

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“__” _____ 2022 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 організації процесу доставки продуктових товарів

Виконав : студент 4 курсу, групи Ю-83
(шифр групи)

Сушильников Кирило Дмитрович

(прізвище, ім’я, по батькові)

(підпис)

Керівник асистент Кочура Ю.П.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) професор, д.т.н., Сімоненко В. П.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

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

Студент _____

(підпис)

Київ – 2022 р.

**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”
спеціальності 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
Сергій СТИРЕНКО

_____ (підпис)

“__” _____ 2022 р.

ЗАВДАННЯ

на бакалаврський дипломний проект студента

Сушильнікова Кирила Дмитровича

1. Тема проекту Веб-сервіс для організації процесу доставки продуктивних товарів
керівник проекту Кочура Ю.П., асистент,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 11 травня 2022 року №1139-с
2. Термін здачі студентом закінченого проекту 14 червня 2022 р.
3. Вихідні дані до проекту технічна документація, теоретичні дані.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)
Розділ 1. Огляд існуючих рішень.
Розділ 2. Функціональний вибір для роботи над веб-сервісом.
Розділ 3. Розробка веб-сервісу, адміністративної панелі.
Розділ 4. Тестування виконаної програми, демонстрування робочих процесів.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) структурна схема системи, функціональна схема (діаграма класів), алгоритм дій.

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В. П.		

7. Дата видачі завдання «2» травня 2022 р.

Календарний план

№ П/П	Найменування етапів дипломного проекту	Терміни виконання етапів проекту	Примітки
1.	<i>Затвердження теми проекту</i>	<i>10.12.2021-18.01.2022</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>02.05.2022-13.05.2022</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>16.05.2022-27.05.2022</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>20.05.2022-03.06.2022</i>	
5.	<i>Програмна реалізація системи</i>	<i>27.05.2022-03.06.2022</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>02.05.2022-10.06.2022</i>	
7.	<i>Захист програмного продукту</i>		
8.	<i>Передзахист</i>		
9.	<i>Захист</i>	<i>21.06.2022</i>	

Студент-дипломник _____ Кирило СУШИЛЬНИКОВ
(підпис)

Керівник проекту _____ Юрій КОЧУРА
(підпис)

АНОТАЦІЯ

У даній дипломній роботі був розглянутий сервіс доставки продуктових товарів як один із найбільш актуальних веб-сервісів в сучасному світі. Було порівняно декілька відомих організацій та на основі цього розроблено новий проект. Адміністративна панель та додаток разом створюють цілу взаємозв'язану організацію, в якій працює процес доставки товарів. Результуючий проект є корисним та дещо спрощує щоденне життя звичайної людини. Побудована програма на основі мови програмування PHP, Javascript, допоміжних бібліотек та роботи бази даних.

ANNOTATION

In this bachelor's degree was considered the service of delivery of groceries as one of the most relevant web services today. Several well-known organizations were compared and a new project was developed based on this. The admin panel and the application together create a whole interconnected organization in which the process of delivery of goods works. The resulting project is useful and somewhat simplifies the daily life of the average person. The program is based on the programming language PHP, Javascript, auxiliary libraries and database operation.

справки	Формат	Значення	Найменування	Кіл. листів	№ екземпля	Додаток
			Документація загальна Знову розроблена			
	A4	ІАЛЦ.467200.002 ТЗ	Веб-сервіс організації процесу доставки продуктових товарів Технічне завдання	3		
	A4	ІАЛЦ.467200.003 ПЗ	Веб-сервіс організації процесу доставки продуктових товарів Пояснювальна записка	63		
	A4	ІАЛЦ.467200.004 Д1	Веб-сервіс організації процесу доставки продуктових товарів Структурна схема системи	1		
	A4	ІАЛЦ.4672008.005 Д2	Веб-сервіс організації процесу доставки продуктових товарів Функціональна схема (діаграма класів)	1		
	A4	ІАЛЦ.467200.006 Д3	Веб-сервіс організації процесу доставки продуктових товарів Алгоритм дій	1		
	A4	ІАЛЦ.467200.007	Веб-сервіс організації процесу доставки продуктових товарів			

					ІАЛЦ.467200.001 ОА				
Зм	Лист	№ докум.	Підп	Дата					
Розроб		Сушильников К.Д.			<i>Веб-сервіс організації процесу доставки продуктових товарів Опис альбому</i>	Літ.	Аркуш	Аркушів	
Перев		Кочура Ю.П.					1	1	
						КПІ ім. Ігоря Сікорського ФІОТ ІО-83			

**ТЕХНІЧНЕ ЗАВДАННЯ
ДО ДИПЛОМНОГО ПРОЕКТУ**

на тему: «Веб-сервіс для організації процесу доставки продуктивих товарів»

Київ – 2022

ЗМІСТ

НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
ДЖЕРЕЛА РОЗРОБКИ.....	2
ТЕХНІЧНІ ВИМОГИ.....	3
Вимоги до розробленого продукту	3
Вимоги до програмного забезпечення	3
Вимоги до апаратної частини	3
ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ.467200.002 ТЗ			
		№ докум.	Підпис	Дата				
Розробив	Сушильников К.Д.				Веб-сервіс для організації процесу доставки продуктових товарів Технічне завдання	Літ.	Аркуш	Аркушів
Перевірів	Кочура Ю.П.						1	3
Н. Контр.	Сімоненко В. П.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, Ю-83		
Затвердив								

1 НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання поширюється на розробку системи адміністрування сервісом доставки продуктових товарів.

