. Завдяки цьому можна, наприклад, витягувати дані про малюнки з бази даних та відобразити їх на сайті, шукати малюнки за різними критеріями, додавати нові малюнки до бази даних через вебформу, оновлювати інформацію про малюнки тощо.

PHP має велику кількість вбудованих функцій та сторонніх бібліотек, що дозволяє розробникам реалізувати різноманітні функції, такі як системи

пошуку, обробка замовлень, корзина покупок тощо.

PHP може бути використаний для створення форми замовлення, де користувачі можуть вибрати малюнки, які вони бажають придбати, ввести інформацію про доставку та спосіб оплати, і надіслати цю інформацію на сервер для обробки.

Коли користувач натискає кнопку «Підтвердити замовлення», дані з форми відправляються на сервер, де PHP може обробити ці дані. Наприклад, PHP може перевірити, чи всі необхідні поля заповнені, перевірити правильність введених даних, розрахувати вартість замовлення, перевірити наявність товару на складі тощо.

Після обробки PHP може зберегти дані про замовлення у базі даних, надіслати підтвердження замовлення користувачеві по електронній пошті, а також надіслати повідомлення адміністраторам сайту для подальшої обробки замовлення.

Отже, PHP дозволяє реалізувати функціональність оформлення замовлень на сайті, що дозволяє користувачам придбати малюнки з сайту зручно та ефективно.

PHP може бути вбудований безпосередньо в HTML-код, що спрощує розробку та підтримку вебсторінок. PHP може бути вбудований безпосередньо в HTML-код в зоні PHP. Це означає, що можна вставляти фрагменти PHP-коду прямо всередину HTML-сторінок. Наприклад, можна використовувати PHP для динамічного виведення даних, генерування списків, виконання умовних операцій тощо.

PHP може використовуватися для обробки запитів, які надходять від JavaScript за допомогою технології AJAX (асинхронні JavaScript і XML). Це дозволяє оновлювати окремі частини вебсторінки без повного перезавантаження сторінки. Наприклад, можна використовувати AJAX для динамічної підгрузки додаткового контенту під час прокрутки або виконання дій користувача без перезавантаження сторінки. Для цього PHP може обробляти AJAX-запити, взаємодіяти з базою даних або іншими джерелами

даних, та повертати відповіді у форматі, який може бути легко оброблений за допомогою JavaScript.

Ці аспекти дозволяють розробникам створювати більш динамічні та інтерактивні вебсайти з використанням PHP в поєднанні з HTML і JavaScript. Такий підхід дозволяє створювати зручні та ефективні вебдодатки, які можуть відповідати потребам користувачів.

PHP є однією з найпопулярніших мов програмування для розробки вебсайтів, і вона часто використовується для створення інформаційних сайтів з продажу.

2.3 JavaScript

JavaScript – це мова програмування, яка використовується для розробки динамічних інтерактивних вебсайтів. Вона дозволяє створювати різноманітні ефекти, взаємодію користувача з вебсторінкою, асинхронне завантаження даних і багато іншого [5–7]. JavaScript використовується як на клієнтській стороні (у браузерах користувачів), так і на серверній стороні (за допомогою платформи Node.js).

JavaScript має можливість взаємодіяти з HTML і CSS. Це означає, що можна змінювати вміст вебсторінки, стилізувати її елементи і керувати їх властивостями. Наприклад, можна змінювати текст на сторінці, приховувати або показувати елементи, змінювати їх розмір, колір, шрифт і так далі.

JavaScript може змінювати вміст HTML елементів. Наприклад, ви можете змінювати текст або HTML вміст елементів, додаючи або видаляючи їх (рис. 2.1).

JavaScript може реагувати на події користувача, такі як кліки мишею, натискання клавіш, наведення курсора тощо, і виконувати відповідні дії (рис. 2.3).


```

<button onclick="changeText()">Змінити текст</button>
<p id="demo">Привіт, світ!</p>

<script>
  function changeText() {
    document.getElementById("demo").innerHTML = "Привіт, JavaScript!";
  }
</script>

```

Рисунок 2.1 – Зміна вмісту HTML елементів

Також JavaScript може змінювати стилі CSS елементів, що дозволяє створювати анімації, змінювати розміри, кольори та інші властивості елементів. (рис. 2.2).

```

<button onclick="changeColor()">Змінити колір</button>
<p id="demo">Привіт, світ!</p>

<script>
  function changeColor() {
    document.getElementById("demo").style.color = "red";
  }
</script>

```

Рисунок 2.2 – Зміна стилів CSS елементів

```

<button onclick="alert('Привіт!')">Натисни мене</button>

```

Рисунок 2.3 – Реакція на натискання кнопки

Взаємодія з HTML і CSS дозволяє створювати динамічні інтерфейси та ефекти, що покращують користувацький досвід вебсторінки.

В JavaScript можна визначати реакцію на різні події, такі як клік мишею, натискання клавіші або завантаження сторінки. Кожна з цих подій може мати визначену в JavaScript функцію, яка виконується при виникненні цієї події.

Наприклад, можна встановити, що при кліку на кнопку виконується певна дія, або при введенні користувачем тексту в поле вводу відбувається якась обробка. Це дозволяє створювати інтерактивність на вебсторінці і реагувати на дії користувача.

Також JavaScript надає можливість динамічно змінювати структуру і вміст вебсторінки, використовуючи Document Object Model (DOM). Завдяки цій можливості, можна динамічно змінювати структуру, вміст і стилі вебсторінки під час її взаємодії з користувачем. Наприклад, можна додавати, видаляти або змінювати елементи на сторінці, змінювати їх атрибути або стилі, а також взаємодіяти з подіями, пов'язаними з цими елементами. Це дозволяє створювати динамічні інтерфейси та забезпечує більшу гнучкість у взаємодії з користувачем на вебсторінці.

JavaScript підтримує асинхронний підхід до програмування. Це означає, що можна виконувати деякі операції без очікування завершення попередніх дій. Зокрема, це стосується виконання операцій, які потребують часу, таких як завантаження даних з сервера або виконання складних обчислень.

Зокрема JavaScript дозволяє виконувати асинхронні запити, щоб завантажити дані з сервера, а також виконувати інші операції, поки запит обробляється. Це дозволяє створювати більш ефективні та ресурсозберігаючі вебдодатки, оскільки користувачі можуть взаємодіяти зі сторінкою, не очікуючи завершення всіх процесів.

Наприклад, можна завантажувати додаткові дані на сторінку під час її завантаження, або відправляти запити на сервер для отримання актуальних даних, під час використання користувачем вашого додатку. Такий асинхронний підхід дозволяє покращити швидкість та ефективність вебзастосунку.

2.4 OpenServer

OpenServer – це локальний вебсервер, який призначений для розробки та

тестування вебсайтів та вебдодатків на комп'ютері [8].

Він має простий інсталятор, який дозволяє швидко встановити та налаштувати всі необхідні компоненти для роботи локального вебсервера, включає в себе популярні вебсервери, бази даних та інші середовища розробки, що дозволяє працювати з різними технологіями веброзробки без додаткових витрат часу на їх налаштування.

Завдяки OpenServer можна розробляти та тестувати вебсайти та вебдодатки на комп'ютері, не підключаючись до зовнішнього сервера, підтримує встановлення різних розширень та модулів, що дозволяє розширити його функціональність та використовувати додаткові інструменти для розробки.

OpenServer включає в себе комплексний набір різноманітних технологій, необхідних для розробки та тестування вебсайтів та вебдодатків. Ось детальний огляд деяких з них:

Вебсервери:

- Apache – один з найпоширеніших вебсерверів, який підтримує велику кількість функцій та можливостей для налаштування вебсайтів;
- Nginx – легкий та швидкий вебсервер, який часто використовується для обслуговування статичного контенту та високонавантажених вебдодатків.

Бази даних:

- MySQL – популярна система керування базами даних з відкритим кодом, яка використовується для зберігання та обробки даних;
- MariaDB – форк MySQL, який зберігає сумісність з оригінальним MySQL та додає деякі нові функції.

Мови програмування:

- PHP – мова програмування, яка широко використовується для розробки вебдодатків та динамічних вебсайтів;
- Python – універсальна мова програмування, яка використовується для розробки вебдодатків, наукових досліджень, автоматизації та інших завдань.

Інші середовища розробки:

- phpMyAdmin – вебінтерфейс для управління базами даних MySQL/MariaDB, який дозволяє легко створювати, редагувати та керувати базами даних;

- Memcached – система кешування, яка використовується для прискорення роботи вебсайтів шляхом зберігання даних у швидкодіючій пам'яті;

- Redis – інша система кешування та зберігання даних у швидкодіючій пам'яті, яка підтримує різні типи даних та операції з ними.

Ці технології входять до складу OpenServer і дозволяють розробникам працювати з різними аспектами веброзробки без необхідності окремого налаштування кожного компонента. Це спрощує процес розробки та дозволяє швидше створювати та тестувати вебдодатки.

2.5 MySQL

Використання баз даних дозволяє зручно та ефективно управляти великим обсягом даних, які пов'язані з інформаційним сайтом, і забезпечує швидкий доступ до цієї інформації для користувачів сайту.

MySQL – це потужна система керування базами даних (СКБД), яка використовується для зберігання, керування та доступу до даних. Розповсюджується під ліцензією GPL (General Public License), що означає, що можна використовувати безкоштовно та змінювати вихідний код за умови дотримання умов ліцензії [2–4].

Надійність MySQL – одна з ключових характеристик, яка робить його популярним вибором для багатьох вебпроектів та корпоративних систем [9–11].

MySQL здатний ефективно обробляти великі обсяги даних та запитів. Він має оптимізований двіжок бази даних, який дозволяє впоратися зі значними обсягами трафіку та високою кількістю одночасних з'єднань.

Може бути налаштований для забезпечення високого рівня доступності

за допомогою реплікації та класів відмовостійкості. Реплікація дозволяє створювати копії бази даних для забезпечення резервних копій та шкалювання навантаження, тоді як відмовостійкість дозволяє системі продовжувати працювати навіть у випадку виникнення проблем або збоїв.

MySQL підтримує транзакції, що дозволяє виконувати групу операцій як єдину атомарну операцію. Це гарантує консистентність та цілісність даних, навіть у випадку виникнення збоїв або помилок.

Також MySQL має механізми автоматичного відновлення після збоїв, що дозволяє відновити базу даних до попереднього стану після непередбачених ситуацій, таких як відмова жорсткого диска або втрата живлення.

Використовує транзакційні журнали для забезпечення стійкості до збоїв та відновлення даних у випадку аварійного завершення роботи системи.

MySQL може включати вебсайти, електронні комерційні платформи, системи управління вмістом, фінансові системи, системи аналізу даних та багато іншого. Наприклад, WordPress використовує MySQL для зберігання контенту, а Facebook використовує його для зберігання мільярдів користувацьких профілів та даних.

2.6 AJAX

AJAX (Asynchronous JavaScript and XML) – це технологія, яка дозволяє вебсторінці асинхронно обмінюватися даними з сервером без перезавантаження всієї сторінки. Це дозволяє створювати більш динамічні та інтерактивні вебсайти, які реагують на дії користувачів без затримок [12–13].

