

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СІПЕНКО

«____» _____ 2021 р.

Дипломний проект

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Комп'ютерні системи та мережі»
спеціальності 123 «Комп'ютерна інженерія»**

на тему: Веб-сервіс для продажу картин та аксесуарів

Виконав: студент 4 курсу, групи ІВ-73

Андрієнко Валерій Романович _____

Керівник професор, д.т.н., професор Луцький Г.М. _____

Консультант (нормконтроль):

Професор, доктор технічних наук

Сімоненко Валерій Павлович _____

Рецензент _____

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

«Комп'ютерні системи та мережі»

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій СТИПЕНКО

«__» _____ 2021 р.

ЗАВДАННЯ

на бакалаврський дипломний проект студента

Андрієнка Валерія Романовича

1. Тема проекту Веб-сервіс для продажу картин та аксесуарів
керівник проекту професор Луцький Георгій Михайлович, професор, д.т.н.
Затверджені наказом по університету від «11» __ 05 __ 2021 року № 1139-с
2. Термін здачі студентом закінченого проекту (роботи) 1.06.2021
3. Вихідні дані до проекту технічна документація, теоретичні та статистичні дані;
4. Зміст розрахунково-пояснювальної записки аналіз архітектурних елементів, проектування веб-сервісу, розробка веб-сервісу
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): схема додавання товару в кошик, схема оформлення замовлення, структурна схема веб-сервісу.

6. Консультанти проекту, з вказівкою розділів проекту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормконтроль	д.т.н., проф. Сімоненко В.П.		

7. Дата видачі завдання 20.09.2020

Календарний план

№ п/п	Найменування етапів дипломного проекту	Терміни виконання етапів проекту	Примітки
1.	<i>Затвердження теми проекту</i>	<i>10.12.2020</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>14.12.2020-13.03.2021</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>13.03-26.03.2021</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>26.03.2021-05.04.2021</i>	
5.	<i>Програмна реалізація системи</i>	<i>05.04.2021-17.04.2021</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>17.04.2021-21.05.2021</i>	
7.	<i>Захист програмного продукту</i>	<i>26.04.2021</i>	
8.	<i>Передзахист</i>	<i>24.05.2021</i>	
9.	<i>Захист</i>	<i>15.06.2021</i>	

Студент

Андрієнко В.Р.

Керівник проекту

Луцький Г.М.

Анотація

Ця дипломна робота присвячена створенню веб-сервісу з продажу картин та аксесуарів, що призначений для певної цільової аудиторії.

Для створення веб-сервісу використовується програмне середовище VSCode.

Аннотация

Данная дипломная работа посвящена созданию веб-сервиса по продаже картин и аксессуаров, который предназначен для определенной целевой аудитории.

Для создания веб-сервиса используется программная среда VSCode.

Annotation

This thesis is devoted to the creation of a web service for the sale of paintings and accessories, designed for a specific target audience.

The VSCode software environment is used to create a web service.

Опис альбому

до дипломного проекту

на тему: «Веб-сервіс для продажу картин та аксесуарів»

[illegible]

Технічне завдання до дипломного проекту

на тему: «Веб-сервіс для продажу картин та аксесуарів»

Зміст

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ РОЗРОБКИ	2
3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1 Вимоги до розробленого продукту	3
5.2 Вимоги до програмного забезпечення	3
6. ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ.467800.002 ТЗ			
Зм	Арк.	№ докум	Підпис	Дата				
Розроб.		Андрієнко В.Р.			Веб-сервіс для продажу картин та аксесуарів Технічне завдання			
Первір.		Луцький Г.М.						
Н.Контр.		Сімоненко В.П.						
Затверд.								
					Літ.	Арк.	Аркушів	
							1	3
					КПІ ім. Ігоря Сікорського ФІОТ, ІВ-73			

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування: розробка веб-сервісу для продажу картин та аксесуарів.

Область застосування: повсякденне життя, організації та компанії.

2. ПІДСТАВИ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи кваліфікаційно-освітнього рівня «бакалавр комп'ютерної інженерії», затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут».

3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є розробка веб-сервісу для продажу картин та аксесуарів який буде доступний і зрозумілий для всіх користувачів.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література, публікації в мережі Інтернет з даних питань, довідники з програмування.

					<i>ІАЛЦ.467800.002 ТЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до розробленого продукту

- Простий і зрозумілий інтерфейс веб-сервіса.
- Актуальна інформація.

5.2 Вимоги до програмного забезпечення

- Windows/Linux/macOS
- Любий Браузер

6. ЕТАПИ РОЗРОБКИ

	Дата
Вивчення літератури	11.12.2020
Складання і узгодження технічного завдання	16.01.2021
Аналіз структури програмного забезпечення	15.02.2021
Створення модулів розроблюваної системи	05.04.2021
Тестування окремих модулів системи	29.05.2021
Доопрацювання, налагодження і виправлення помилок	30.05.2021
Оформлення документації дипломної роботи	03.06.2021

					ІАЛЦ.467800.002 ТЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Пояснювальна записка

до дипломного проекту

на тему: «Веб-сервіс для продажу картин та аксесуарів»

Зміст

ВСТУП	3
РОЗДІЛ 1	6
АНАЛІЗ АРХІТЕКТУРНИХ ЕЛЕМЕНТІВ ВЕБ-САЙТІВ	6
1.1 Загальні відомості про веб-сайти	6
1.2 Веб-сервіси. Переваги та Недоліки	9
1.3 Платформи для розробки веб-сервісів.	13
1.4 Огляд аналогів веб-сервісів з продажу картин.	15
Висновок до Розділу 1	17
РОЗДІЛ 2	18
ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ДЛЯ ПРОДАЖУ КАРТИН ТА АКСЕСУАРІВ	18
2.1 Опис функціоналу сервісу	18
2.2 Порівняльний аналіз програм для розробки веб-сервісів	18
2.2.1 Порівняння PHP та Python	18
2.2.2 Порівняння JavaScript та Java	22
2.2.3 Вибір між C # та C ++ для веб-розробки	24
2.2.4 Сучасні бази даних: призначення, переваги і недоліки	25
Висновок до Розділу 2	31
РОЗДІЛ 3	32
РОЗРОБКА ВЕБ-СЕРВІСУ	32
3.1 Вибір програми	32
3.2 Складова HTML документа	34

					ІАЛЦ.467800.003 ПЗ			
Зм	Арк.	№ докум	Підпис	Дата	Веб-сервіс для продажу картин та аксесуарів Пояснювальна записка	Літ.	Арк.	Аркушів
Розроб.	Андрієнко В.Р.						1	56
Первір.	Луцький Г.М.							
Н.Контр.	Сімоненко В.П.					КПІ ім. Ігоря Сікорського ФІОТ, ІВ-73		
Затверд.								

3.3 Розробка веб-сторінки.....	35
3.3.1 Розробка першої веб-сторінки.....	35
3.3.2 Розробка другої веб-сторінки.....	39
3.3.3 Розробка третьої веб-сторінки.....	41
3.3.4 Розробка четвертої веб-сторінки.....	42
3.3.5 Розробка п'ятої веб-сторінки.....	43
3.3.6 Розробка шостої веб-сторінки.....	49
Висновок до Розділу 3.....	53
Висновки.....	54
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	55

					ІАЛЦ.467800.003 ПЗ			
Зм	Арк.	№ докум	Підпис	Дата				
Розроб.		Андрієнко В.Р.			Веб-сервіс для продажу картин та аксесуарів Пояснювальна записка			
Первір.		Луцький Г.М.						
Н.Контр.		Сімоненко В.П.						
Затверд.								
					Літ.	Арк.	Аркушів	
							2	59
					КПІ ім. Ігоря Сікорського ФІОТ, ІВ-73			

ВСТУП

Тенденції сучасного світу перебувають у постійному потоці та безупинному розвитку. Найбільш вагомим ресурсом стає час, саме тому всі життєві процеси прискорюються, а разом із ними і технології. Все частіше основні запити людини припиняють обмежуватись стандартними засобами здійснення у фізичному стані, що потребує великі витрати як у часовому аспекті, так і у кількісному по відношенню до дій. Враховуючи специфіку темпоральності, можна простежити явну перевагу технологічного чиннику над фізичним, що передбачає зріст актуальності використання веб-сервісів. Сучасна людина звикла до прискореного темпу життя, тому й передбачає, що використання веб-сервісів буде відповідати її потребам. Таким чином, створення інтернет-магазинів є одним із найзручніших та найвигідніших варіантів електронної комерції, що забезпечує не лише успіх бізнесу, при правильному позиціонуванні, але й зручність та вигідність для користувачів.

Виходячи з реалій сучасного світу, електронна торгівля займає міцну комерційну позицію завдяки своїй зручності, швидкості використання та надійності. Потрібно звернути увагу на той факт, що вдалий варіант інтернет-магазину визначається його інтерактивністю та доступністю. Окрім того, важливим аспектом є тотожність до зручності фізичного відвідування магазину, аби клієнт відчував себе так само комфортно, як при живому огляді товару.

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Таким чином, створення якісного концепту інтернет-магазину може забезпечити приріст кількості потенційних клієнтів фірми та покращення становища власного бізнесу. Основними критеріями створення вдалого веб-сервісу можна виділити зручність та простоту його інтерфейсу. Юзабіліті, як спосіб можливостей та зручності використання передбачає створення відповідних умов для користувачів у орієнтуванні, що підкріплюються швидкістю, інтуїтивністю, зручністю, простотою та широким спектром можливостей. Таким чином, потрібно враховувати досвід користувача, та спиратися на його логіку, аби функціонал сайту міг відповідати його можливостям та потребам.

Завдання дипломної роботи полягає у створенні сайту з продажу товарів мистецького характеру, зокрема картин та аксесуарів. Структура сайту передбачає три сторінки: головна, про нас, товари, а також розділ з контактами, що розташований у нижній частині сайту та кошик з обраними продуктами. Головна сторінка наповнена основною інформацією та зображенням прикладів товару, для привернення уваги клієнтів. Сторінка про нас розкриває основні факти про фірму, її переваги та надійність. Сторінка товарів виконана в якості електронної вітрини, що забезпечує огляд товарів та їх упорядкування за допомогою спеціальних розділів та підрозділів. Розділ контактів поєднаний із основними запитами адміністративного характеру. Кошик транслює кількість обраних товарів з їх зображеннями та загальною сумою. Обладнаний кнопкою переходу до сторінки оформлення замовлення. Загальний вигляд веб-сервісу виконаний у мінімалістичному стилі з ухилом на простоті та зручності.