Областю застосування цієї системи є проектування та аналіз даних сервісу, керування власним магазином (підприємством), забезпечення клієнта безпекою та гарантії, дизайн продукту.

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки даної системи є завдання для виконання роботи кваліфікаційно-освітнього рівня «бакалавр комп'ютерна інженерія», який був затверджений факультетом «Інформатики та обчислювальної техніки» кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

3 МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою та призначенням даної роботи є розробка веб-сервісу, системи адміністрування, головним дійством якої є організація процесу доставки продуктових товарів.

4 ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки даного дипломного проекту є офіційні документації, публікації та статті в мережі Інтернет на дану тему, науково-технічна література, лекційні матеріали.

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

5 ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

Розроблена система має виконувати такі вимоги:

- Простий і інтуїтивно-зрозумілий інтерфейс системи.
- Надати користувачам можливість реєстрації та авторизації.
- Надати можливість користувачам керувати власним магазином.
- Надати можливість користувачам власноруч додавати, редагувати, змінювати або видаляти продукти.
- Надати вичерпну та зрозумілу документацію для розробленого продукту.

5.2. Вимоги до програмного забезпечення

- ОС Windows, Mac чи Linux.
- Visual Studio Code 2017 IDE версії або вище.

5.3. Вимоги до апаратної частини

- ЦП не менше ніж Intel® Core (TM) 2 Duo.
- ROM не менше ніж 32 ГБ.
- RAM не менше ніж 4 ГБ.

6 ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	10.12.2021-18.01.2022
Вивчення та аналіз завдання	02.05.2022-13.05.2022
Розробка архітектури та загальної структури системи	16.05.2022-27.05.2022
Розробка структур окремих частин системи	20.05.2022-03.06.2022
Програмна реалізація системи	27.05.2022-03.06.2022
Виправлення помилок	03.06.2022-11.06.2022
Оформлення пояснювальної записки	02.05.2022-10.06.2022

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Веб-сервіс для організації процесу доставки продуктових товарів»

Київ – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ	4
ВСТУП.....	5
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	6
1.1 Внутрішня структура сервісу.....	6
1.1.1 Загальні поняття	6
1.1.2 Веб архітектура.....	8
1.2 Огляд існуючих рішень.....	10
1.2.1 Клієнтський сервіс.....	10
1.2.2 Адміністративна панель	12
1.2.3 Огляд існуючих панелей управління.....	13
ВИСНОВОК ДО ПЕРШОГО РОЗДІЛУ	17
РОЗДІЛ 2. ФУНКЦІОНАЛЬНИЙ ВИБІР ДЛЯ РОБОТИ НАД ВЕБ СЕРВІСОМ	18
2.1 Frontend частина. Огляд та аналіз функціоналу для клієнтської частини.....	18
2.1.2 HTML	18
2.1.3 CSS	20
2.1.4 Препроцесор SCSS	21
2.2 Backend частина. Огляд та аналіз серверної частини.....	22
2.2.1 Мова PHP	22
2.2.2 Мова Javascript.....	25
2.2.3 База даних MySQL.....	28
2.3 UI/UX дизайн	32
ВИСНОВОК ДО ДРУГОГО РОЗДІЛУ	34
РОЗДІЛ 3. РОЗРОБКА ВЕБ-СЕРВІСУ, АДМІНІСТРАТИВНОЇ ПАНЕЛІ.....	35
3.1 Розробка серверної логіки. OpenServer GA.....	35
3.1.1 Підключення до бази даних	37

					ІАЛЦ.467200.003 ПЗ					
Зм.	Арк.	№ докум.	Підпис	Дата						
Розробив		Сушильников К. Д.			Веб-сервіс для організації процесу доставки продуктових товарів		Літ.	Аркуш	Аркушів	
Перевірив		Кочура Ю.П.					1	63		
Реценз.							НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ПІ-73			
Н. Контр.		Сімоненко В.П.								
Затвердив										
					Пояснювальна записка					