Під час використання AJAX, вебсторінка може відправляти запити на сервер та отримувати відповіді без перезавантаження всієї сторінки. Це означає, що користувач може продовжувати взаємодіяти з іншими елементами сторінки, включаючи введення тексту або натискання кнопок, під час очікування на відповідь від сервера.

AJAX-запити виконуються в фоновому режимі, що означає, що вони не блокують виконання інших скриптів або операцій користувача на сторінці. Користувач може продовжувати робити інші дії на сторінці, поки відбувається обмін даними з сервером.

Після відправки запиту до сервера, вебсторінка продовжує виконувати інші дії, а коли від сервера надходить відповідь, вона обробляється асинхронно. Це дозволяє забезпечити швидку відповідь на запити та покращити взаємодію з користувачем.

AJAX дозволяє створювати більш інтерактивні вебсайти, які реагують на дії користувача майже миттєво, оскільки вони можуть виконувати запити до сервера та оновлювати частини сторінки без перезавантаження.

За допомогою AJAX можна обмінюватися різними типами даних з сервером. AJAX може відправляти та отримувати звичайні текстові дані. Це може бути, наприклад, зміст статті, зміст файлу або будь-який інший текстовий зміст, який потрібно передати між вебсторінкою та сервером.

Також AJAX може отримувати HTML-код з сервера та оновлювати частини вебсторінки без перезавантаження всієї сторінки. Це дозволяє динамічно оновлювати зміст сторінки, наприклад, під час відповіді на дії користувача або оновлення інформації.

JSON (JavaScript Object Notation) є популярним форматом обміну даними, особливо у веброзробці. AJAX може використовувати JSON для взаємодії з сервером, передаючи структуровані дані, такі як об'єкти та масиви, між вебсторінкою та сервером.

XML (Extensible Markup Language) є ще одним форматом для обміну даними, який може бути використаний з AJAX. Хоча JSON став більш популярним у веброзробці, XML залишається корисним для деяких типів даних та взаємодії з іншими системами.

AJAX дозволяє оновлювати лише певні частини вебсторінки, не перезавантажуючи всю сторінку. Ця можливість відкриває безліч можливостей для поліпшення користувацького досвіду:

Коли користувач взаємодіє з вебсторінкою, наприклад, натискаючи кнопку або вводячи текст у поле, AJAX дозволяє оновити лише ту частину сторінки, яка потребує змін. Наприклад, ви можете виконати пошук на сайті і показати результати без перезавантаження всієї сторінки.

Оскільки AJAX запити виконуються асинхронно, користувачі можуть отримувати оновлення майже миттєво, без очікування перезавантаження сторінки. Це робить вебдодатки більш плавними та швидкими в порівнянні з традиційними підходами.

Оскільки лише частини сторінки оновлюються за допомогою AJAX, це допомагає економити трафік та ресурси сервера. Замість того, щоб кожного разу пересилати всю сторінку, AJAX дозволяє взаємодіяти лише з необхідною інформацією.

Завдяки можливості оновлювати окремі елементи сторінки без перезавантаження, ви можете створювати більш динамічний та зручний користувацький інтерфейс. Наприклад, ви можете динамічно завантажувати нові повідомлення у потоці новин, не перезавантажуючи всю сторінку.

Для виконання AJAX-запитів у браузері використовуються два основних інструменти: об'єкт XMLHttpRequest (XHR) та Fetch API.

XMLHttpRequest є традиційним методом виконання AJAX-запитів у браузері. Цей об'єкт дозволяє створювати HTTP-запити до сервера без перезавантаження сторінки.

Fetch API надає більш розширені можливості контролю над запитом та відповіддю, такі як управління заголовками, файлами, кукісами тощо; підтримує потоки (streams), що дозволяє обробляти великі об'єми даних асинхронно та ефективно.

Обидва ці методи дозволяють виконувати асинхронні запити до сервера та обмін даними між клієнтом та сервером без перезавантаження сторінки, проте Fetch API вважається більш передовим та зручним у використанні завдяки своїм сучасним можливостям.

2.7 Bootstrap

Bootstrap – це потужний фреймворк для розробки вебсайтів та вебдодатків з використанням HTML, CSS і JavaScript. Він надає набір готових стилів, компонентів і інструментів, які допомагають створювати швидкі, ефективні та адаптивні користувацькі інтерфейси. Також він містить велику кількість готових компонентів, таких як кнопки, форми, навігаційні панелі, каруселі, модальні вікна та багато іншого. Ці компоненти можна легко використовувати в будь-якому проекті, щоб швидко створити стильні та функціональні інтерфейси [14–15].

Bootstrap пропонує готові класи для створення адаптивних інтерфейсів, які автоматично пристосовуються до різних розмірів екрану. Це дозволяє вебсайту виглядати добре на будь-якому пристрої, включаючи комп'ютери, планшети та смартфони.

Приклад адаптивного дизайну за допомогою Bootstrap, в якому використано адаптивне зображення (рис. 2.4.) У цьому прикладі примінено клас `img-fluid`, який вказує Bootstrap зробити зображення адаптивним. Це означає, що зображення буде автоматично масштабуватися, щоб зберегти свій вигляд і пропорції на будь-якому пристрої.

Bootstrap базується на системі сітки з колонками та рядами, що дозволяє легко розміщувати вміст на сторінці і керувати його розташуванням на різних екранах. Він надає ряд налаштувань та можливостей для керування виглядом і поведінкою компонентів. Ви можете змінювати кольори, розміри, шрифти та інші стилі за допомогою власних CSS правил або вбудованих класів фреймворка.

Bootstrap підтримує всі основні веббраузери і забезпечує єдинообразний вигляд вебсайту незалежно від того, який браузер використовується. Він може бути легко розширений за допомогою власних CSS стилів, JavaScript плагінів або розширень сторонніх розробників. Можна створювати власні компоненти, які інтегруються з фреймворком або розширюють його функціональність.


```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Адаптивне зображення за допомогою Bootstrap</title>
  <!-- Підключення Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css
</head>
<body>
  <div class="container">
    
  <!-- Підключення Bootstrap JavaScript (необов'язково, але корисно для певних компонентів) -->
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js
</body>
</html>

```

Рисунок 2.4 – Приклад адаптивного дизайну за допомогою Bootstrap

Отже, Bootstrap дозволяє розробникам швидко створювати стильні та функціональні вебсайти з мінімальними зусиллями, що робить його одним з найпопулярніших фреймворків для веброзробки.

2.8 Висновки до другого розділу

Після аналізу програмних засобів для реалізації інформаційного сайту з продажу малюнків, можна зробити наступні висновки.

PHP є потужною серверною мовою програмування, яка широко використовується для розробки вебсайтів. Її можливості дозволяють створювати динамічний та інтерактивний контент, обробляти форми, працювати з базами даних та здійснювати взаємодію з користувачами. Використання PHP в проекті дозволить реалізувати багато функціональних можливостей сайту.

JavaScript є найпоширенішою мовою програмування для реалізації динамічності та інтерактивності на вебсайтах. Вона дозволяє взаємодіяти з

користувачем без перезавантаження сторінки, а також створювати асинхронні запити до сервера за допомогою AJAX. Використання JavaScript дозволить покращити користувацький досвід та забезпечити більшу функціональність сайту.

OpenServer є середовищем для локальної розробки та тестування вебсайтів на власному комп'ютері. Він надає можливість швидко встановити та налаштувати необхідні компоненти для роботи сервера, що дозволяє розробникам працювати над проектом в зручному середовищі.

MySQL є однією з найпопулярніших систем управління базами даних (СУБД) та широко використовується для зберігання та управління даними на вебсайтах. Використання MySQL дозволить забезпечити надійну та ефективну роботу з базою даних для сайту з продажу малюнків.

AJAX дозволяє виконувати асинхронні запити до сервера без перезавантаження сторінки, що забезпечує більш швидкі та зручні взаємодії з користувачем. Використання AJAX дозволить реалізувати динамічне оновлення контенту на вебсайті та зменшити час завантаження сторінок.

Bootstrap є потужним фреймворком для розробки вебсайтів з великою кількістю готових компонентів та стилів. Використання Bootstrap дозволить швидко та ефективно створити сучасний та адаптивний дизайн для сайту з продажу малюнків.

Загалом, комбінація вищезазначених програмних засобів дозволить створити потужний та ефективний інформаційний сайт для продажу малюнків, забезпечуючи високу функціональність, ефективність та користувацький комфорт.

РОЗДІЛ 3 РОЗРОБКА

3.1 Проектування бази даних

База даних буде мати наступні сутності та атрибути.

Сутність «Користувач» в базі даних використовується для зберігання інформації про користувачів системи. Ось атрибути даної сутності.

Код: унікальний ідентифікатор користувача, який буде використовуватися для посилання на користувача в системі та для внутрішньої ідентифікації.

Пошта: адреса електронної пошти користувача, яка буде використовуватися для відновлення пароля, сповіщень та іншої комунікації.

Логін: ім'я користувача (унікальна ідентифікація), що буде використовуватися для входу в систему.

Пароль: зашифрований пароль користувача, який дозволить захищати доступ до облікового запису.

Атрибути «Прізвище», «Ім'я», «По батькові»: особисті дані користувача, які будуть використовуватися для ідентифікації та персоналізації в системі.

Стать: інформація про стать користувача, яка буде використана для персоналізації контенту та функцій системи.

Дата народження: дата народження користувача, яка буде використана для персоналізації контенту, а також для встановлення вікових обмежень.

Мобільний номер: номер телефону користувача, який буде використаний для сповіщень та двофакторної аутентифікації.

Роль: роль або права доступу користувача в системі, які будуть визначати його можливості та дозволи в системі (наприклад, адміністратор, модератор, звичайний користувач тощо).

Сутність «Рисунок» у базі даних буде використовуватися для зберігання інформації про малюнки та графічні зображення. Ось атрибути даної сутності.

Код: унікальний ідентифікатор рисунка, який буде використовуватися

для посилання на нього в системі або для внутрішньої ідентифікації.

Код стилю (підстилю): посилання на код стилю, який буде вказувати на певний стиль малюнка та художнього напрямку, що допоможе класифікувати рисунок та організувати їх в системі.

Код формату: посилання на код формату, який буде вказувати на формат зображення (JPEG, PNG, SVG тощо) та на формат збереження малюнка в базі даних.

Сутність «Формат» у базі даних буде використовуватись для зберігання інформації про різні типи форматів даних. Ось атрибути даної сутності.

Код: унікальний ідентифікатор формату, який буде використовуватися для посилання на нього в системі та для внутрішньої ідентифікації.

Тип формату: опис типу формату даних (JPEG, PNG, SVG, PDF, DOCX тощо). Це поле визначає, які типи файлів будуть підтримуватись системою та які додаткові дії будуть виконані з цими типами файлів.

Сутність «Стиль малюнка» у базі даних буде використовуватися для зберігання інформації про стилі малюнків та графічних зображень. Ось атрибути даної сутності.