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

Враховуючи інтенсифікацію розвитку електронної торгівлі та постійне збільшення інтернет-магазинів у мережі, виникає потреба у виділенні серед конкурентів та створенні привабливих умов для збільшення потенційних клієнтів. Більша частина сайтів побудована за принципом досвіду користувача, тому більшість основних елементів повторюються, згідно зі звичним форматом їх використання. Проте, багато сайтів не нехтують заспамленістю рекламою та перенавантаженням контенту. Так само більшість досі використовує окремий розділ реєстрації нового клієнта, що забирає багато часу та може привести до втрати клієнта. Саме тому, мета цієї дипломної роботи полягає у розробці максимально зручного веб-сервісу з продажу картин та аксесуарів, переваги якого будуть зосереджуватись на уникненні нашарування рекламою та відмові від окремого пункту реєстрації нових користувачів магазину. Також важливим пунктом є забезпечення можливості для використання осіб із різним досвідом застосування веб-сервісів.

Для ефективного дослідження було розглянуто програми за допомогою яких розробляються веб-сайти. Окремо було проведено їх порівняльну аналітику між ними

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
						5
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 1

АНАЛІЗ АРХІТЕКТУРНИХ ЕЛЕМЕНТІВ ВЕБ-САЙТІВ

1.1 Загальні відомості про веб-сайти

Для початку почнемо з того, що таке веб-сайт? Веб-сайт - це сукупність логічно зв'язаної гіпертекстової інформації, оформленої у вигляді окремих сторінок і доступної в мережі Інтернет. Таке визначення веб-сайту було правильним на початку існування Інтернету, коли Мережа і веб-сайти використовувалися в основному як система для розваг. До кінця 90-х років веб-сервіси були в основному статичними сторінками (складаються з незмінних сторінок). Для створення веб-сайту потрібно було знати лише мову гіпертекстової розмітки — HTML (*HyperText Markup Language*). Про зручність і красу тогочасних веб-сайтів взагалі особливо не доводилося говорити. Час іде, розвиваються мови програмування, розширюються канали передачі інформації. Сьогодні мільйони людей вранці встають і йдуть в Інтернет, і там проводять цілий день.

Веб-сайт, як система електронних документів може належати як організації, так і приватній особі, і може бути доступним в комп'ютерній мережі та IP-адресі (Internet Protocol – міжмережний протокол). Поділ даних на пакети та їх збирання у пункті призначення здійснюється під керуванням протоколу TCP (Transmission Control Protocol). У Інтернеті використовується велика кількість протоколів,

					ІАЛЦ.467800.003 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

завдяки чому існує широкий спектр служб, які надаються та підтримуються за допомогою цієї глобальної мережі. Найпопулярнішою зі служб є Всесвітня павутина (World Wide Web - WWW). Це інформаційна мультимедійна система, яка об'єднує в єдиному просторі інформацію різних типів.

HTTP (Hyper Text Transfer Protocol) – протокол прикладного рівня, тобто це взаємодія мережі та користувача, передає данні у вигляді гіпертекстових документів у форматі HTML.

URL (Uniform Resource Locator – єдиний вказівник на ресурс) - стандартизована адреса певного ресурсу, такого як документ чи зображення.

Типовий URL має такий вигляд: протокол://адреса, де протокол визначає метод доступу до ресурсу, наприклад http, ftp; адреса описує місце розташування ресурсу і включає назву сервера, шлях до документа і його ім'я.

Сторінки сайтів – набір текстових файлів, розмічених на мові HTML. Ці файли, будучи завантаженими на комп'ютер користувача, розпізнаються і оброблюються браузером і виводяться на монітор, екран телефону тощо. Мова HTML дозволяє формувати текст, виділяти в ньому функціональні елементи, створювати гіпертекстові посилання і вставляти на відображену сторінку зображення, відео та аудіозаписи та будь-які інші мультимедійні елементи. Відображення сторінки змінюється за допомогою додавання стилів, написаних на мові CSS, що дозволяє змінити розмір, колір блоків, шрифт тексту та інше, а також шляхом додавання сценаріїв створених на мові JavaScript.

					ІАЛЦ.467800.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

Сайти мають підрозділи, які орієнтовані на ту чи іншу аудиторію, такі розділи називають версіями сайту. Існують мобільні версії сайтів які призначенні для користування за допомогою портативних пристроїв, і деяка аудиторія взагалі не має комп'ютера і вони можуть зайти тільки з портативного пристрою. Такі версії сайту потребують гнучку вертку сайту, яка налаштовується під розмір екрану, тип пристрою, іншого дизайну, тощо. Такий підхід дозволяє користувачу з комфортом працювати в Інтернеті [1].

Зараз розглянемо класифікацію інтернет-ресурсів. Сайти відрізняються:

1) За способом доступу:

- Відкритий доступ
- Обмежений доступ (комерційні, корпоративні, особисті обмеження)

2) Стосовно власника ресурсу до авторства:

- Власного авторства
- Авторства третьої сторони

3) За способом виробництва ресурсу:

- Створені в цифровому вигляді
- Оцифровані (власником ресурсу або третьою стороною)

4) За розміщенням:

- Автономні сайти
- Розділи сайтів

5) За складом:

- Моноресурси
- Колекції

					ІАЛЦ.467800.003 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

6) По регулярності поновлення:

- Періодично оновлювані
- Нерегулярно оновлювані
- Неоновлювані

7) За способом передачі інформації:

- Текст
- Відео
- Аудіо
- Графіка
- Мультимедіа
- Квест

8) За способом генерації інформації:

- Статичні
- Динамічні

1.2 Веб-сервіси. Переваги та Недоліки

1) Переваги:

Веб-служби пропонують багато переваг перед іншими типами розподілених обчислювальних архітектур.

- Сумісність - це найважливіша перевага веб-служб. Веб-служби, як правило, працюють поза приватними мережами, пропонуючи

					ІАЛЦ.467800.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

розробникам власний шлях до своїх рішень. Отже, розвинені послуги, ймовірно, матимуть триваліший термін життя, пропонуючи кращу віддачу від інвестицій у розроблену послугу. Веб-служби також дозволяють розробникам використовувати бажані мови програмування. Крім того, завдяки використанню стандартних методів комунікації, веб-служби фактично не залежать від платформи.

- Зручність використання - веб-служби дозволяють діловій логіці багатьох різних систем виставляти інформацію через Інтернет. Це надає вашим додаткам свободу вибору веб-служб, які їм потрібні. Замість того, щоб повторно винаходити колесо для кожного клієнта, вам потрібно лише включити додаткову бізнес-логіку, що стосується конкретних додатків, на стороні клієнта. Це дозволяє розробляти послуги та / або код на стороні клієнта, використовуючи потрібні мови та інструменти.
- Багаторазове використання - Веб-служби надають не компонентну модель розробки додатків, а найближче до нульового кодування розгортання таких служб. Це полегшує повторне використання компонентів веб-служби, як це підходить для інших служб. Це також полегшує розгортання застарілого коду як веб-служби.
- Можливість розгортання - веб-служби розгортаються за стандартними Інтернет-технологіями. Це дає можливість розгортати веб-служби навіть через протипожежну стіну на серверах, що працюють в Інтернеті на іншій стороні земної кулі.

					ІАЛЦ.467800.003 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Також завдяки використанню перевірених стандартів спільноти, основна безпека (наприклад SSL) вже вбудована.

2) Недоліки:

- Хоча простота веб-служб є певною перевагою, але це також може бути перешкодою. Веб-служби використовують протоколи простого тексту, які використовують досить детальний метод для ідентифікації даних. Це означає, що запити веб-служб більші за запити, кодовані за допомогою двійкового протоколу. Надмірний розмір насправді є проблемою лише для низькошвидкісних з'єднань або надзвичайно зайнятих з'єднань.
- Хоча HTTP і HTTPS (основні веб-протоколи) прості, вони насправді не були призначені для довготривалих сесій. Як правило, браузер встановлює з'єднання HTTP, запитує веб-сторінку і, можливо, деякі зображення, а потім відключається. У типовому середовищі CORBA або RMI клієнт підключається до сервера і може залишатися на зв'язку протягом тривалого періоду часу. Сервер може періодично надсилати дані клієнту. Цей тип взаємодії важкий із веб-службами, і вам потрібно трохи попрацювати, щоб компенсувати те, що HTTP не робить для вас.
- Проблема HTTP і HTTPS, коли мова заходить про веб-служби, полягає в тому, що ці протоколи не мають статусу держави - взаємодія між сервером і клієнтом, як правило, коротка, і коли

					ІАЛЦ.467800.003 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

немає даних, що обмінюються, сервер і клієнт не знають один про одного . Більш конкретно, якщо клієнт робить запит на сервер, отримує деяку інформацію, а потім негайно припиняє роботу через відключення електроенергії, сервер ніколи не знає, що клієнт більше не активний. Серверу потрібен спосіб відстежувати, що робить клієнт, а також визначити, коли клієнт більше не активний.

- Як правило, сервер надсилає клієнту якусь ідентифікацію сеансу, коли клієнт вперше звертається до сервера. Потім клієнт використовує цю ідентифікацію, коли робить подальші запити на сервер. Це дозволяє серверу згадувати будь-яку інформацію про клієнта. Сервер зазвичай повинен покладатися на механізм очікування, щоб визначити, що клієнт більше не активний. Якщо сервер не отримує запит від клієнта через заздалегідь визначений проміжок часу, він припускає, що клієнт неактивний і видаляє будь-яку інформацію про клієнта, яку він зберігав. Ці додаткові накладні витрати означають більше роботи для розробників веб-служб [2].

Веб-сервіси розділяються на велику кількість підтипів:

- Пошукові сервіси (Goggle, Yandex, Yahoo!)
- Поштові сервіси (Gmail, ukr.net, yandex.ru)
- Веб-форуми – сервіси для користувачів які можуть розміщувати посилання на будь-яку вподобану інформацію в інтернеті і обговорювати її (Наприклад, найпопулярніший із таких сервісів є reddit)

					ІАЛЦ.467800.003 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

- Хмарне сховище даних – сервіс який дозволяє зберігати ваші дані, виконувати операції над ними, тощо(Наприклад: Google Drive)
- Сервіс редагування ваших документів (Google Docs)

1.3 Платформи для розробки веб-сервісів.

- Google App Engine(служба хостингу сайтів та web-додатків на серверах Google з безкоштовним ім'ям <ім'я_сайта> .appspot.com або з власним ім'ям, задіяним за допомогою служб Google)
- DO App Platform
- Appfleet
- Platform.sh
- Jelastic
- Azure App Service
- Heroku Platform
- AWS Elastic Beanstalk

					ІАЛЦ.467800.003 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

- AngularJS (потужна платформа для розробки на основі JavaScript, розроблена Google і вперше випущена в 2009 році. Ця мережа з відкритим кодом, як відомо, використовується такими технічними гігантами, як Netflix, PayPal, Upwork та іншими.)
- Asp.Net (ідеальна платформа для створення динамічних та масштабованих настільних додатків. Побудована на Common Language Runtime (CLR), платформа полегшує написання коду під час його розгортання.)
- Node.js (потужна платформа на основі Java-сценарію. Він пропонує повне середовище JavaScript, ідеально підходить для розробників, яким потрібно створювати масштабовані та швидкі мережеві програми, здатні обробляти велику кількість одночасних з'єднань.)
- PHP (одна з найпопулярніших мов з відкритим кодом, яка широко використовується для розробки веб-додатків. Завдяки підвищеній безпеці, швидшому виконанню та доступності. Завдяки його численним перевагам провідні онлайн-компанії, такі як Yahoo, Facebook, Flickr, Wikipedia, використовують цю мову сценаріїв.)
- VSCode (редактор вихідного коду, розроблений Microsoft для Windows, Linux і macOS. Позиціонується як «легкий» редактор коду для кроссплатформенної розробки веб і хмарних додатків. Включає в себе відладчик, інструменти для роботи з Git, підсвічування синтаксису, IntelliSense і засоби для рефакторинга. Має широкі можливості для кастомізації: поєднання клавіш і файли конфігурації.) [3].

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

1.4 Огляд аналогів веб-сервісів з продажу картин

1.4.1 ARTFACTOR

Цей веб-сервіс був створений для продажу картин. На рисунку (рис.1.1) ми бачимо, що сайт має 7 розділів(галерея, колекції і так далі), а ці розділи мають підрозділи. Також на головному екрані є анімація, що додає унікальності сайту. Головна сторінка має містити банер актуальних товарів, які б привертали увагу користувача та сприяли прискоренню його рішення до здійснення покупки. Товар повинен бути у швидкому доступі до погляду користувача, а також містити загальну інформацію та характеристики продукту. Особливо важливим елементом інтернет-магазинів є кнопки, що спонукають користувача до дій, тобто направляють його та підказують варіанти його подальших дій. Таким чином, розробники сайту забезпечують зручність та інтуїтивність в орієнтування своїм клієнтам.

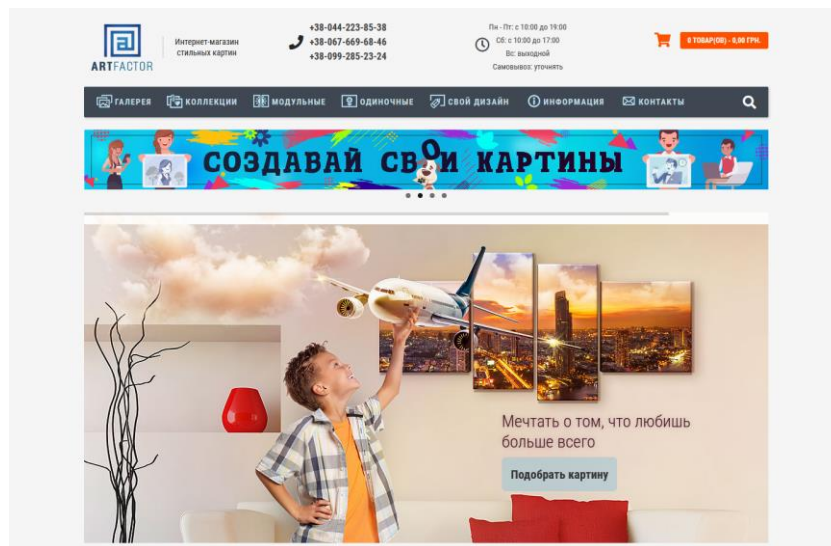


Рис. 1.1 - Головна сторінка

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

15

Також цей веб-сервіс пропонує оглянути на головній сторінці варіації розділів та обрати більш підходящий для користувача. Також у користувача є можливість регулювати контент по розділам за тематикою популярністю та новизною. Обов'язковим елементом всіх інтернет магазинів є корзина, яка стає перехідним етапом до розділу оформлення замовлення товару. Інтернет магазин має гарантувати надійність здійснення оплати та умови доставки що прописані в окремому пункті.

1.4.2 Rozetka

Є можливість залишати відгуки про товари. Саме за ними користувач може орієнтуватися на якість або популярність товару. Окремою функцією на сайті є можливість придбати платну підписку що передбачає привілейовані умови для користувача що її оформить, а саме отримувати безкоштовну доставку протягом року від фірми (рис.1.2).

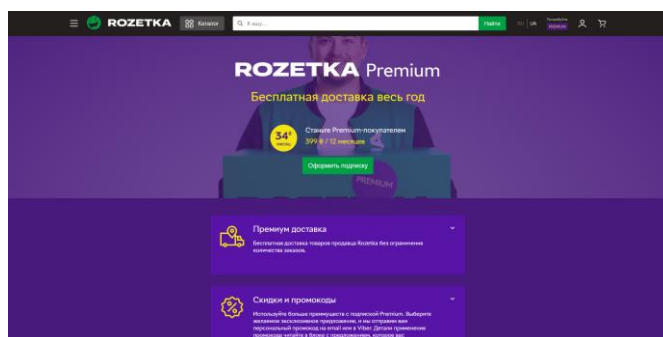


Рис. 1.2 - Платна підписка

					ІАЛЦ.467800.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновок до Розділу 1

В першому розділі були наведені теоретичні відомості про призначення та види інтернет-ресурсів, наведена їх класифікація. Розглянули переваги та недоліки веб-сервісів. Наведені загальні положення про веб-сервіси, розглянули їх підтипи. Ми оглянули платформи для розробки веб-сервісів. Провели порівняльний огляд аналогів веб-сервісів з продажу картин, де ми з'ясували особливості і переваги кожного веб-сервісу.

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		17

РОЗДІЛ 2

ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ДЛЯ ПРОДАЖУ КАРТИН ТА АКСЕСУАРІВ

2.1 Опис функціоналу сервісу

Основні функції створюваної системи:

- Простий інтерфейс;
- Зручність використання;
- Доступність до будь-якого рівня користувачів;

2.2 Порівняльний аналіз програм для розробки веб-сервісів

2.2.1 Порівняння PHP та Python

Основна відмінність між Python та PHP полягає в тому, що: PHP в основному використовується для веб-розробки, тоді як Python в основному використовується для аналізу даних. З огляду на це, ви можете використовувати будь-яку мову для різних цілей, і тому я розкрию основні відмінності між Python та PHP.

Коли обрати Python або PHP?

PHP - це популярна мова кодування для веб-розробки. Зазвичай PHP використовується для створення веб-сайтів, веб-додатків та блогів. Усі сайти WordPress створені на PHP.

					ІАЛЦ.467800.003 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

Вікіпедія та Facebook також були побудовані за допомогою PHP.

Python - це мова програмування загального призначення, яка використовується в різних галузях - від науки про дані до робототехніки. Google використовує Python для розробки, а користувачі Lyft Python для аналізу даних. Python можна використовувати для створення веб-сайту (наприклад, PHP), використовуючи популярний фреймворк Python Django.

Мови

Python безпосередньо підходить до програмування, і його основна мета - зробити все видимим для програміста. Він був розроблений таким чином, щоб його легко було підібрати новачкам, і як результат, його легко вивчити та налагодити. Ось чому сьогодні багато класів та курсів програмування для початківців починаються з Python. Також є додаткова перевага в тому, що Python поставляється попередньо встановленим на комп'ютерах Linux та Mac.

У наші дні PHP іноді є прикладом жартів розробників. Існує дві причини: історія випадкових недоліків безпеки та синтаксис, що дозволило розробникам початківців писати дуже неакуратний код. Ці дві проблеми з часом удосконалювались, але, здається, ці проблеми з минулого мови ніколи не зникають. З огляду на це, почати роботу з PHP неймовірно просто, якщо ви хочете зайнятися веб-розробкою, оскільки не потрібні складні конфігурації.

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

Веб-розробка

Як початківець, можливо, ви захочете просто зайти і почати писати код, який просто працює. Якщо це так, і ви зацікавлені у веб-розробці, PHP - хороший вибір. Він універсально встановлений практично на кожному сервісі спільного хостингу, тому ви можете почати писати код і публікувати свій перший веб-сайт в Інтернеті за дуже короткий проміжок часу.

Ось простий приклад програми в PHP:

```
<? php echo "Привіт, світе!"; ?>
```

Цей код миттєво надрукує привітання на веб-сторінці. Це так просто. На відміну від PHP, написання “Hello World” за допомогою коду Python настільки ж просте. І хоча деякі справді хороші фреймворки, такі як Django, існують для веб-розробки, почати роботу складніше, ніж у PHP. Також важче знайти дешеві послуги веб-хостингу, які підтримують Python, що може розчарувати.

Інше, що полегшує PHP, це те, що ви можете додавати PHP між HTML. Отже, якщо у вас є досвід роботи з HTML та базовою інтерфейсною веб-розробкою, ви можете дуже легко перетворити статичний веб-сайт на динамічний:

```
<html> <body> <? [PHP-код тут]?> </body> </html>
```

У цьому прикладі вихід PHP просто замінить теги PHP, що надзвичайно зручно при роботі з веб-сайтами. Переважна більшість серверного програмного забезпечення, систем ведення блогів, CMS та платформ електронної комерції написані на PHP, і з поважних причин це просто працює, коли справа стосується веб-розробки.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

Крім того, практично все програмне забезпечення веб-сервера (наприклад, Apache та Nginx) підтримує PHP прямо з комплекту. На відміну від PHP, налаштування Python може бути більш складним. PHP простіше як підтримувати, так і розгортати [4].