Код: унікальний ідентифікатор стилю малюнка, який буде використовуватися для посилання на нього в системі та для внутрішньої ідентифікації.

Назва стилю: назва та опис стилю рисунка, що дозволить класифікувати різні малюнки за їхнім художнім напрямком та стилем.

Сутність «Малюнок у прайсі» у базі даних буде використовуватися для зберігання інформації про ціни на різні малюнки у прайс-листі. Ось атрибути даної сутності.

Код: унікальний ідентифікатор запису про малюнок у прайс-листі, який буде використовуватися для посилання на нього в системі та для внутрішньої ідентифікації.

Код малюнка: посилання на код конкретного малюнка з бази даних, що буде вказувати на те, який саме малюнок включений до прайс-листа.

Код прайса: посилання на код прайс-листа та цінової категорії, до якої відноситься малюнок. Це дозволить організувати різні групи малюнків з різними цінами.

Ціна: вартість конкретного малюнка в рамках прайс-листа, яка буде визначатися відповідно до цінової політики та інших факторів.

Сутність «Замовлення» у базі даних буде використовуватися для зберігання інформації про замовлення клієнтів на різні товари та послуги. Ось атрибути даної сутності.

Код: унікальний ідентифікатор запису про замовлення, який буде використовуватися для посилання на нього в системі та для внутрішньої ідентифікації.

Код користувача: посилання на код користувача, який зробив замовлення. Це дозволить встановити зв'язок між замовленням і конкретним користувачем.

Код малюнка в прайс-листі: посилання на код малюнка з прайс-листа, який буде замовлений користувачем.

Дата дедлайна: дата, до якої буде потрібно виконати замовлення. Це дозволить контролювати терміни виконання та вчасно виконувати замовлення.

Кількість осіб на малюнку: кількість осіб, які повинні бути на малюнку та зображенні. Це буде важливою інформацією для виконавців, які створюють рисунок.

Дата оформлення замовлення: дата, коли замовлення буде оформлено користувачем. Це дозволить відстежувати часову історію замовлення та вирішувати питання щодо термінів.

Сутність «Прайс» у базі даних буде використовуватися для зберігання інформації про ціни на товари або послуги. Ось атрибути даної сутності.

Код: унікальний ідентифікатор запису про прайс, який буде використовуватися для посилання на нього в системі та для внутрішньої ідентифікації.

Дата встановлення ціни: дата, коли була встановлена ціна на товар та послугу. Це дозволить відстежувати історію зміни цін та визначати, як ціни змінюються з часом.

Сутність «Підстиль малюнка» у базі даних буде використовуватися для зберігання інформації про підстилі малюнків та графічних зображень. Ось атрибути даної сутності.

Код: унікальний ідентифікатор запису про підстиль рисунка, який буде використовуватися для посилання на нього в системі та для внутрішньої ідентифікації.

Назва підстилю: назва та опис підстилю малюнку, що дозволить класифікувати різні малюнки за їхнім художнім підстилем та технікою малювання.

Сутність «Підстиль у стилі» буде використовуватися для зв'язування підстилів малюнків з конкретними стилями малювання. Ось атрибути даної сутності.

Код: унікальний ідентифікатор запису про підстиль в стилі, який буде використовуватися для посилання на нього в системі та для внутрішньої ідентифікації.

Код стилю: посилання на код конкретного стилю малювання з бази даних, що буде вказувати на те, який саме стиль включений до підстилю в стилі.

Код підстилю малюнка: посилання на код підстилю малюнка з бази даних, який буде відповідати конкретному стилю малювання.

Оскільки в базі даних багато сутностей і вони всі між собою пов'язані, проаналізуємо всі зв'язки між сутностями.

«Користувач» – «Замовлення». Зв'язок один-до-багатьох: один користувач може мати багато замовлень, але кожне замовлення належить лише одному користувачу.

Поле «Код користувача» у сутності «Замовлення» є зовнішнім ключем, який посилається на поле «Код» у сутності «Користувач».

«Користувач» – «Малюнок у прайсі». Зв'язок багато-до-багатьох: один користувач може замовити багато різних малюнків, і кожен малюнок може бути замовлений багатьма користувачами.

Цей зв'язок вимагає введення додаткової таблиці, яка буде мати зовнішні ключі на обидва записи: «Код користувача» та «Код малюнка у прайсі».

«Малюнок» – «Малюнок у прайсі». Зв'язок один-до-багатьох: один малюнок може бути включений у багато прайсів, але кожен запис про малюнок в прайсі належить лише одному малюнку.

Поле «Код малюнка» у сутності «Малюнок у прайсі» є зовнішнім ключем, який посилається на поле «Код» у сутності «Малюнок».

«Малюнок» – «Підстиль малюнка». Зв'язок один-до-багатьох: один малюнок може мати багато різних підстилів, але кожен підстиль належить лише одному малюнку.

Цей зв'язок вимагає введення додаткової таблиці, яка буде мати зовнішні ключі на обидва записи: «Код малюнка» та «Код підстилю малюнка».

«Прайс» – «Малюнок у прайсі». Зв'язок один-до-багатьох: один прайс може мати багато різних записів про малюнки в прайсі, але кожен запис про малюнки в прайсі належить лише одному прайсу.

Поле «Код прайсу» у сутності «Малюнок у прайсі» є зовнішнім ключем, який посилається на поле «Код» у сутності «Прайс».

«Підстиль малюнка» – «Підстиль у стилі». Зв'язок один-до-багатьох: один підстиль малюнка може належати до багатьох підстилів у стилі, але кожен підстиль у стилі належить лише одному підстилю малюнка.

Поле «Код підстилю малюнка» у сутності «Підстиль у стилі» є зовнішнім ключем, який посилається на поле «Код» у сутності «Підстиль малюнка».

Ці зв'язки допомагають встановити співвідношення між сутностями бази даних і визначити, які дані належать до кожної сутності та як вони пов'язані один з одним.

3.2 Види профілей користувачів

Даний web-сайт має 3 види користувацьких профілів:

1. адміністратор;
2. зареєстрований користувач;
3. незареєстрований користувач.

Адміністраторський профіль дає доступ до перегляду всіх даних та всіх таблиць з бази даних, та також, так як він являє собою зареєстрованого користувача – зробити замовлення.

Зареєстрований користувацький профіль дозволяє тільки користуватись контентом та надає змогу оформити замовлення. Користувач може переглядати всю інформацію по стилям та контактну інформацію.

Незареєстровані користувачі можуть лише подивитись контент сайту, але не можуть зробити замовлення. Головна сторінка сайту має такий зовнішній вид (рис. 3.1).

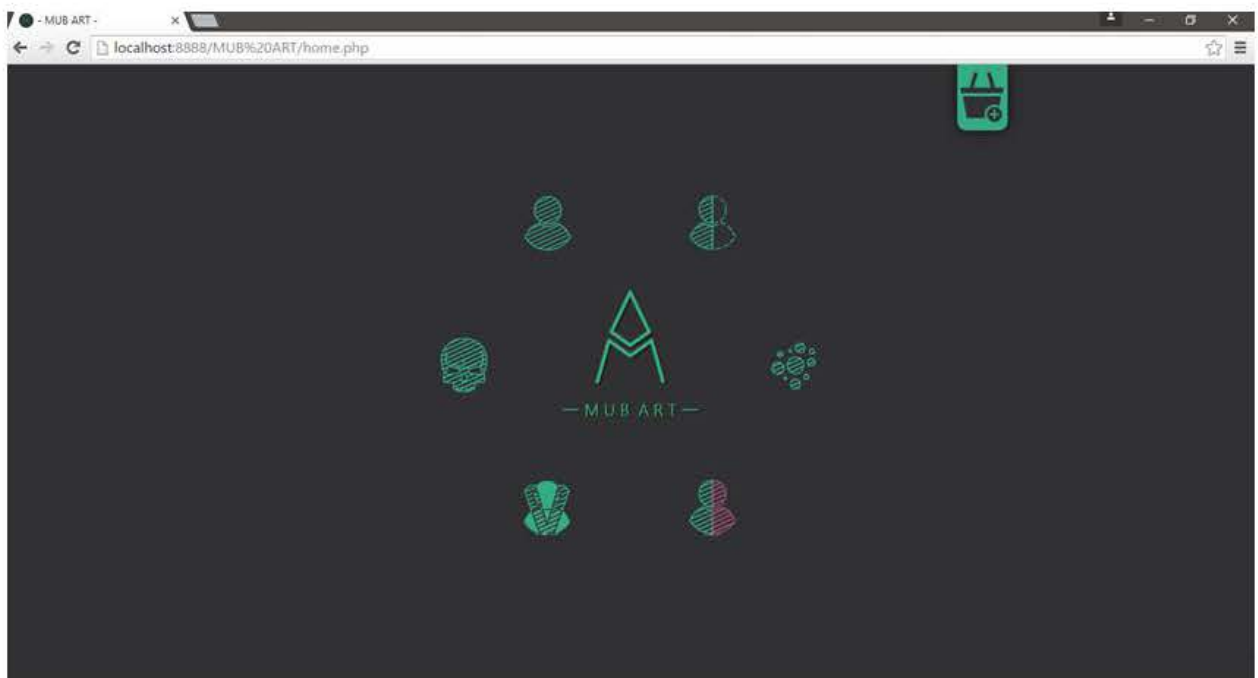


Рисунок 3.1 – Головна сторінка сайту

3.3 Реєстрація користувачів. Авторизація.

Після завантаження головної сторінки на екрані відображаються:

- головний логотип «- MUB ART -» (перехід на контактну інформацію);
- 6-ть стилів малювання;
- «Корзина».

Гість має право передивлятися будь яку інформацію стосовно малюнків, але після переходу у «Корзину» він повинен авторизуватися, для подальшого оформлення замовлення, якщо ні – він продовжить користування сайтом як «Гість» (рис. 3.2).