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		21

2.2.2 Порівняння JavaScript та Java

JavaScript

У перші часи Інтернету, приблизно в 1990 р., Internet Explorer і Netscape Navigator були двома найпопулярнішими веб-браузерами, якими користувалася широка громадськість. Буквально за 10 днів розробник на ім'я Брендан Ейч з Netscape, Inc. зміг створити мову програмування, яка могла б вказувати комп'ютеру про те, як взаємодіяти з користувачем після отримання вводу користувача. Вони назвали цю мову "Livescript" та інтегрували її безпосередньо в Навігатор. Це означало, що браузер буквально інтерпретував команди користувача; код не потрібно було компілювати. Інша мова програмування, Java, набирала популярності приблизно в той самий час, хоча Java вимагала окремого плагіна для роботи. Netscape побачив можливість і перейменував свою мову на "JavaScript". JavaScript був незамінним у перші часи Інтернету.

Java

Java - це мова програмування, винайдена Джеймсом Гослінгом та розроблена Sun Microsystems. Розробка зайняла трохи більше 10 днів - це зайняло близько чотирьох років. У 1991 році група інженерів Sun Microsystem під назвою «Зелена команда» працювала день і ніч над створенням Java. Вперше він був випущений в 1995 році, і з тих пір було запущено кілька нових версій. Сучасний Інтернет абсолютно насичений програмним забезпеченням на Java, і ми маємо Java подякувати за багато програм, що полегшують повсякденне життя. На даний момент Java належить Oracle.

					ІАЛЦ.467800.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

Основні відмінності між Java та JavaScript

Як Java, так і JavaScript написані, зібрані та виконуються по-різному, і кожен із них має суттєві відмінності, коли справа стосується того, що він здатний зробити. Java використовується в широкому діапазоні місць, включаючи програми для Android, програмування кредитних карток, а також для створення настільних програм та веб-корпоративних програм. Для порівняння, JavaScript в основному використовується для того, щоб зробити веб-сторінки більш інтерактивними. Його можна використовувати як альтернативу Flash, хоча більшість програмістів скажуть вам, що JavaScript є більш популярним і має більше функціональних можливостей, ніж Flash. JavaScript можна використовувати для таких акуратних речей, як створення анімації в HTML. У двох словах, коли мова заходить про те, як використовується кожна мова, Java, як правило, використовується для всієї розробки на стороні сервера, тоді як JavaScript зарезервований для розробки сценаріїв на стороні клієнта для таких функцій, як перевірка та інтерактивність. Є деякі інші ключові відмінності:

- Код Java повинен бути скомпільований, а код JavaScript є повністю текстовим.
- Для кожної мови потрібні різні плагіни.
- Код JavaScript запускається лише у браузері, тоді як Java створює програми, що працюють у віртуальній машині або браузері.
- Java - це мова ООП (об'єктно-орієнтоване програмування), а JavaScript - це спеціально мова сценаріїв ООП.

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

23

Новий студент програмування дізнається, що як Java, так і JavaScript є вартими для додавання мовами до свого кодування. Насправді, багато програмістів та розробників кажуть, що вивчення цих двох мов на ранніх стадіях, спиною до спини, є дуже мудрим кроком для нового програміста. Без сумніву, наявність Java і JavaScript під своїм колом зробить програміста більш працездатним і призведе до більш прибуткових можливостей для роботи.

2.2.3 Вибір між C # та C ++ для веб-розробки

Порівняння веб-розробок C ++ та C # - це складне питання. Розробники звертаються до C ++, коли їх пріоритетом є швидкість додатків. Люди також пов'язують цю мову з більш складними та потужними проектами, такими як ігри, браузері або операційні системи. З іншого боку, у нас є C #, який є кращим для створення програмного забезпечення Windows та веб-розробки (як правило, фонові). Крім того, C # є більш популярним, ніж C ++ у веб-розробці.

C # - це мова .NET, що означає, що вона націлена на веб-розробку як на одне з основних цілей. Це мова для початківців, але вам також потрібно дізнатись про ASP.NET. Це фреймворк з відкритим вихідним кодом для серверних веб-додатків. C # працює з ASP.NET для створення веб-додатків. Одне, на що слід звернути увагу, - це те, що код, створений в C #, потрібно буде перетворити в HTML за допомогою .NET framework. В іншому випадку ваш код не працюватиме у браузерах.

					ІАЛЦ.467800.003 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

Слід пам'ятати, що C # призначений для програмування на стороні сервера. Зовнішній вигляд веб-додатків можна покращити за допомогою фреймворку Bootstrap (або іншого).

В обговоренні C # проти C ++ останній також є кандидатом на веб-розробку. Тим не менше, C ++ рідко застосовується в цій галузі. Частіше бачити C ++ у таких частинах веб-додатків, які мають критичне значення швидкості. Наприклад, ми можемо знайти сліди C ++ на Amazon.com, Google або Facebook, щоб гарантувати приємну продуктивність для користувачів. Крім того, C ++ є більш схильним до помилок і занадто низьким рівнем, щоб бути мовою для веб-розробки. Ще одним яскравим недоліком використання C ++ у цій галузі є те, що є кращі, більш зручні для початківців варіанти. Тому аналіз C # проти C ++ з точки зору веб-розробки призводить до того, що C # є більш підходящим вибором.

2.2.4 Сучасні бази даних: призначення, переваги і недоліки

Існують сотні баз даних SQL і NoSQL. Одні популярні, інші ігноруються. Деякі прості і добре задокументовані, а деякі складні у використанні. Одні мають відкритий код, а інші пропрієтарні. Що, можливо, найбільш важливо, деякі масштабуються, оптимізовані, високодоступних, а деякі складно масштабувати або підтримувати.

					ІАЛЦ.467800.003 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

Виникає природне запитання: яку базу даних вибрати? Щоб відповісти на нього, ми повинні вирішити, чого ми хочемо досягти з допомогою бази даних. Щоб скласти уявлення, необхідно відповісти на наступні питання:

- Чи потрібен нам аналітичний доступ до бази даних?
- Чи потрібно нам писати або читати в реальному часі?
- Скільки таблиць ми хочемо зберегти?
- Яка доступність нам потрібна?
- Чи зможемо ми отримати доступ до таблиць, відфільтрованим за стовпцями або по рядках?

Специфіка кожної БД може відрізнятися і тому потрібно пам'ятати, що може запропонувати та чи інша база даних.

Реляційні бази даних SQL

Цей тип, є найпопулярнішим і поширеним. Такі БД дозволяють зберігати дані в реляційних таблицях з певними стовпцями певного типу. Реляційні таблиці гарні для нормалізації і об'єднання.

Переваги:

- підтримка SQL
- ACID-транзакції
- Підтримка індексації та поділу

					ІАЛЦ.467800.003 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

Недоліки:

- Погана підтримка неструктурованих даних
- Складне масштабування
- Погана оптимізація обробки подій

Приклади: Oracle DB, MySQL, PostgreSQL.

Документно-орієнтовані бази даних

Якщо ми не хочемо об'єднувати кілька таблиць для отримання потрібних даних, ми можемо поглянути на документно-орієнтовані бази даних. Вони дозволяють зберігати записи в форматі JSON. У цьому форматі ми можемо створити складне значення для будь-якого ключа і відразу включити всю структуру даних в одну запис.

Переваги:

- Немає прив'язки до схеми
- Підходить для OLTP
- Гарна підтримка складних типів

Недоліки:

- Погана підтримка транзакцій
- Слабка аналітична підтримка

Приклади: MongoDB.

					ІАЛЦ.467800.003 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

Бази даних в оперативній пам'яті

Бази даних цього типу можуть надавати в реальному часі відповідь для вибору і вставки певних записів. Більшість з них в основному зберігають дані в ОЗУ, але в деяких випадках вони також пропонують постійне сховище на жорстких дисках або твердотільних накопичувачах. Більшість цих баз даних працюють із записами «ключ-значення», тому значення можна запам'ятовувати в форматі, орієнтованому на документи. Але деякі бази даних також працюють за допомогою стовпців і дозволяють вторинне індексування тієї ж таблиці. Використання ОЗУ дозволяє обробляти дані швидко, але робить їх більш нестабільними і дорогими [5].

Переваги:

- Швидке написання
- Швидке читання

Недоліки:

- Важкодосяжна надійність
- Дороге масштабування

Приклади: Redis, Tarantool, Apache Ignite.

					ІАЛЦ.467800.003 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

Пошукова система

Якщо ми хочемо отримати доступ до даних за допомогою фільтра по будь-якому значенню і навіть по будь-якому слову в стовпці, ми повинні згадати про пошукові системи. Ці бази даних виконують індексацію кожного слова в стовпчиках і дозволяють виконувати повнотекстовий пошук. Вони ідеально підходять для зберігання і аналізу журналів або великих текстових значень.

Переваги:

- Швидкий доступ по будь-якому слову
- Гарна масштабованість

Недоліки:

- Підходить тільки для пакетних вставок
- Погана аналітична підтримка

Приклади: Elastic.

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

Графові бази даних

Для деяких випадків підходять графові структури даних. Якщо завдання вимагають роботи з графами, існують спеціальні бази даних, які задовольняють потреби.

Переваги:

- Структура даних графа
- Гнучкі конструкції

Недоліки:

- Спеціальна мова запитів
- Важко масштабувати

Приклади: Neo4j.

					ІАЛЦ.467800.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновок до Розділу 2

У другому розділі ми провели порівняння існуючих програм для розробки веб-сервісу, і зробили висновок що деякі програми можуть не підходити для розробки сайту. З цих висновків ми прийшли до рішення вибрати одне влучне програмне забезпечення для розробки веб-сервісу. Це VSCode.

Особливості Visual Studio Code:

- VS Code дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ.
- У редакторі присутні вбудований отладчик, інструменти для роботи з Git і засоби рефакторинга, навігації по коду, автодоповнення типових конструкцій і контекстної підказки.
- Продукт підтримує розробку для платформ ASP.NET і Node.js, і вважається легковажним рішенням, яке дозволяє обійтися без повної інтегрованого середовища розробки.
- Великим плюсом редактора є підтримка великої кількості мов, таких як C++, C#, Python, PHP, JavaScript та інших.
- Підтримка TypeScript і JavaScript;
- Інтерфейс можна розділити на дві панелі для порівняння коду;

Враховуючи все вищесказане, можна відмітити переваги сервісу у простоті його інтерфейсу і зручності використання.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

РОЗДІЛ 3

РОЗРОБКА ВЕБ-СЕРВІСУ

3.1 Вибір програми

Щоб полегшити розробку веб-сервісу треба обрати зручну й функціональну програму. Мій вибір впав на VSCode, і тепер розглянемо чого ця програма найкраща для написання веб-сервісу.

1. Щоб не витратити багато часу на підготовку html документу, ця програма облегшує цю процедуру. Наприклад, кожен html документ має такий вихідний код (рис.3.1)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Рис. 3.1

Елемент <!DOCTYPE> призначений для вказівки типу поточного документа - DTD (document type definition, опис типу документа). Це необхідно, щоб браузер розумів, як слід інтерпретувати поточну веб-сторінку, адже HTML існує в декількох версіях. Щоб браузер не плутався й розумів, за яким стандартом відображати веб-сторінку й необхідно в першому рядку коду задавати <!DOCTYPE> [8].

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

32

2. Ми можемо швидко створити будь-які HTML Теги, визначити їх ідентифікатор, імена класів, знаючи лише кілька швидких скорочень. Наприклад, якщо ми хочемо створити невпорядкований список із чотирьох записів, кожна з яких має унікальну назву класу, треба ввести `ul> li.item $ * 4`, і отримаємо результат як на зображенні: (рис.3.2)

```
<ul>
  <li class="item1"></li>
  <li class="item2"></li>
  <li class="item3"></li>
  <li class="item4"></li>
</ul>
```

Рис. 3.2

У цьому прикладі ми використовували скороченні символи:

- 1) `>` вказує що наступний тег повинен бути нащадком попереднього
- 2) `.` вказує що наступний текст, це назва класу
- 3) `$` використовується для друку унікального числа
- 4) `*` дозволяє швидко помножити тег

3. Теги автоматичного оновлення

При зміні тегу пов'язана функція редагування автоматично оновлює відповідний закриваючий тег. Наприклад, я видалив тег `ul` і програма автоматично видала закриваючий тег (рис.3.3).

```

<>
<li class="item1"></li>
<li class="item2"></li>
<li class="item3"></li>
<li class="item4"></li>
</>

```

Рис. 3.3

4. Live server

Якщо створити новий елемент у HTML, наприклад `<h1>VScode</h1>`, то після збереження сторінка автоматично перезавантажиться, і ми можемо побачити зміни.

3.2 Складова HTML документа

Кожний документ HTML складається з 5 головних тегів.

- 1) Тег `<html>` - визначає початок файлу HTML, всередині якого зберігається заголовок і тіло документа (`<head>` - заголовок, `<body>` - тіло документа).
- 2) Тег або блок `<head>` - може містити текст і теги, але вміст цього розділу не відображається на сторінці, за винятком контейнера `<title>`.
- 3) Тег `<meta>` є універсальним і додає цілий клас можливостей, зокрема, за допомогою метатегів, можна змінювати кодування сторінки, додавати ключові слова, опис документа і багато іншого. Щоб браузер розумів, що має справу з кодуванням UTF-8 (Unicode transformation format, формат перетворення Юнікод) і додається даний рядок.
- 4) Тег `<title>` визначає заголовок веб-сторінки, це один з важливих елементів (рис.3.4).

Рис. 3.4

5) Тег <body> - тіло документа , призначений для тегів і змістовної частини веб-сторінки.

3.3 Розробка веб-сторінки

3.3.1 Розробка першої веб-сторінки

Для початку щоб створити веб-сторінку, треба створити папку де буде зберігатися наш веб-сервіс. Папка з усіма файлами виглядає так: (рис.3.5).

css	31.05.2021 17:14	Папка с файлами	
img	31.05.2021 17:14	Папка с файлами	
js	31.05.2021 17:14	Папка с файлами	
aboutus	30.05.2021 21:08	Chrome HTML Досу...	8 КБ
basket	30.05.2021 22:26	Chrome HTML Досу...	6 КБ
index	27.05.2021 14:07	Chrome HTML Досу...	8 КБ
ordering	31.05.2021 17:13	Chrome HTML Досу...	6 КБ
productcard	30.05.2021 15:28	Chrome HTML Досу...	8 КБ
products	30.05.2021 15:25	Chrome HTML Досу...	10 КБ

Рис. 3.5

Ми бачимо, що для кожної сторінки, треба створити окремий html файл. Також, щоб використовувати різні фотографії або зображення треба зробити окрему папку у головній папці, й потім у html коді вказати на цей файл.

Отже, створили перший html файл, це буде наша перша сторінка або головна, де будуть відображатись частини інших сторінко даного веб-сервісу (рис.3.6).

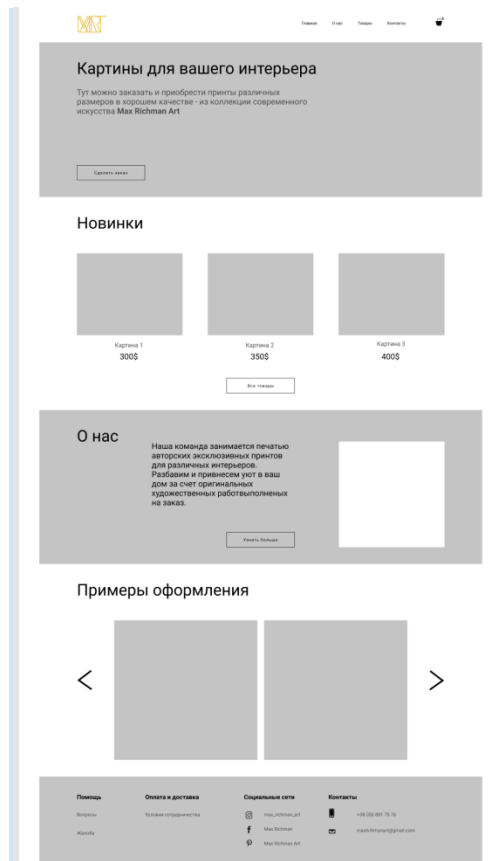


Рис. 3.6

Перейдемо до коду. Спочатку треба підготувати html документ, ми вже розглянули як його підготувати у розділі 3.1.(рис.3.7)

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

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="css/normalize.css">
  <link rel="stylesheet" href="css/slick.css">
  <link rel="stylesheet" href="css/styles.css">
  <title>MRXART</title>
</head>
```

Рис. 3.7

У підготовці HTML документа, ми підключили до нього CSS документ. Також назвали сам веб-сервіс у тезі <title>.

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

36

Далі, нам треба зробити “шапку” веб-сторінки. Для цього у тезі <body> ми додаємо тег <header>(задає шапку сайту або розділу, в якій зазвичай розташовується заголовок)(рис.3.8)

```
<header class="header">
  <div class="container">
    <div class="header__inner">
      <div class="logo">
        <a href="#">
          
        </a>
      </div>
      <nav class="nav header__nav">
        <ul class="nav__list">
          <li class="nav__item"><a href="#" class="nav__link">Главная</a></li>
          <li class="nav__item"><a href="#" class="nav__link">О нас</a></li>
          <li class="nav__item"><a href="#" class="nav__link">Товары</a></li>
          <li class="nav__item"><a href="#" class="nav__link">Контакты</a></li>
        </ul>
      </nav>
      <ul class="header__shop">
        <li class="shop__item">
          <a href="#" class="basket__link">
            <img class="basket" src="">
            <span class="cart__quantity">2</span>
          </a>
        </li>
      </ul>
    </div>
  </div>
</header>
```

Рис. 3.8

Ми присвоїли тегу <header> клас щоб, при стилізації веб-сторінки нам було зручно редагувати тільки шапку. Для цього й присвоюються класи для кожного тегу, щоб кожний тег, можливо було редагувати окремо, а не всі разом.

Отже, ми зробили шапку сайту и ось так вона виглядає: (рис.3.9).



Рис. 3.9

Далі ми переходимо до головного блоку. У головного блоку відображені чотири підблоки. Код першого підблока.(рис.3.10)

```
<section class="top">
  <div class="container container--bg">
    <div class="top_block">
      <h1 class="top_title">Картины для вашего интерьера</h1>
      <p class="top_subtitle">
        Тут можно заказать и приобрести принты различных размеров в хорошем качестве - из коллекции
        современного искусства <strong>Max Richman Art</strong>
      </p>
      <button class="btn-order">Сделать заказ</button>
    </div>
  </div>
</section>
```

Рис. 3.10

Тут ми застосували тег `<section>` , який може використовуватися для блоку новин, контактної інформації. Також ми додали кнопку для замовлення, щоб користувач міг одразу замовити товар на головній сторінці. Так само ми робимо для інших підблоків. В останньому підблоці, ми закріптували фотографії, щоб у користувача була можливість переглянути приклади оформлення.

Переходимо до останнього блоку, блоку `<footer>`. Тег footer задає “підвал” сайту, в якому може розташовуватися ім’я автора або контактна інформація(рис.3.11).