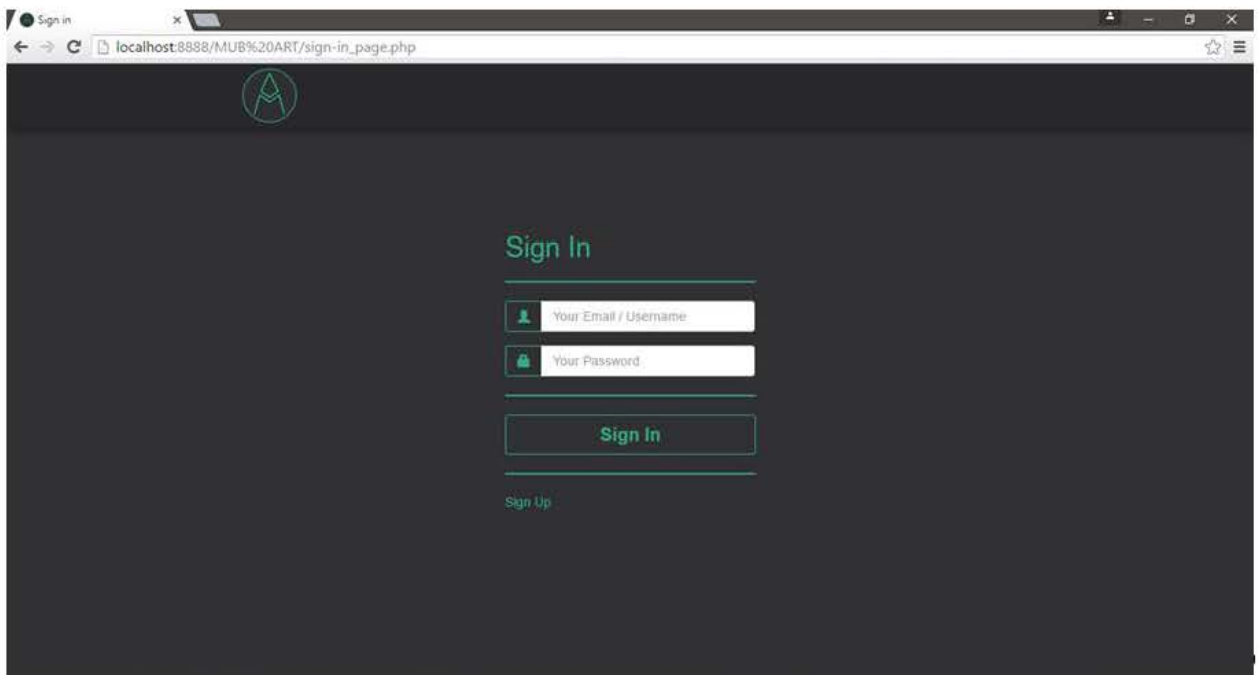
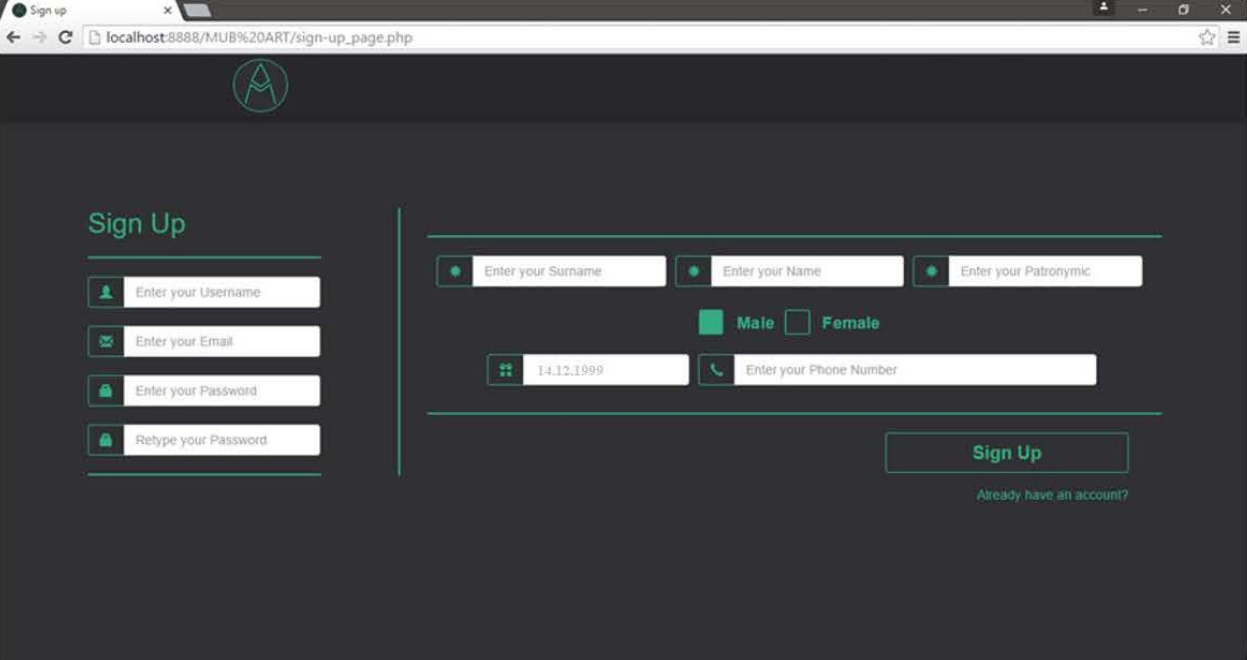


Рисунок 3.2 – Форма авторизації користувача

У випадку, коли користувач не був зареєстрований на сайті він має право зареєструватися. Для цього потрібно перейти на сторінку реєстрації, щоб продовжити подальші дії (рис. 3.3).



Sign up

localhost:8888/MUB%20ART/sign-up_page.php

Sign Up

Enter your Username

Enter your Email

Enter your Password

Retype your Password

Enter your Surname

Enter your Name

Enter your Patronymic

Male Female

14.12.1999

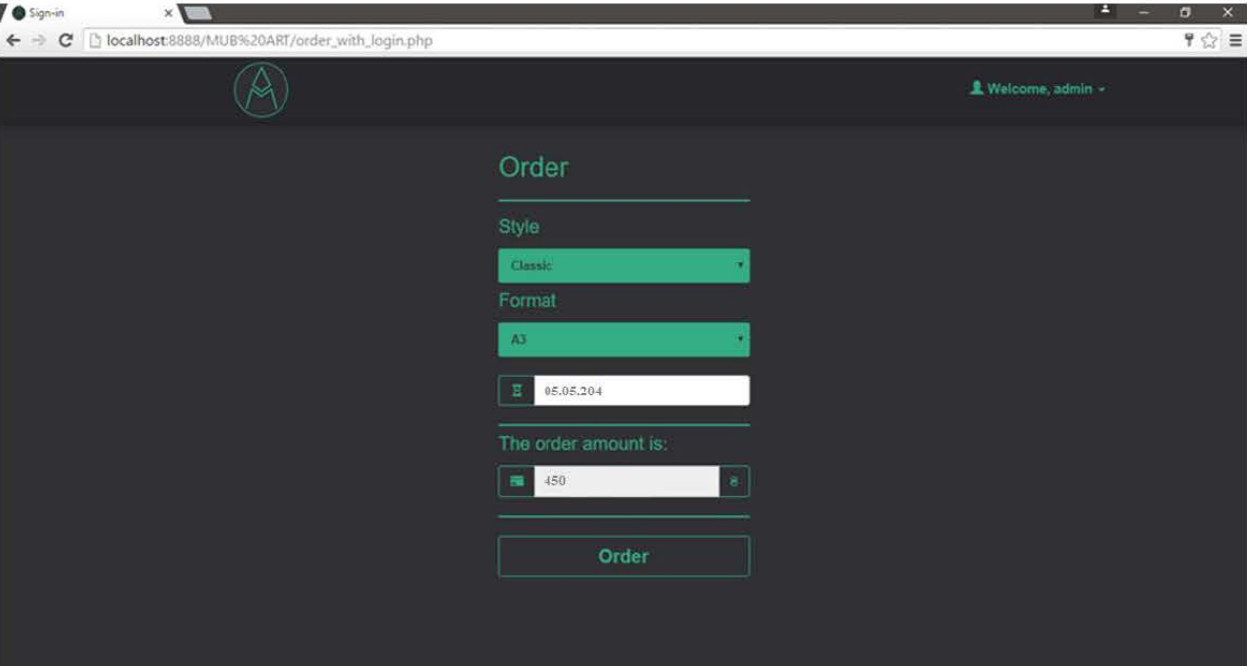
Enter your Phone Number

Sign Up

Already have an account?

Рисунок 3.3 – Форма реєстрації користувача

Після того як була зроблена вдала реєстрація або авторизація користувач автоматично переходить на сторінку замовлення (рис. 3.4).



Sign-in

localhost:8888/MUB%20ART/order_with_login.php

Welcome, admin

Order

Style

Classic

Format

A3

05.05.2024

The order amount is:

450

Order

Рисунок 3.4 – Форма оформлення замовлення

3.4 Користування контентом

Як вже було показано головній сторінці (рис. 3.1) його меню складають 6-ть стилів (при переході на які ми потрапляємо на відповідний стиль з його короткою інформацією), головний логотип (при переході ми потрапляємо на інформацію митця та його дані) та «Корзина». При наведенні курсора на кнопку стилю можна побачити його назву та візуально впевнитись, що саме ми натискаємо (рис. 3.5).

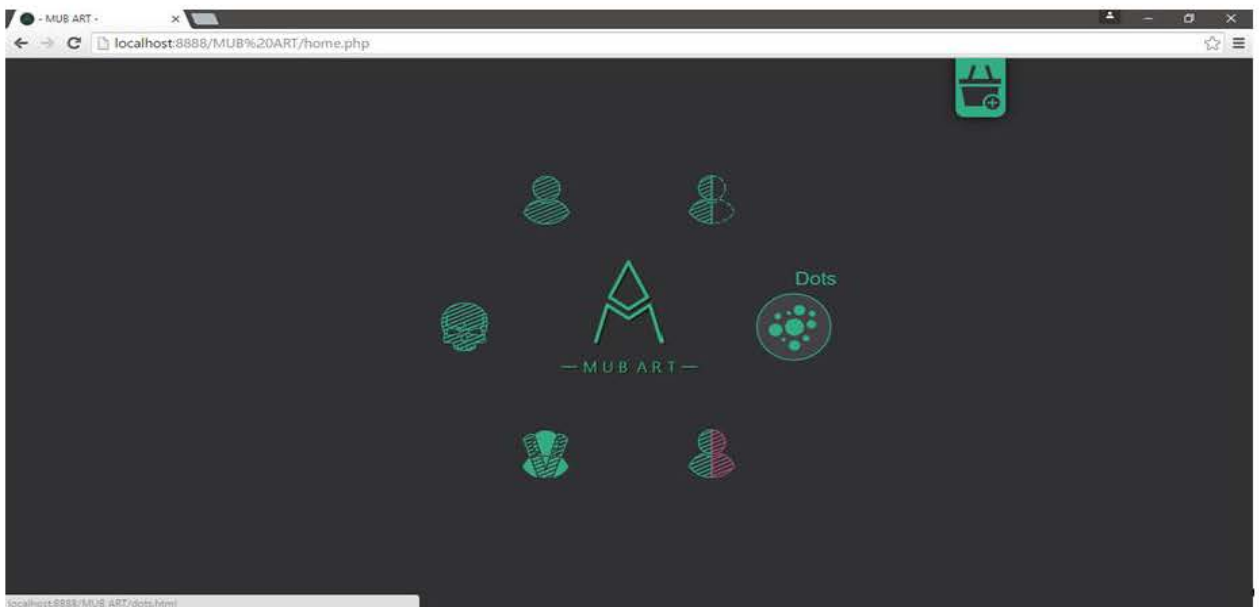


Рисунок 3.5 – Наведення на кнопку стилю «Крапки»

3.4.1 Користування сервісом перегляду інформації

Оскільки структура наповнення однакова, подивимось сторінку на прикладі стилю «Класика» (рис. 3.6 та рис. 3.7).

Сторінка складається з:

- фіксованої панелі (дозволяє повернутися на головну сторінку та

перейти на сторінку оформлення замовлення);

- кнопки детальної інформації (i) (дозволяє перейти на сторінку детальної інформації);
- опису короткої інформації;
- зображення прайсу;
- галереї малюнків (по відповідному стилю).

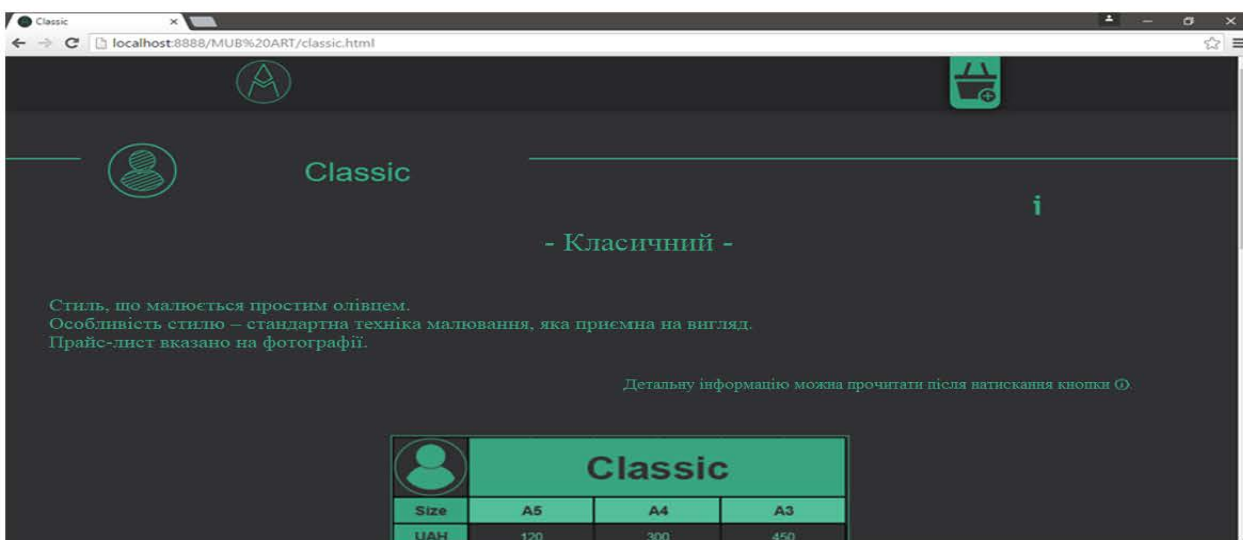


Рисунок 3.6 – Сторінка стилю «Класика» (Частина 1)

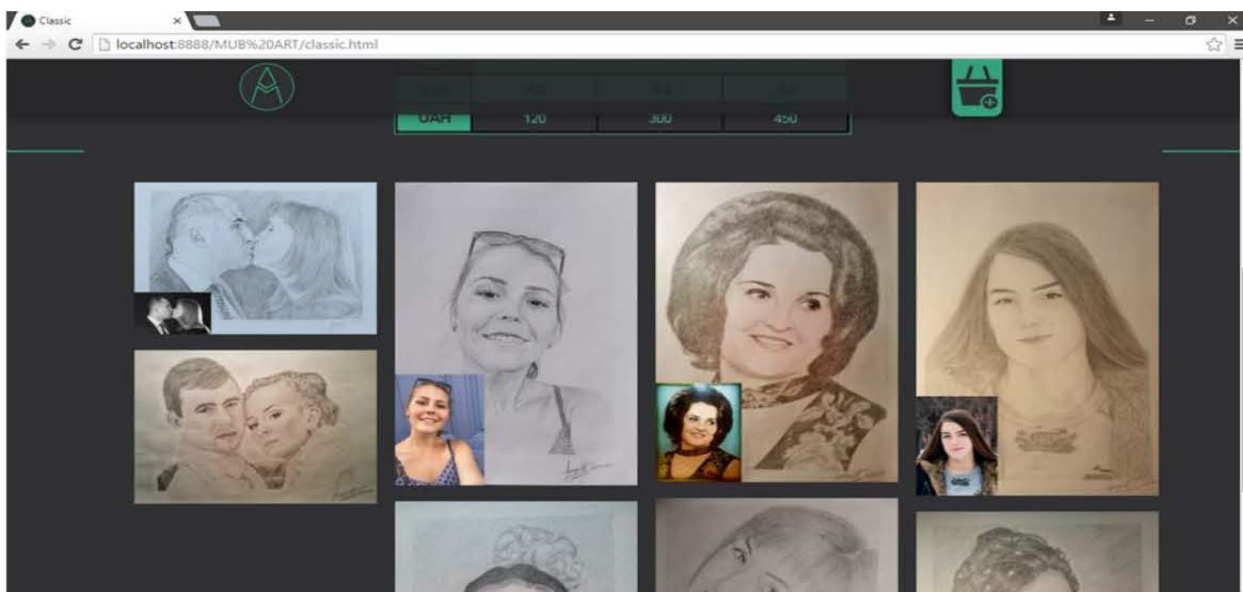


Рисунок 3.7 – Сторінка стилю «Класика» (Частина 2)

Після натискання на кнопку «і» ми отримуємо довгу сторінку з додаткової інформації, переходи на якій можливо також робити за допомогою фіксованої панелі, яка закріплена у верхній частині сторінки. (рис. 3.8). Переходи відбуваються плавно, за допомогою якірних посилань.

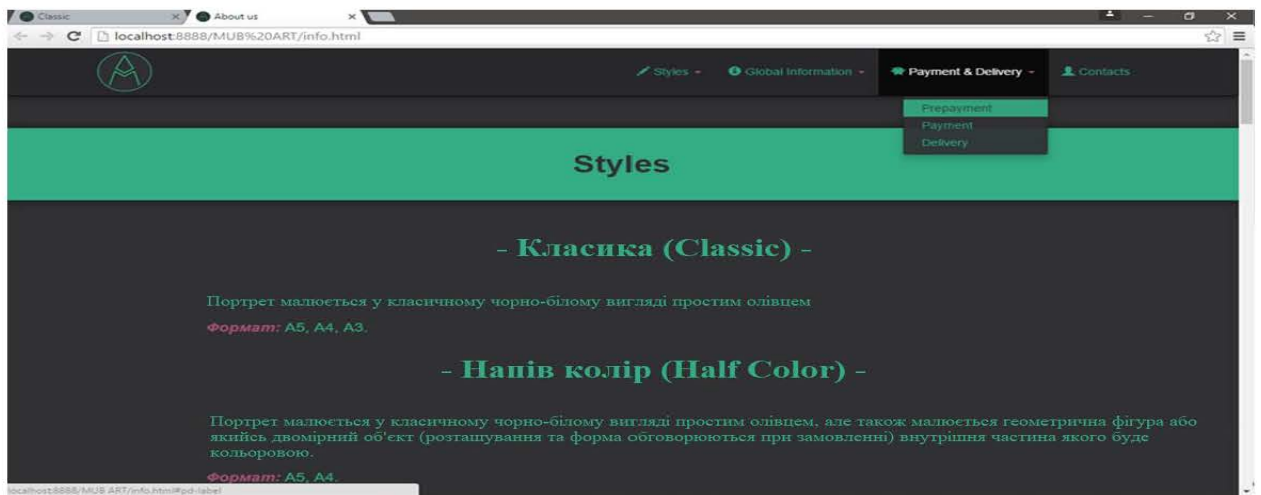


Рисунок 3.8 – Сторінка додаткової інформації

Повертаючись до головної сторінки (рис. 3.1) та при натисканні на головний логотип ми переходимо на сторінку короткої контактної інформації про митця (рис 3.9).

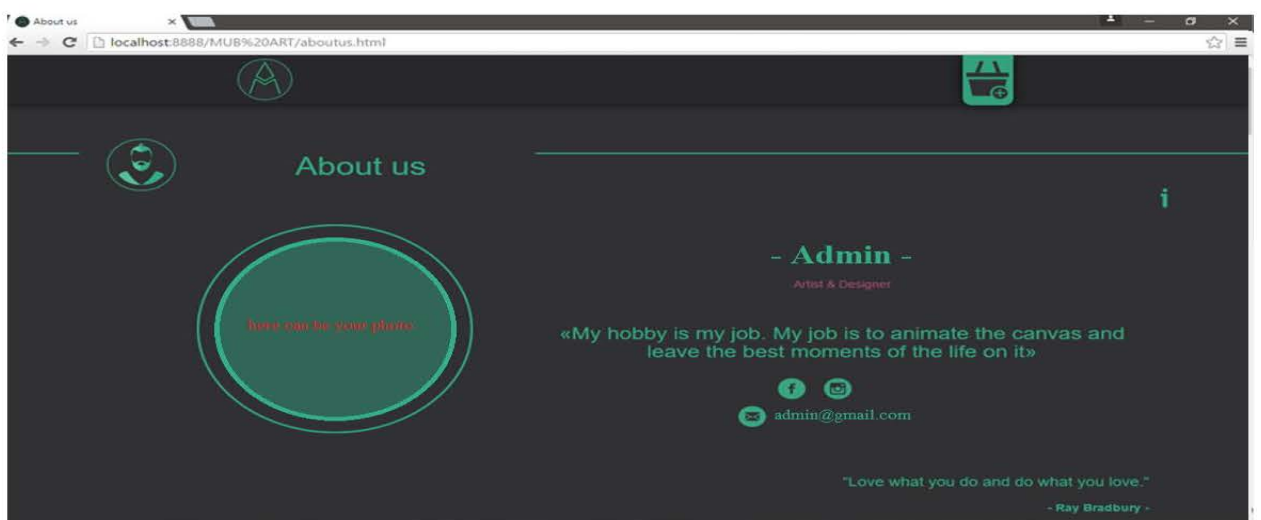


Рисунок 3.9 – Сторінка короткої контактної інформації про митця

3.4.2 Користування сервісом оформлення замовлення

Майже з будь якої сторінки сайту, де є кнопка «Корзина», після натискання, можна перейти на сторінку замовлення, як це було показано на рисунку 3.4, але тільки для авторизованих на сайті користувачів. Якщо користувач не авторизований, то він переходить на сторінку авторизації (рис. 3.2). На цій сторінці також можна побачити ім'я облікового запису, вказаного при реєстрації, користувача, який авторизувався. При натисканні на ім'я надається змога вийти з сайту.

На сторінці замовлення користувач повинен вибрати відповідний стиль, підстиль, формат малюнка та дату, до якої замовлення повинно бути зроблене.

Після цього він переходить на сторінку вітання, де йому дякують за замовлення та повідомляють про те, що замовлення було зроблене і з ним у короткий час зв'яжеться менеджер для перевірки коректності замовлення та повного завершення його оформлення (рис. 3.10).