```

<footer class="footer">
  <div class="container">
    <div class="footer_items">
      <div class="footer_item">
        <h4 class="social_title">Социальные сети</h4>
        <ul class="social">
          <li class="social_item">
            <a class="social_link social_link--in" href="#" target="_blank">max_richman_art</a>
          </li>
          <li class="social_item">
            <a class="social_link social_link--fb" href="#" target="_blank">Max Richman</a>
          </li>
          <li class="social_item">
            <a class="social_link social_link--pin" href="#" target="_blank">Max Richman Art</a>
          </li>
        </ul>
      </div>
      <div class="footer_item">
        <h4 class="contact_title">Контакты </h4>
        <ul class="contacts">
          <li class="contact_tel">
            <a href="tel:+38 050 881 75 76">+38 050 881 75 76</a>
          </li>
          <li class="contact_mail">
            <a href="mailto:maxrichmanart@gmail.com">maxrichmanart@gmail.com</a>
          </li>
        </ul>
      </div>
    </div>
  </div>
  <div class="footer_copy">
    <div class="copy">
      All rights reserved 2021
    </div>
  </div>
</footer>

```

Рис. 3.11

Отже, ми зробили першу сторінку нашого веб-сервісу.

3.3.2 Розробка другої веб-сторінки

Друга веб-сторінка як і перша складається з “шапки” та “підвалу”, але друга призначена для товарів.(рис.3.12)

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

39

Товары

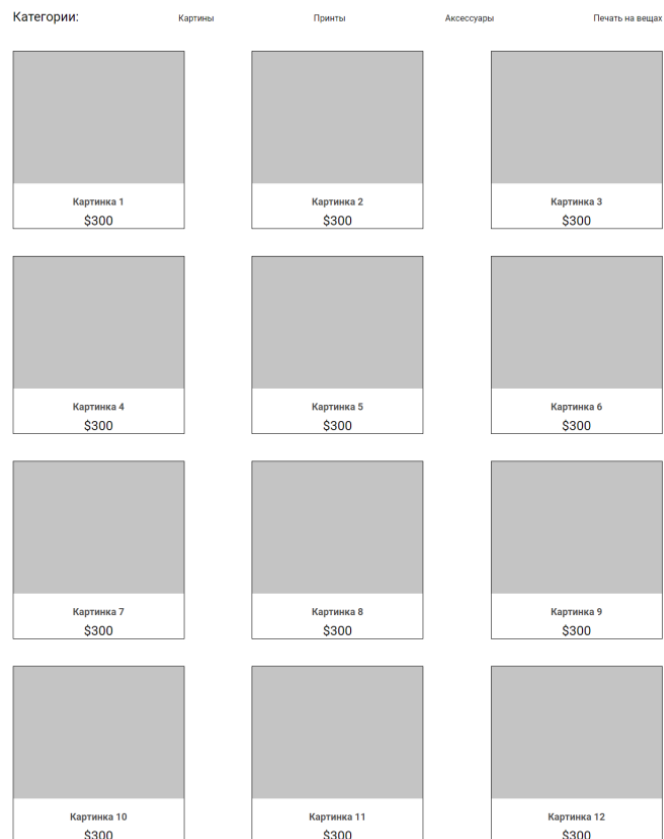


Рис. 3.12

В цьому випадку для всіх товарів ми присвоїли один блок, а для кожного трьох товарів, під блок. Кожен блок товару має, фотографію, назву та ціну(рис.3.13).

```

<div class="items_inner">
  <div class="items_product">
    <a class="item_link" href="#" target="_blank">
      
    </a>
    <h5 class="item_desc">Картинка 1</h5>
    <p class="item_price">$300</p>
  </div>

```

Рис. 3.13

Для того щоб відображувалась рамка для кожного товару, ми прописуємо у CSS документі параметр оформлення (рис.3.14).

```
.items__product {  
  border: 1px solid;  
}
```

Рис. 3.14

Border – це, безпосередньо, сама рамка, а 1px розмір цієї рамки.

І це все що було зроблено на другій сторінці.

3.3.3 Розробка третьої веб-сторінки

Третя веб-сторінка це картка товару, тобто коли користувач натисне на товар на другій веб-сторінці, його перекине на картку товару, де буде відображатися розмір товару, колір та стиль. Також, на цій сторінці користувач може замовити або додати товар у корзину (рис.3.15).



Рис. 3.15

					ІАЛЦ.467800.003 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

Також на цій сторінці були додані варіанти використання та схожі товари, для того щоб користувач мав можливість переглянути, чи підійде цей товар для його інтер'єру.

3.3.4 Розробка четвертої веб-сторінки

Четверта веб-сторінка складається з інформації про художника. Ця веб-сторінка складається як і попередні сторінки з шапки, підвалу та інформації, а інформація складається з трьох під блоків:

- 1) Проста інформація
- 2) Інформація з малюнками
- 3) Та соц. Мережі

Під блок про соц. мережу має слайди, щоб переглянути життя художника, та кнопка підписки на цю соц. мережу(рис.3.16).

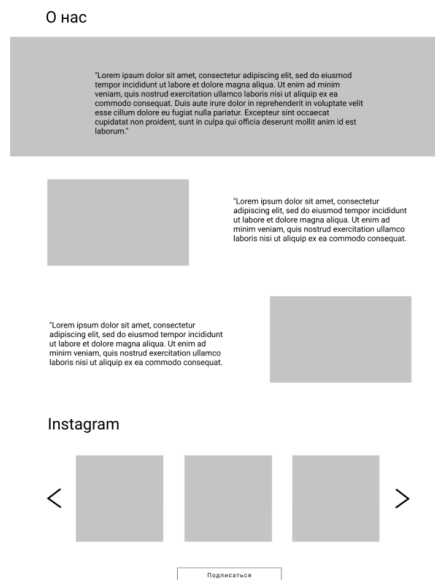


Рис. 3.16

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

42

У самому коді ми заскриптували кнопку, тобто коли користувач на неї натисне його перекине на соц. мережу.

3.3.5 Розробка п'ятої веб-сторінки

П'ята веб-сторінка – це кошик де користувач, який обрав товари, які йому до вподоби, може зробити замовлення(рис.3.17).

Корзина

Товаров в корзине: 2

1		Картина 1 150x270 Количество: - 1 +	Цена: 300\$
2		Картина 2 150x270 Количество: - 1 +	Цена: 300\$
Сума: 600\$			Оформить заказ

Рис. 3.17

Користувач може обрати кількість одного товару натиснув на кнопку біля картини. Також ми заскриптували що при додаванні товарів у кошик, сума всіх товарів буде або збільшуватися або зменшуватися. При натискання кнопки “оформити замовлення”, користувача перекине на останню сторінку нашого веб-сервісу, де він і зможе зробити замовлення товару.

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

43

Користувачі інтернет-магазинів регулярно додають і видаляють елементи зі свого «кошика», і тому кількість товару на складі може постійно змінюватися протягом покупки кілька разів, до того ж користувач магазину може відмовитися від покупок в будь-який момент, а іноді можуть виникнути інші непередбачені проблеми для вирішення яких потрібно скасувати замовлення.

Все це може трохи ускладнювати облік наявних на складі товарів, і нам потрібно переконатися що користувачі не зможуть «купити» ті товари яких вже зарезервовані.

Тому у кошики буде час на протязі якого, якщо вона була неактивна, то зарезервовані товари знову стануть доступними для інших і корзина буде очищена.

У колекцію Inventory повинні зберігатися документи, що містять поточну кількість запасів по кожній одиниці складського обліку (SKU; або item), а також список кошиків з кількістю товарів в кожній. Тоді ми можемо повернути доступні залишки, якщо вони перебували в кошику, але клієнт не користувався кошиком якийсь час [7].

У наступному прикладі `_id` поле що зберігає SKU

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		44

```
{
  _id: '00e8da9b',
  qty: 16,
  carted: [
    { qty: 1, cart_id: 42, timestamp: ISODate("2012-03-09T20:55:36Z") },
    { qty: 2, cart_id: 43, timestamp: ISODate("2012-03-09T21:55:36Z") },
  ]
}
```

В даному прикладі використовується проста схема, в продакшені ми можемо об'єднати цю схему з каталогом товарів.

Для нормального моделювання кошика ми потребуємо SKU, quantity полях, а також в якихось уточнюючих полях, що знаходяться в item_details

```
{
  _id: 42,
  last_modified: ISODate("2012-03-09T20:55:36Z"),
  status: 'active',
  items: [
    { sku: '00e8da9b', qty: 1, item_details: {...} },
    { sku: '0ab42f88', qty: 4, item_details: {...} }
  ]
}
```

					ІАЛЦ.467800.003 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

Операції

Додавання елемента в кошик

Основним моментом у роботі з кошиком є переміщення товару з наявних запасів на складі в кошик.

Важливо не помістити в кошик вже зарезервований товар. Реалізуємо це за допомогою функції `add_item_to_cart`.

```
def add_item_to_cart(cart_id, sku, qty, details):
    now = datetime.utcnow()
    result = db.cart.update(
        {'_id': cart_id, 'status': 'active' },
        {'$set': {'last_modified': now}, '$push': {'items': {'sku': sku, 'qty': qty,
        'details': details}}},
        w=1
    )
    if not result['updatedExisting']:
        raise CartInactive()
    result = db.inventory.update(
        {'_id': sku, 'qty': {'$gte': qty}},
        {'$inc': {'qty': -qty}, '$push': {'carted': {'qty': qty, 'cart_id': cart_id,
        'timestamp': now}}},
        w=1
    )
```

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

46

```

if not result['updatedExisting']:
    db.cart.update( { '_id': cart_id }, { '$pull': { 'items': {'sku': sku } } })
    raise InadequateInventory()

```

Перша операція в цій функції - це перевірка, що кошик активний і тільки після цього туди додається товар. Тоді перевіряємо доступність потрібної кількості залишків на складі і якщо так, то зменшуємо їх.

Зміна кількості в кошику

Додаток має перевіряти, що коли користувач збільшує кількість товарів в кошику, вони є в наявності на складі.

```

def update_quantity(cart_id, sku, old_qty, new_qty):
    now = datetime.utcnow()
    delta_qty = new_qty - old_qty
    result = db.cart.update(
        { '_id': cart_id, 'status': 'active', 'items.sku': sku },
        { '$set': { 'last_modified': now, 'items.$.qty': new_qty } },
        w=1
    )
    if not result['updatedExisting']:
        raise CartInactive()
    result = db.inventory.update(
        { '_id': sku, 'carted.cart_id': cart_id, 'qty': { '$gte': delta_qty } },
        { '$inc': { 'qty': -delta_qty }, '$set': { 'carted.$.qty': new_qty, 'timestamp': now } },

```

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

```

        w=1
    )
    if not result['updatedExisting']:
db.cart.update( {'_id': cart_id, 'items.sku': sku }, {'$set': { 'items.$.qty': old_qty } }
)
        raise InadequateInventory()

```

Повернення товарів на склад.

Періодичного має обнуляти неактивні кошика і повертати товари назад на склад. Мінлива timeout містить в собі час закінчення активності кошика.