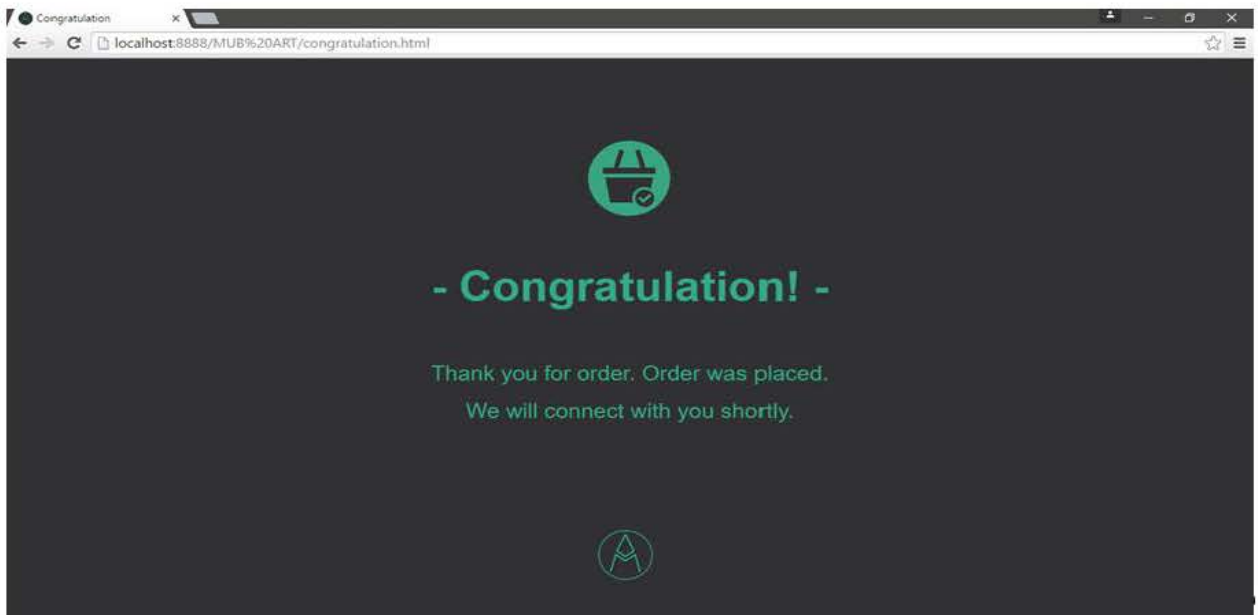


Рисунок 3.10 – Сторінка інформування про успішно зроблене замовлення

3.5 Різниця між профілями

Різниця в користуванні контентом полягає в тому, що адміністратор має право на перегляд даних з усіх таблиць. На головній сторінці у адміністратора додається кнопка, яка дозволяє перейти на сторінку для перегляду даних по таблицям (рис. 3.10 та рис. 3.11).



Рисунок 3.10 – Відображення головного меню при авторизації адміністратора

ID_order	ID_user	ID_pp	deadline_order	orderDate_order
2	9	19	2024-05-08	2024-05-05 20:05:31
3	9	22	2024-05-20	2024-05-05 22:16:10
4	8	16	2024-05-10	2024-05-06 10:02:28
5	9	16	2024-05-14	2024-05-06 12:32:14
6	8	16	2024-05-16	2024-05-06 15:44:21
7	10	19	2024-05-07	2024-05-06 18:10:48
8	9	16	2024-05-25	2024-05-06 23:26:55

ID_pp	ID_picture	ID_price	price_pp
14	18	1	120
15	19	1	300
16	20	1	450
17	21	1	170

Рисунок 3.11 – Відображення усіх даних по таблицям

3.6 Висновки до третього розділу

У даному розділі розглянуто основні аспекти створення інтернет-ресурсу для продажу малюнків. Це включає в себе розробку структури бази даних, обробку запитів до бази даних, розробку інтерфейсу вебзастосунку, визначення стилю та мережевого інтерфейсу, а також програмні механізми, необхідні для правильної роботи системи.

Розділ присвячений розробці структури бази даних для системи. Це включає визначення сутностей, атрибутів, зв'язків між ними, а також розробку схеми бази даних. Правильне проектування бази даних важливо для забезпечення ефективного зберігання та обробки даних системи.

В цьому розділі досліджується різноманітність профілів користувачів системи та їх особливості. Виділення різних типів профілів дозволяє краще враховувати потреби різних груп користувачів і забезпечити їм належний рівень сервісу.

Розділ 3 охоплює процеси реєстрації нових користувачів у системі та авторизації вже зареєстрованих. Це включає в себе збір необхідних даних від користувачів, їх перевірку та зберігання в базі даних, а також надання доступу до особистого кабінету.

Також в цьому розділі розглядається спосіб навігації користувачів системи серед доступного контенту, включаючи пошук, фільтрацію та відображення різних типів інформації з урахуванням вибору користувача.

Розглядається взаємодія користувачів з інтерфейсом для перегляду інформації про доступні малюнки, процес оформлення замовлення на певний малюнок, вибір малюнка, внесення необхідних даних для доставки та оплати, а також підтвердження замовлення.

Загалом, детальний аналіз цих аспектів дозволяє зрозуміти потреби користувачів і розробити ефективну систему, яка відповідає їх вимогам та очікуванням.

ВИСНОВКИ

Метою роботи було створення інтернет-платформи, що надає користувачам можливість перегляду, вибору та придбання малюнків онлайн. Головною метою було створення зручного та ефективного середовища для покупців, де вони зможуть швидко знайти та придбати потрібні малюнки, а також для продавців, щоб вони змогли представляти свої твори та здійснювати продажі через цю платформу.

Основні цілі включають:

1) створення вебсайту з можливістю продажу малюнків, який буде забезпечувати користувачів зручним інтерфейсом для перегляду, вибору та купівлі мистецьких творів.

2) забезпечення привабливого та зручного дизайну вебсайту, що привертає увагу користувачів та забезпечує їм позитивний користувацький досвід.

3) забезпечення високого рівня безпеки для клієнтських даних, включаючи персональну інформацію та фінансові дані, щоб гарантувати конфіденційність та запобігти можливим кібератакам.

4) створення можливостей для розширення бізнесу художників та інших творчих осіб, які продають свої роботи, шляхом підвищення їхньої видимості та доступу до нових ринків.

У першому розділі розглянуто існуючі вебсайти, що спеціалізуються на продажу малюнків та проаналізовано методи їх розробки.

Попит на інтернет-платформи для продажу малюнків є актуальним і постійно зростає. Різноманітність підходів до організації контенту, функціоналу та дизайну наявних сайтів свідчить про широкі можливості у цьому сегменті.

Створення вебсайту для продажу малюнків потребує використання різноманітних технологій та інструментів. Від вибору мов програмування та фреймворків до підбору бази даних та заходів безпеки, кожен етап потребує детального аналізу для вибору оптимального рішення.

Важливими аспектами для вебсайтів, що працюють із значним обсягом контенту та особистих даних користувачів, є продуктивність та безпека. Вибір технологій та методів повинен забезпечити високу швидкість та ефективний захист від потенційних загроз.

У другому розділі описано існуючі програмні інструменти для розробки вебсайту з продажу малюнків.

PHP є потужним інструментом для серверної розробки, що широко використовується для створення динамічного та інтерактивного контенту. Використання PHP дозволяє реалізувати різноманітні функціональні можливості сайту.

JavaScript є ключовою мовою програмування для надання динамічності та інтерактивності на вебсайтах. Вона дозволяє взаємодіяти з користувачем без перезавантаження сторінки та забезпечує більшу функціональність сайту.

OpenServer забезпечує зручне середовище для локальної розробки та тестування вебсайтів, що дозволяє розробникам ефективно працювати над проектом.

MySQL є популярною системою управління базами даних, що забезпечує надійну та ефективну роботу з даними на вебсайтах з продажу малюнків.

AJAX дозволяє виконувати асинхронні запити до сервера без перезавантаження сторінки, що полегшує взаємодію з користувачем та зменшує час завантаження сторінок.

Bootstrap є потужним інструментом для розробки сучасного та адаптивного дизайну вебсайтів. Використання Bootstrap дозволяє ефективно створити стильний та користувацьки зручний інтерфейс.

Загалом, поєднання вищезазначених програмних інструментів дозволяє створити потужний та ефективний вебсайт для продажу малюнків, забезпечуючи високу функціональність, ефективність та зручність для користувачів.

У третьому розділі було розглянуто основні етапи створення інтернет-

платформи для продажу малюнків, що включає розробку структури бази даних, обробку запитів до бази даних, розробку інтерфейсу вебзастосунку, визначення стилю та мережевого інтерфейсу, а також програмні механізми, необхідні для оптимальної роботи системи.

Було спроектовано структуру бази даних для системи, включаючи визначення сутностей, атрибутів, зв'язків між ними. Правильне проектування бази даних є ключовим для забезпечення ефективного зберігання та обробки даних системи.

В третьому розділі було досліджено різноманітність профілів користувачів системи та їх особливості. Виділення різних типів профілів дозволило краще враховувати потреби різних груп користувачів і забезпечило їм належний рівень обслуговування, процеси реєстрації нових користувачів у системі та авторизації вже зареєстрованих.

Також в цьому розділі розглянуто спосіб навігації користувачів системи серед доступного контенту, пошук, фільтрацію та відображення різних типів інформації.

Загалом, детальний аналіз цих аспектів дозволив краще зрозуміти потреби користувачів та розробити ефективну систему, яка відповідає їх вимогам та очікуванням.

Об'єктом дослідження цієї роботи був сайт з продажу малюнків.

Предметом дослідження цієї роботи був процес розробки та реалізації інформаційного сайту з продажу малюнків.

У даному проєкті було вирішено наступні завдання:

1) проведено вибір програмних інструментів для розробки вебсайту для продажу малюнків, що включав аналіз та порівняння різних технологій та інструментів розробки;

2) розроблено та впроваджено базу даних для зберігання інформації про малюнки, включаючи опис структури бази даних та реалізацію запитів до неї;

3) створено інтерфейсну частину вебсайту, що включає розробку користувацького інтерфейсу для реєстрації клієнтів та митців, замовлення

малюнків, перегляду малюнків тощо;

4) розроблено стиль вебсайту з метою створення зручного та привабливого інтерфейсу для користувачів;

5) реалізовано мережевий вебінтерфейс для забезпечення доступу до вебсайту через Інтернет;

6) описано програмні механізми, які використовуються в розробленому вебзастосунку.

Сайт по продажу малюнків має значення як для художників, так і для покупців.

Для художників:

– малюнки можуть бути візуально представлені на сайті, що дозволяє художникам залучити більше уваги та збільшити свою аудиторію;

– продаж малюнків через сайт може стати додатковим джерелом прибутку для художників;

– сайт може надати зручний інтерфейс для завантаження, категоризації та керування малюнками.

Для покупців:

– покупці мають можливість переглядати широкий асортимент малюнків, вибирати та придбавати ті, які вони справді цікавляться;

– покупці можуть бути впевнені в безпеці та надійності операцій, що проводяться через сайт.

Загалом, сайт по продажу малюнків має практичне значення, як для творців, так і для покупців, надаючи їм зручний та ефективний спосіб обміну творчістю та придбання унікальних мистецьких робіт.

Кваліфікаційна робота виконана у відповідності до стандарту спеціальності 121 «Інженерія програмного забезпечення» і демонструє володіння такими компетентностями як:

– здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення;

- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;
- здатність розробляти архітектури, модулі та компоненти програмних систем;
- здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами;
- здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу;
- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;
- здатність до алгоритмічного та логічного мислення тощо.

Серед програмних результатів, визначених стандартом, кваліфікаційна робота реалізовує наступні:

- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- сміти розробляти людино-машинний інтерфейс;
- знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення;
- проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування;

- вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання;
- застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення;
- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і 10 знань;
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення;
- знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних;
- вміти документувати та презентувати результати розробки програмного забезпечення тощо.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) PHP Documentation. URL: <https://www.php.net/>
- 2) Mike McGrath. Php Mysql In Easy Steps, 2018.
- 3) Welling, L., & Thomson, L. PHP and MySQL Web Development. Addison-Wesley, 2016.
- 4) Julie C. Meloni. PHP, MySQL & JavaScript All in One, Sams Teach Yourself. 6th Edition, 2019.
- 5) Eric T. Freeman, Elisabeth Robson. Head First JavaScript Programming, 2017.
- 6) Ethan Brown. Web Development with Node and Express: Leveraging the JavaScript Stack, 2017.
- 7) JavaScript Documentation. URL: <https://javascript.info/>
- 8) OpenServer Documentation. URL: <https://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc35400.1570/pdf/osref.pdf>
- 9) MySQL Documentation. URL <https://www.mysql.com/>
- 10) G. V. Reilly. SQL Queries for Mere Mortals: A Hands-On Guide to Data Manipulation in SQL. Addison-Wesley, 2018.
- 11) Karthik Appigatla. MySQL 8 Cookbook: Over 150 recipes for high-performance database querying and administration, 2014.
- 12) Thomas A. Powell. Ajax: The Complete Reference, 2008.
- 13) Ajax Documentation. URL: <https://api.jquery.com/category/ajax/>
- 14) Bootstrap Documentation. URL: <https://getbootstrap.com/>
- 15) Jacob Lett. Bootstrap 4 Quick Start: A Beginner's Guide to Building Responsive Layouts with Bootstrap 4 (Bootstrap 4 Tutorial, Band 1), 2018.

ДОДАТОК А

Створення файлів CSS

aboutUs.css

```
#my-info-logo {  
    height: 75px;  
    width: 75px;  
    background: url("../svg/logo/my_info_logo.svg");  
    background-repeat: no-repeat;  
}  
#my_info {  
    color: #34aa84;  
    text-align: center;  
    font-weight: 500;  
}  
.text-block {  
    height: 350px;  
    color: #34aa84;  
}  
.text-block-paragraph {  
    margin-top: 40px;  
}  
#text-block-paragraph-fio {  
    text-align: center;  
}  
#text-block-paragraph-speciality {  
    margin-top: 20px;
```



```
    text-align: center;
    color: #a45479;
}
#text-block-paragraph-description {
    margin-top: 50px;
    text-align: center;
}
.text-block-paragraph-ray-bradbury {
    margin-top: 70px;
}
#ray-bradbury-description {
    text-align: right;
}
#ray-bradbury {
    float: right;
}
```

```
.gallery-block {
    width: 400px;
    height: 400px;
    margin: 0 auto;
}
```

```
#myPhoto {
    height: 300px;
    margin-left: 80px;
    margin-top: 20px;
}
/* MEDIA */

@media only screen and (min-width: 1200px) {
    #info-logo {
        margin-right: 6%;
    }
}

@media only screen and (min-width: 992px) and (max-width: 1200px) {
    #info-logo {
        margin-right: 7%;
    }
    #myPhoto {
        height: 250px;
    }
    .text-block-paragraph {
        margin-top: 30px;
    }
    #text-block-paragraph-speciality {
        margin-top: 10px;
    }
    #text-block-paragraph-description {
        margin-top: 20px;
    }
}
```

```
.social {
  display: flex;
  justify-content: center;
  margin-right: 29px;
  margin-top: 15px;
}
.social-fb,
.social-vk,
.social-instagram,
.social-mail {
  width: 30px;
  height: 30px;
  background-repeat: no-repeat;
  float: left;
  margin-left: 20px;
}

#social-mail-text {
  margin-left: 40px;
  margin-top: 8px;
}

.text-block-paragraph-ray-bradbury {
  margin-top: 40px;
}
}

@media only screen and (min-width: 768px) and (max-width: 992px) {
  #my_info {
```

```
    font-size: 27px;
}
#my-info-logo {
    height: 55px;
    width: 55px;
}
#myPhoto {
    height: 250px;
}
#info-logo {
    margin-right: 10%;
}
.text-block-paragraph {
    margin-top: -100px;
}
#text-block-paragraph-fio {
    font-size: 35px;
}
#text-block-paragraph-speciality {
    margin-top: 20px;
}
#text-block-paragraph-description {
    margin-top: 10px;
    font-size: 20px;
    padding: 15px;
}
.social {
    margin-top: 15px;
```

```
        margin-right: 129px;
    }
    #social-mail-text {
        margin-left: 40px;
        font-size: 12px;
    }
    .text-block-paragraph-ray-bradbury {
        margin-top: 10px;
        padding: 15px;
    }

    .gallery-block {
        width: 220px;
        height: 400px;
        margin: 0 auto;
    }
    #myPhoto {
        height: 250px;
        margin-left: -15px;
    }
}

@media only screen and (min-width: 480px) and (max-width: 768px) {
    #my_info {
        font-size: 27px;
    }
    #my-info-logo {
```

```
    height: 55px;
    width: 55px;
}
#myPhoto {
    height: 230px;
}
#info-logo {
    margin-right: 10%;
}
.text-block-paragraph {
    margin-top: -140px;
}
#text-block-paragraph-fio {
    font-size: 35px;
}
#text-block-paragraph-speciality {
    margin-top: 20px;
}
#text-block-paragraph-description {
    margin-top: 10px;
    font-size: 20px;
    padding: 15px;
}
.social {
    margin-top: 15px;
    margin-right: 129px;
}
#social-mail-text {
```



```
    margin-left: 40px;
    font-size: 12px;
}
.text-block-paragraph-ray-bradbury {
    margin-top: 10px;
    padding: 15px;
}

.gallery-block {
    width: 220px;
    height: 400px;
    margin: 0 auto;
}
#myPhoto {
    height: 220px;
    margin-left: 0px;
}
}
@media only screen and (min-width: 320px) and (max-width: 480px) {
    #my_info {
        font-size: 27px;
    }
    #my-info-logo {
        height: 55px;
        width: 55px;
    }
    #info-logo {
        margin-right: 10%;
    }
}
```

```
}  
.text-block-paragraph {  
    margin-top: -140px;  
}  
#text-block-paragraph-fio {  
    font-size: 30px;  
}  
#text-block-paragraph-speciality {  
    margin-top: 20px;  
}  
#text-block-paragraph-description {  
    margin-top: 10px;  
    font-size: 20px;  
    padding: 15px;  
}  
.social {  
    margin-top: 15px;  
    margin-right: 155px;  
    padding: 15px;  
}  
#social-mail-text {  
    margin-left: 40px;  
    font-size: 12px;  
}  
.text-block-paragraph-ray-bradbury {  
    margin-top: 10px;  
    padding: 15px;  
}
```

```
.gallery-block {
  width: 220px;
  height: 400px;
  margin: 0 auto;
}
#myPhoto {
  height: 220px;
  margin-left: 0px;
}
}
@media only screen and (max-width: 320px) {
  #my_info {
    font-size: 26px;
  }
  #my-info-logo {
    height: 55px;
    width: 55px;
  }
  #info-logo {
    margin-right: 10%;
  }
  .text-block-paragraph {
    margin-top: -140px;
  }
  #text-block-paragraph-fio {
    font-size: 30px;
  }
  #text-block-paragraph-speciality {
```

```
    margin-top: 20px;
}
#text-block-paragraph-description {
    margin-top: 10px;
    font-size: 20px;
    padding: 15px;
}
.social {
    margin-top: 15px;
    margin-right: 155px;
    padding: 15px;
}
#social-mail-text {
    margin-left: 40px;
    font-size: 12px;
}
.text-block-paragraph-ray-bradbury {
    margin-top: 10px;
    padding: 15px;
}
.gallery-block {
    width: 220px;
    height: 400px;
    margin: 0 auto;
}

#myPhoto {
    height: 220px;
```

```
    margin-left: 0px;
  }
}
```

global.css

```
a,
a:link,
a:visited,
a:hover,
a:active {
    text-decoration: none;
}
* {
    margin: 0;
    padding: 0;
}
html {
    overflow-x: hidden;
}
body {
    background-color: #313133;
}
.homepage {
    width: 100%;
    margin-left: auto;
    margin-right: auto;
}
```

```
.order-button {
  background-color: #34ac84;
  display: inline-block;
  -moz-border-radius: 0 0 10px 10px;
  -webkit-border-radius: 0 0 10px 10px;
  border-radius: 0 0 10px 10px;
  cursor: pointer;
}
.order-button:hover {
  padding-bottom: 60px;
}
.order-button:hover > .svg-block > .basket {
  transform: translate(0, 60px);
  transition: all 0.20s linear;
}
.transition {
  transition: 0.20s;
}
.shadow {
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.7);
}
.shadow-green {
  box-shadow: 0 0 20px rgba(52, 170, 132, 0.7);
}
.basket {
  width: 55px;
  height: 55px;
  margin: 15%;
```

```
background: url("../svg/basket/basket_logo.svg");
background-repeat: no-repeat;
position: relative;
left: -4px;
}
img {
  pointer-events: none;
}
.back-button {
  height: 60px;
  width: 60px;
  background:
url("../svg/animation/MUB_ART_logo_animation_0_deg_with_border.svg");
  background-repeat: no-repeat;
  position: relative;
  left: -4px;
  border-radius: 100px;
  margin-top: 5px;
  display: inline-block;
}
.back-button:hover {
  background:
url("../svg/animation/MUB_ART_logo_animation_100_deg_with_border.svg");
  transform: rotate(-90deg);
}
.container-navbar {
  background-color: #28282A;
  opacity: 0.93;
  height: 70px;
```



```
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.7);
    position: fixed;
    z-index: 100;
    width: 100%;
    left: 0px;
}
.container-control-block {
    margin-top: 120px;
}
#info-logo {
    cursor: pointer;
    height: 35px;
    width: 35px;
    background: url("../svg/info/info_button_unhover.svg");
    background-repeat: no-repeat;
    float: right;
    margin-right: 15%;
    margin-top: -0.5%;
}
#info-logo:hover {
    background: url("../svg/info/info_button_hover.svg");
}
#info-container-header {
    margin-top: 40px;
}
hr {
    border: none;
    color: #34aa84;
```

```
/* Цвет линии для остальных браузеров */
background-color: #34aa84;
/* Цвет линии для браузера Firefox и Opera */
height: 2px;
/* Толщина линии */
}
.price-block {
    text-align: center;
    position: relative;
    top: 57%;
}
/* MEDIA */
@media only screen and (min-width: 320px) and (max-width: 480px) {
    .back-button {
        height: 50px;
        width: 50px;
    }
    .container-navbar {
        height: 60px;
    }
}
@media only screen and (max-width: 320px) {
    .back-button {
        height: 50px;
        width: 50px;
    }
    .basket {
        width: 50px;
    }
}
```

```
    height: 50px;
  }
  .container-navbar {
    height: 60px;
  }
}
```