```

def expire_carts(timeout):
    now = datetime.utcnow()
    threshold = now - timedelta(seconds=timeout)

db.cart.update(

    {'status': 'active', 'last_modified': { '$lt': threshold } },

    {'$set': { 'status': 'expiring' } },

    multi=True

)

for cart in db.cart.find({'status': 'expiring'}):

#Повернення товару на склад

for item in cart['items']:
db.inventory.update(
    { '_id': item['sku'], 'carted.cart_id': cart['id'], 'carted.qty': item['qty'] },

```

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

48

```
{ '$inc': { 'qty': item['qty'] }, '$pull': { 'carted': { 'cart_id': cart['id'] } } }
```

)

```
db.cart.update( { '_id': cart['id'] }, { '$set': { 'status': 'expired' } })
```

Ця функція:

- 1) Знаходить всі кошики, які старше заданого часу.
- 2) Для кожної неактивного кошика, усі товари повертає в доступні залишки на складі.
- 3) Як тільки усі товари будуть повернуті на склад, встановимо стан кошика expired.

3.3.6 Розробка шостої веб-сторінки

Шоста веб-сторінка – це замовлення товару, тобто в цьому блоці будуть поля, де треба вписати інформацію користувача: його телефон, адресу, а також, як користувач хоче заплатити, картою або готівкою при доставці товару (рис.3.18).

Рис. 3.18

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

49

Код даної веб-сторінки виглядає так (рис.3.19)

```
<section class="ordering">
  <div class="container">
    <div class="ordering_info">
      <h1 class="ordering_title">Оформление заказа</h1>
      <input type="text" id="fname" name="firstname" placeholder="Имя">
      <input type="text" id="lname" name="lastname" placeholder="Фамилия">
      <input type="tel" placeholder="Телефон">
      <input type="email" placeholder="Email">
    </div>
    <div class="ordering_list">
      <input type="text" id="address" placeholder="Адрес">
      <button class="btn-order btn-ordering">Заказать</button>
    </div>
  </div>
  <div class="col">
    <div class="container">
      <div class="ordering_block">
        <a class="order_link" href="#" target="_blank">
          
        </a>
        <h5 class="order_desc">Картинка 1</h5>
        <p class="order_price">$300</p>
      </div>
      <div class="ordering_block">
        <a class="order_link" href="#" target="_blank">
          
        </a>
        <h5 class="order_desc">Картинка 2</h5>
        <p class="order_price">$300</p>
      </div>
      <p>Сума <span class="price"></span><b>600$</b></p>
    </div>
  </div>
</section>
```

Рис. 3.19

Цей код складається з тегу `<input type=` який повідомляє браузеру, до якого типу належить елемент форми. Тобто, якщо ми хочемо додати порожнє поле, де можна вписати ім'я або будь-який текст, ми прописуємо `<input type="text">`, або, якщо хочемо вписати номер телефону: `<input type=tel:+777 777 77>`.

					ІАЛЦ.467800.003 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

В таблиці відображено частина типів та їх опис

Таблиця 3.1

<input type="text">	Можна вписати любий текст.
<input type="number">	Для чисел.
<input type="password">	Для паролю.
<input type="email">	Для email.
<input type="url">	Для посилання.
<input type="tel">	Застосовується для написання мобільного телефону.
<input type="search">	Для пошуку.
<input type="color">	Для вибору кольору.

Коли користувач натисне кнопку “заповіти” інформація надійде до бази даних, і через неї ми зможемо дізнатися що заповнив користувач та зв’язатися з ним.

Для початку потрібно визначити схему Джейсона, тобто всі ці поля повинні бути в ньому задані

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467800.003 ПЗ

Арк.

51