ДОДАТОК Б

Код php

```
home.php
<?php
ob_start();
session_start();
include_once("mysql_connect.php");

if (isset($_SESSION['login_user']) != "")
{
    $resultUser = mysql_query("SELECT * FROM User_table WHERE
ID_user=".$_SESSION['login_user']);
    $userRow = mysql_fetch_array($resultUser);
}

if (isset($_SESSION['login_user']) != "" && $userRow['role_user'] == 'admin')
{
    $adminButtonBlock = '<div id="admin-block"> <a href="adminPage.php"
class="admin-button transition shadow"> <span class="input-group-addon
transition"><span class="glyphicon glyphicon-list-alt"></span></span> </a>
</div>';
}
else
{
    $adminButtonBlock = "";
}
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  <title>- MUB ART -</title>
```

```
  <link rel="icon" href="Reference/logo_ico.ico" type="image/x-icon" />
```

```
  <link rel="stylesheet" href="Reference/bootstrap-3.3.7-  
dist/css/bootstrap.min.css">
```

```
  <link rel="stylesheet" href="Reference/css/home.css">
```

```
  <link rel="stylesheet" href="Reference/css/global.css">
```

```
<style>
```

```
  .admin-button {
```

```
    cursor: pointer;
```

```
    width: 55px;
```

```
    height: 55px;
    border-radius: 0 0 10px 10px;
}
.admin-button>.input-group-addon {
    border-radius: 0 0 10px 10px;
}
.admin-button>.input-group-addon>span {
    color: #313133;
}
.input-group-addon {
    padding: 0px 0px;
    font-size: 20px;
    font-weight: normal;
    line-height: 3;
    background-color: #34ac84;
    border: 0px solid #ccc;
}
.input-group-addon:hover {
    background-color: #3FD0A0;
}
</style>
</head>

<body>
    <div id="page-wrapper">
        <div id="header-wrapper">
```

```

<div class="container-fluid">
  <div class="row">
    <div class="col-xs-3 col-sm-3 col-md-3 col-lg-3"><?php echo
$adminButtonBlock; ?></div>
    <div class="col-xs-6 col-sm-6 col-md-6 col-lg-6"></div>
    <div class="col-xs-3 col-sm-3 col-md-3 col-lg-3">
      <div id="order-block">
        <a href="order_with_login.php" class="order-button transition
shadow">
          <div class="svg-block">
            <div class="basket img-responsive"></div>
          </div>
        </a>
      </div>
    </div>
  </div>
</div>
<div id="container-wrapper">
  <div class="container-fluid">
    <div class="row">
      <div class="col-xs-3 col-sm-3 col-md-3 col-lg-3"></div>
      <div class="col-xs-6 col-sm-6 col-md-6 col-lg-6">
        <div id="menu-block">
          <div id="radial_container">
            <ul class="list">
              <li class="item">
                <a href="dots.html">
                  <div class="my_class img1 transition">

```



```
<p id="position_img1">Dots</p>
</div>
</a>
</li>
<li class="item">
  <a href="negative.html">
    <div class="my_class img2 transition">
      <p id="position_img2">Negative</p>
    </div>
  </a>
</li>
<li class="item">
  <a href="vectorGraphic.html">
    <div class="my_class img3 transition">
      <p id="position_img3">Vector graphic</p>
    </div>
  </a>
</li>
<li class="item">
  <a href="sketchesForTattoo.html">
    <div class="my_class img4 transition">
      <p id="position_img4">Sketches for tattoo</p>
    </div>
  </a>
</li>
<li class="item">
  <a href="classic.html">
    <div class="my_class img5 transition">
```

```

        <p id="position_img5">Classic</p>
    </div>
</a>
</li>
<li class="item">
    <a href="halfColor.html">
        <div class="my_class img6 transition">
            <p id="position_img6">Half color</p>
        </div>
    </a>
</li>
</ul>
</div>
<a href="aboutus.html">
    <div id="big" class="my_class transition"></div>
</a>
</div>
</div>
<div class="col-xs-3 col-sm-3 col-md-3 col-lg-3"></div>
</div>
</div>
<!-- JS start -->
<script src="Reference/js/jquery.min.js"></script>
<script type="text/javascript" src="Reference/bootstrap-3.3.7-
dist/js/bootstrap.min.js"></script>
<!--    <script type="text/javascript"
src="templates/pcv/js/pcvector.js"></script>-->
<script

```

```
src="https://cdnjs.cloudflare.com/ajax/libs/prefixfree/1.0.7/prefixfree.min.js"></script>
```

```
<script type="text/javascript"
src="Reference/js/jquery.radmenu.min.js"></script>
```

```
<script type="text/javascript"
src="Reference/js/radial_container_params.js"></script>
```

```
<!-- JS end -->
```

```
<!-- <script type="text/javascript"
src="Reference/js/prohibitions.js"></script> -->
```

```
</div>
```

```
</body>
```

```
</html>
```

```
<?php ob_end_flush(); ?>
```

```
sign-in_page.php
```

```
<?php
```

```
ob_start();
```

```
session_start();
```

```
include_once("mysql_connect.php");
```

```
if ( isset($_SESSION['login_user']) != "" ) {
```

```
header("Location: order_with_login.php");
```

```
exit;
```

```
}
```

```
$error = false;
```

```
// Метод шифровки
```

```
function decryptIt( $q ) {
```

```

    $cryptKey = 'iGG0rGtIn5UB1xG03efyCi';

    $qDecoded = rtrim( mdecrypt_decrypt( MCRYPT_RIJNDAEL_256,
md5( $cryptKey ), base64_decode( $q ), MCRYPT_MODE_CBC, md5( md5(
$cryptKey ) ) ), "\0");

    return( $qDecoded );
}

```

// Метод дешифровки

```

function encryptIt( $q ) {

    $cryptKey = 'iGG0rGtIn5UB1xG03efyCi';

    $qEncoded = base64_encode( mencrypt_encrypt(
MCRYPT_RIJNDAEL_256, md5( $cryptKey ), $q, MCRYPT_MODE_CBC,
md5( md5( $cryptKey ) ) ) );

    return( $qEncoded );

}

```

```

if(isset($_POST['submit'])) {

```

```

    $emailLogin = trim($_POST['emailLogin']);
    $emailLogin = strip_tags($emailLogin);
    $emailLogin = htmlspecialchars($emailLogin);

    $pass = trim($_POST['pass']);
    $pass = strip_tags($pass);
    $pass = htmlspecialchars($pass);

```

// Проверка входных данных

```

        if(empty($emailLogin)){
            $error = true;
            $emailLoginError = "Please enter your Email address or
Username.";
        }

        if(empty($pass)){
            $error = true;
            $passError = "Please enter your Password.";
        }

// Если ошибок нет
        if (!$error) {

            $password = encryptIt($pass);

            $result = mysql_query("SELECT ID_user, login_user,
password_user FROM User_table WHERE email_user='$emailLogin'");
            $row = mysql_fetch_array($result);
            $count = mysql_num_rows($result);

            $result2 = mysql_query("SELECT ID_user, login_user, password_user
FROM User_table WHERE login_user='$emailLogin'");
            $row2 = mysql_fetch_array($result2);
            $count2 = mysql_num_rows($result2);

            if( $count == 1 && $row['password_user'] == $password )
        {

```

```

        $_SESSION['login_user'] = $row['ID_user'];
        header("Location: order_with_login.php");

    }

    else if ( $count2 == 1 && $row2['password_user'] == $password )
    {

        $_SESSION['login_user'] = $row2['ID_user'];
        header("Location: order_with_login.php");

    }

    else {

        $errMSG = "Incorrect credentials!";

    }

}

?>

<!DOCTYPE html>
<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Sign in</title>
    <link rel="icon" href="Reference/logo_ico.ico" type="image/x-icon" />
    <link rel="stylesheet" href="Reference/bootstrap-3.3.7-
dist/css/bootstrap.min.css" type="text/css" />
    <link rel="stylesheet" href="Reference/css/global.css">
    <link rel="stylesheet" href="Reference/css/sign-in.css"/>
</head>

```

```

<body>

<div id="page-wrapper">
  <div id="header-wrapper">
    <div class="container-fluid">
      <div class="container-navbar">
        <div class="row">
          <div class="col-xs-9 col-sm-9 col-md-9 col-lg-9">
            <div class="row">
              <div class="col-xs-3 col-sm-3 col-md-3 col-lg-3"></div>
              <div class="col-xs-9 col-sm-9 col-md-9 col-lg-9">
                <div id="back-block">
                  <a href="home.php">
                    <div class="back-button transition"></div>
                  </a>
                </div>
              </div>
            </div>
          </div>
          <div class="col-xs-3 col-sm-3 col-md-3 col-lg-3"></div>
        </div>
      </div>
    </div>
  </div>
</div>

<div id="container-wrapper">

```



```

<div class="container-fluid">
  <div class="row">
    <div class="col-sm-4 col-md-4 col-lg-4"></div>
    <div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
      <div class="row">
        <div class="col-sm-1 col-md-3 col-lg-2"></div>
        <div class="col-xs-12 col-sm-10 col-md-8 col-lg-8">
          <div id="login-form">
            <form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>" autocomplete="off">

              <h2 class="login-form-text">Sign In</h2>

              <hr />

              <?php if ( isset($errMSG) ) { ?>
                <div class="login-form-error">
                  <div class="alert alert-danger">
                    <span class="glyphicon glyphicon-info-
sign"></span>
                    <?php echo $errMSG; ?>
                  </div>
                </div>
              </div>
              <?php } ?>

            <div class="login-form-email">
              <div class="input-group">

```

```

        <span class="input-group-addon"><span
class="glyphicon glyphicon-user"></span></span>

        <input type="text" name="emailLogin" class="form-
control" placeholder="Your Email / Username" value="<?php echo $emailLogin;
?>" maxlength="30" />

    </div>

    <span class="text-danger"><?php echo
$emailLoginError; ?></span>
</div>

<div class="login-form-password">
    <div class="input-group">
        <span class="input-group-addon"><span
class="glyphicon glyphicon-lock"></span></span>

        <input type="password" name="pass" class="form-
control" placeholder="Your Password" maxlength="15" />
    </div>

    <span class="text-danger"><?php echo $passError;
?></span>
</div>

<hr />

<div class="login-form-button">
    <button type="submit" class="sign-in-btn-primary"
name="submit">Sign In</button>
</div>

<hr />

<div class="login-form-sign-up">

```

```
        <a href="sign-up_page.php">Sign Up</a>
    </div>

    </form>
</div>

</div>
<div class="        col-sm-1 col-md-3 col-lg-2"></div>

</div>
</div>
<div class="        col-sm-4 col-md-4 col-lg-4"></div>
</div>
</div>
</div>
</div>
</body>

</html>
<?php ob_end_flush(); ?>
```