```

{
  orderCredentials: {
    name: "",
    surname: "",
    ...
  },
  order: [
    {
      name: картина1,
      price: ""
    },
    ...
  ]
}

```

Після цього потрібно прописати ендпоінти, за допомогою яких можна звертатися до бд і брати, видаляти, змінювати або додавати нові дані.

Наприклад: "/ api / order" - метод get, за допомогою нього можна отримати замовлення

Метод post - додати замовлення

put - редагувати замовлення

І при зверненні на цей ендпоінт, буде викликатися функція, яка бере передані дані і кладе їх в бд.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

Висновок до Розділу 3

В цьому розділі ми розібрали програму на якій ми створили веб-сервіс і зробили висновок що ця програма є найкращою для створення веб-сервісу, тому що вона дає користувачу багато можливостей, такі як стилізація програми, спрощення написання коду та багато іншого. Також описали складову HTML документа, з чого він складається, а саме, основної частини (підключення HTML документа до документа CSS, з “шапки” та “підвалу”). Розібрали кожну сторінку нашого веб-сервісу, привели приклади різних тегів і де вони застосовуються. Описали принцип роботи баз даних.

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		53

Висновки

Цей проект присвячений повній розробці веб-сервісу, головним завданням якого було створення інтернет магазину для продажу картин та аксесуарів. Для виконання цього завдання, було необхідним проаналізувати загальну структуру веб-сервісів та способи їх розробки. Таким чином було проведено порівняльний аналіз програм для розробки веб-сервісів, та обрано найбільш вдалий варіант для реалізації проекту.

Для повноцінного розуміння контексту заданого сайту, потрібно було провести огляд його складових, ідейного наповнення, характерних елементів. В процесі роботи, важливим етапом стало порівняння сайтів зі схожою тематикою, завдяки чому стало можливим виявлення основоположних компонентів для Інтернет магазину.

Реалізований веб-сервіс має простий та зручний інтерфейс, який зрозумілий на інтуїтивному рівні. Відмінною рисою, що відрізняє веб-сервіс від конкурентів є відсутність реєстраційного поля, що відлякує покупців та забирає багато часу.

					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		54

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

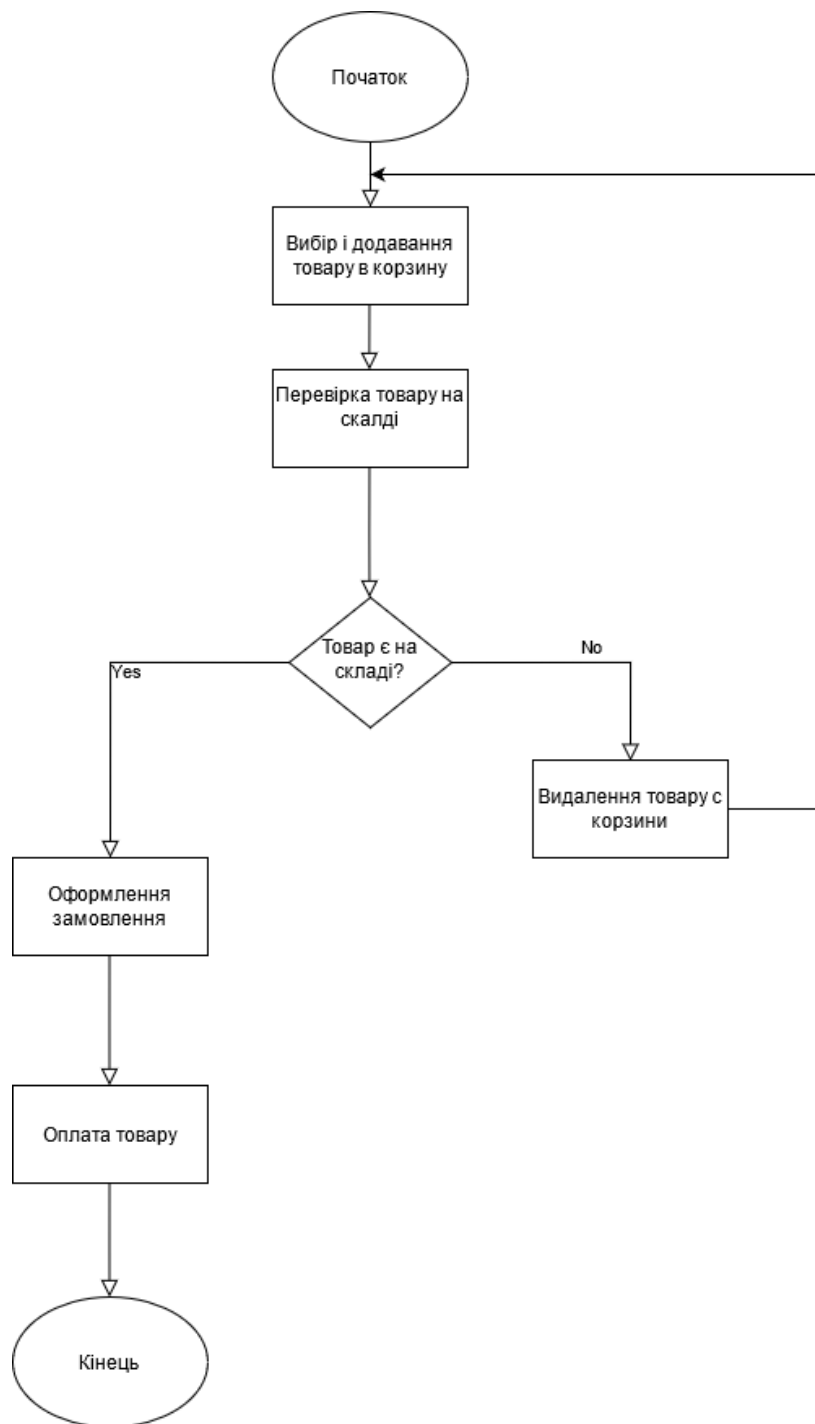
1. Веб-сайт: визначення й застосування [Електронний ресурс] – Режим доступу до ресурсу: <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site>
2. Основи веб-дизайну / О. Г. Пасічник, О. В. Пасічник, І. В. Стеценко : [Навч. посіб.]. — К.: Вид. група ВНУ. — 2009. — 336 с: іл. [Електронний ресурс] – Режим доступу до ресурсу: <https://ktpu.kpi.ua/wp-content/uploads/2014/02/Pasichnik-O.-G.-Pasichnik-O.-V.-Stetsenko-I.-V.-Osnovi-veb-dizajnu.pdf>
3. 8 PaaS to Build and Host Your Modern Applications. [Електронний ресурс] – Режим доступу до ресурсу: <https://geekflare.com/paas-for-modern-apps/>
4. Chris Castiglione. Python vs. PHP: What’s the Difference? [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.onemonth.com/python-vs-php/>
5. А. Гогин. 7 типов современных баз данных: предназначение, достоинства и недостатки. [Електронний ресурс] – Режим доступу до ресурсу: <https://tproger.ru/articles/kakuju-bazu-dannyh-vybrat-dlja-proekta-chtoby-potom-ne-zhalet-ob-jetom/>
6. YauhenKavalchuk. Front-end 2021. Roadmap & Trends. [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/YauhenKavalchuk/useful/blob/main/front-end-2021.md>

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

7. Примеры использования MongoDB в e-commerce [Электронный ресурс] – Режим доступа до ресурсу: <https://habr.com/ru/post/260291/#3>
8. HTML in Visual Studio Code [Электронный ресурс] – Режим доступа до ресурсу: <https://code.visualstudio.com/docs/languages/html>
9. Р.Ивойлов.9 полезных плагинов VS Code для вёрстки [Электронный ресурс] – Режим доступа до ресурсу: <https://htmlacademy.ru/blog/boost/tools/vscode-addons>
10. Довідник W3Schools [Электронный ресурс] – Режим доступа до ресурсу: <https://w3schoolsrus.github.io/index.html>
11. Довідник htmlbook.ru [Электронный ресурс] – Режим доступа до ресурсу: <http://htmlbook.ru/html/span>

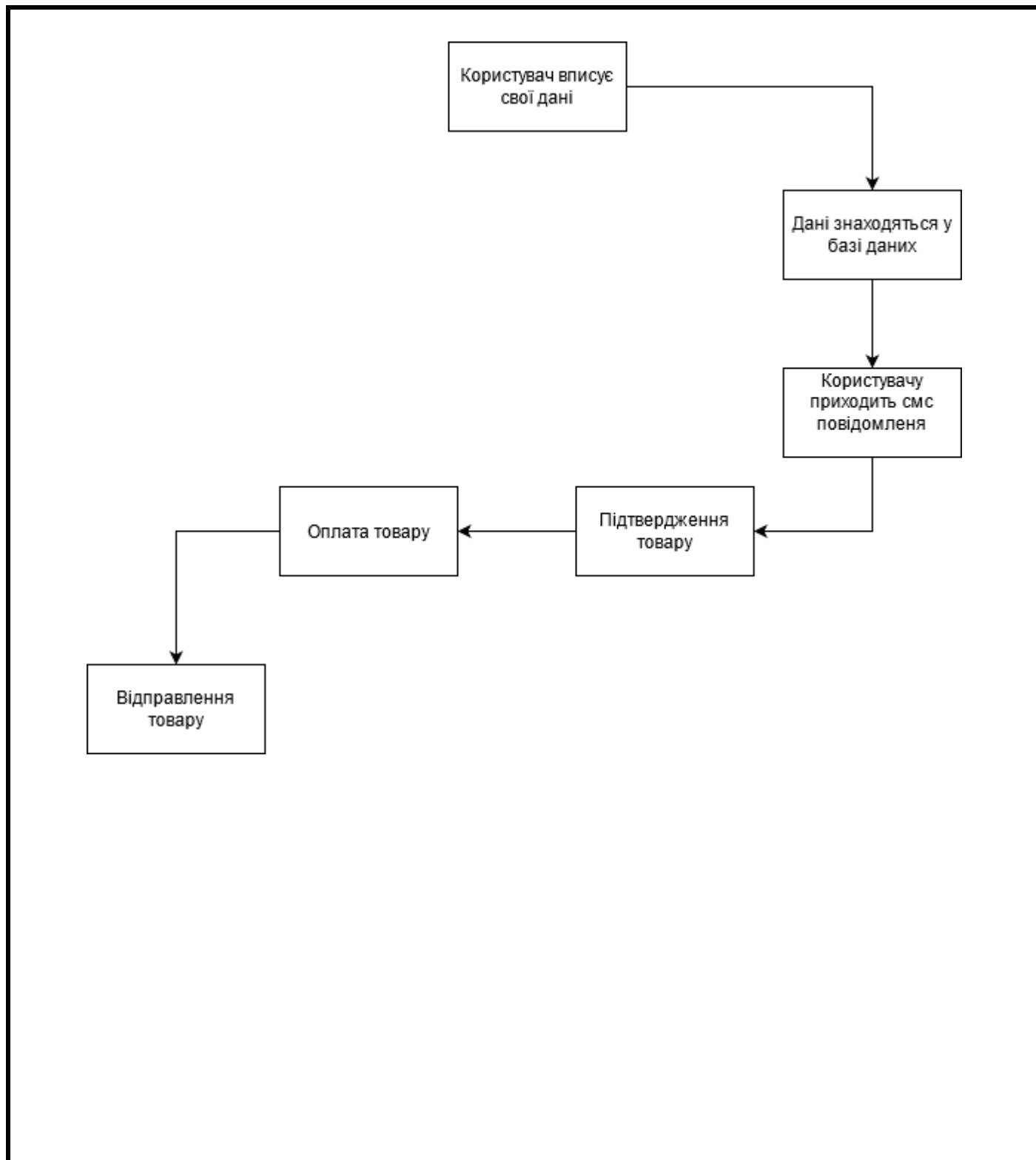
					<i>ІАЛЦ.467800.003 ПЗ</i>	Арк.
						56
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Додаток 1
До дипломного проекту
на тему: «Веб-сервіс для продажу картин та аксесуарів»



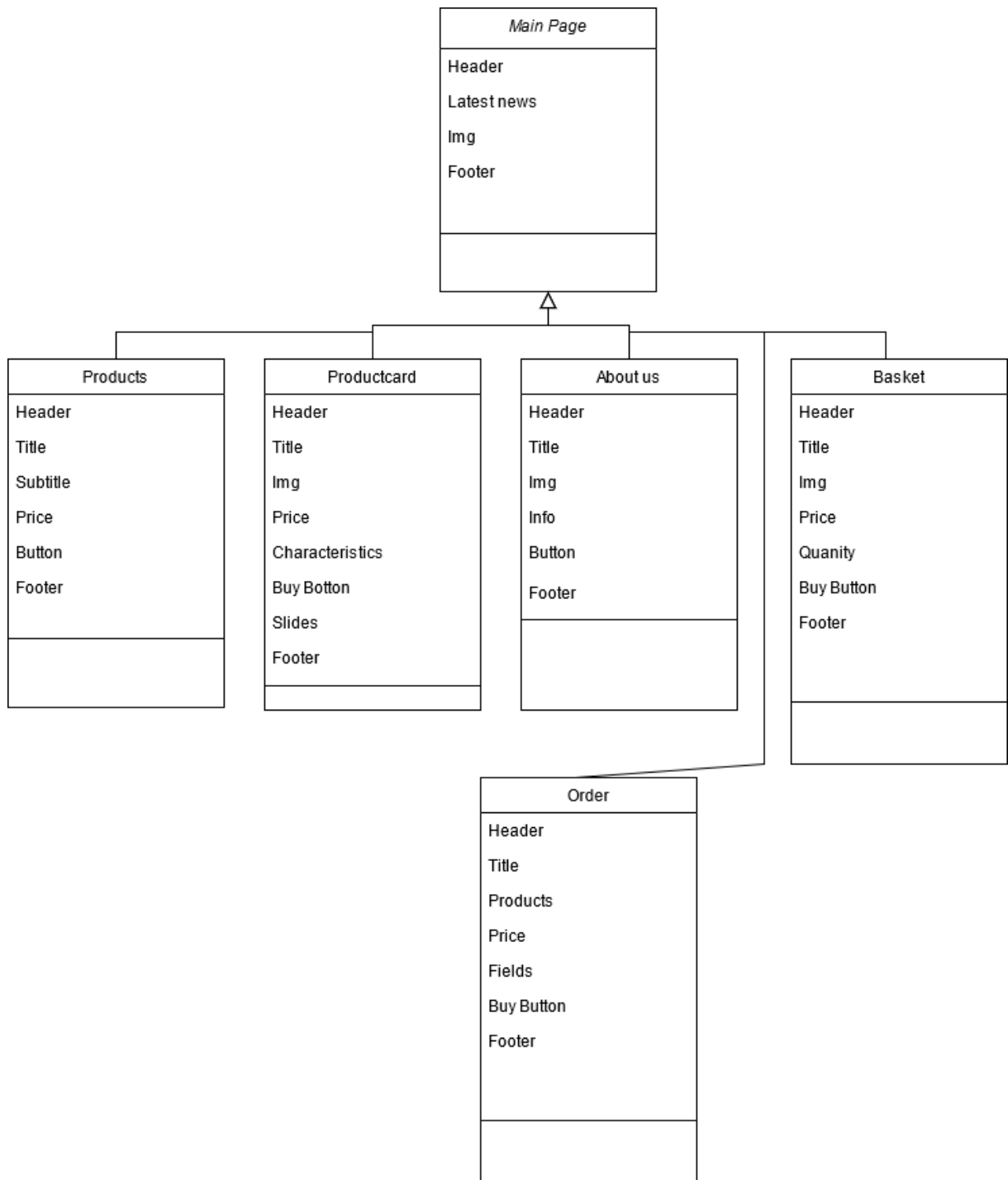
					<i>ІАЛЦ.467800.004 Д1</i>								
Зм	Арк.	№ докум	Підпис	Дата									
Розроб.		Андрієнко В.Р.			<i>Веб-сервіс для продажу картин та аксесуарів</i> <i>Схема додавання товару в козину</i>				Літ.	Арк.	Аркушів		
Первiр.		Луцький Г.М.									1	1	
									<i>КПІ ім. Ігоря Сікорського</i> <i>ФІОТ, ІВ-73</i>				
Н.Контр.		Сімоненко В.П.											
Затверд.													

Додаток 2
До дипломного проекту
на тему: «Веб-сервіс для продажу картин та аксесуарів»



					ІАЛЦ.467800.005 Д2			
					Веб-сервіс для продажу картин та аксесуарів Схема оформлення замовлення			
Зм	Арк.	№ докум	Підпис	Дата				
Розроб.		Андрієнко В.Р.						
Первір.		Луцький Г.М.						
Н.Контр.		Сімоненко В.П.						
Затверд.								
						Літ.	Арк.	Аркушів
							1	1
						КПІ ім. Ігоря Сікорського ФІОТ, ІВ-73		

Додаток 3
До дипломного проекту
на тему: «Веб-сервіс для продажу картин та аксесуарів»



					ІАЛЦ.467800.006 ДЗ								
Зм	Арк.	№ докум	Підпис	Дата									
Розроб.		Андрієнко В.Р.			Веб-сервіс для продажу картин та аксесуарів Структурна схема веб-сервісу				Лім.	Арк.	Аркушів		
Первір.		Луцький Г.М.									1	1	
									КПІ ім. Ігоря Сікорського ФІОТ, ІВ-73				
Н.Контр.		Сімоненко В.П.											
Затверд.													