3.1.2	Робота з базою даних.....	38
3.1.3	Розгляд форми реєстрації, авторизації	39
3.1	Реалізація клієнтської частини	43
3.2.1	Модель сервісу.....	43
3.2.2	Використання CSS, SASS, SCSS.....	44
3.2.3	Використання Javascript	48
ВИСНОВОК ДО ТРЕТЬОГО РОЗДІЛУ		49
РОЗДІЛ 4 ТЕСТУВАННЯ ВИКОНАНОЇ ПРОГРАМИ. ДЕМОНСТРАЦІЯ РОБОЧИХ ПРОЦЕСІВ		50
4.1	Інтерфейс та навігація	50
4.2	Огляд списку товарів.....	55
4.3	Вікно замовлень.....	56
4.4	Огляд інтерфейсу програми.....	57
ВИСНОВОК ДО ЧЕТВЕРТОГО РОЗДІЛУ		59
ВИСНОВКИ		60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		61
ДОДАТОК 1		64
ДОДАТОК 2		66
ДОДАТОК 3		68

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

API	(англ. Application Programming Interface) прикладний програмний інтерфейс
SOAP	(англ. Simple Object Access Protocol) протокол обміну структурованими повідомленнями в розподілених обчислювальних системах
WSDL	(англ. Web Services Description Language) Мова опису веб-сервісів, на основі мови XML
XML	(англ. eXtensible Markup Language) Розширювана мова розмітки
HTTP	(англ. HyperText Transfer Protocol) Протокол прикладного рівня передачі даних
JIT	(англ. Just-in-Time) Динамічна компіляція, технологія збільшення продуктивності програмних систем, які використовують байт-код, шляхом компіляції байт-коду в машинний код під час роботи програми
WAMP/WNMP	Акронім, який визначає набір або комплекс серверного програмного забезпечення
SQL	(англ. Structured Query Language) Мова структурованих запитів

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

На теперішній час в сучасному суспільстві з'являється все більше нових, небажаних але неминучих глобальних проблем. Однією з таких, являється пандемія, яка виникла на початку 2020 року у всьому світі. Внаслідок цього були започатковані вимушені обмеження щодо громадян. Виходячи з таких умов, своєї найбільшої популярності набули такі сервіси як онлайн-магазини, служби доставки, інтернет-зв'язок.

Дана бакалаврська робота направлена на те, щоб вирішити низку соціальних проблем та бути корисною користувачам. Готується розробка саме організації доставки продуктових товарів, керування заказами клієнтів, адміністрування веб-сервісу, насамперед.

Так як поширеність такого виду сервісів розпочалася відносно нещодавно, з'являється все більше й більше нових конкурентних рішень в цьому напрямку, які спроможні витіснити інших з ринку. Щоб стати поперед них нам необхідно проаналізувати всі переваги та недоліки схожих сервісів, робити свої висновки та реалізовувати нові нестандартні задумки.

Отже, метою розробки даного проекту було поставлено вирішення питання щоденних труднощів користувачів. Веб-сервіс повинен мати привабливий, по можливості мінімалістичний дизайн та інтуїтивно зрозумілий інтерфейс для полегшення праці з ним. Саме від зовнішнього виду залежить перше враження користувача, в подальшому - чи залишиться він з нами. Розробка повинна бути продумана наперед та мати можливість вдосконалення в планових оновленнях, мати всі гарантії щодо забезпечення безпеки в усіх можливих непередбачуваних випадках, та завжди залишатися на стороні наших клієнтів, користувачів.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Внутрішня структура сервісу

1.1.1 Загальні поняття

Веб-сервіс — програмна система, розроблена для забезпечення взаємодії між декількома комп'ютерами через мережу. У веб-додатку це зазвичай набір API, який можна використовувати через інтернет для виконання дій на віддаленому сервері, обслуговуючому веб-сервіс. Наприклад, клієнт, заснований на Flex, може викликати функції, реалізовані на сервері у PHP-додатку. У якості базового рівня протоколу використовується SOAP.

Веб-сервіс можна визначити наступними способами:

1. Це клієнт-серверна програма або компонент програми для спілкування.
2. Метод зв'язку між двома пристроями через мережу.
3. Це програмна система для взаємодії між машинами.
4. Це набір стандартів або протоколів для обміну інформацією між двома пристроями або програмою.

Нижче наведено пояснення того, як це працює:

1. Веб-сервіс функціонує як шаблон запит-відповідь, який забезпечує комунікацію між різними додатками за допомогою відкритих пакетів, наприклад, HTML, XML, WSDL і SOAP.

2. Елемент зажадає певної підтримки свого партнера, щоб бути постачальником послуг конкретного спеціаліста. На вимогу спеціаліст кооператив відреагує повідомленням реакції. Таким чином, є два повідомлення, які включають одне повідомлення запиту (XML) і одне повідомлення-відповідь (XML). Ми можемо створити веб-

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

адміністрування на основі Java щодо Solaris, яке відкривається з вашої програми Visual Basic, яка зростає попитом на Windows.

3. Адміністрація використовує XML для позначення інформації, SOAP для останнього переміщення повідомлення, WSDL для зображення доступності адміністрацій [1].

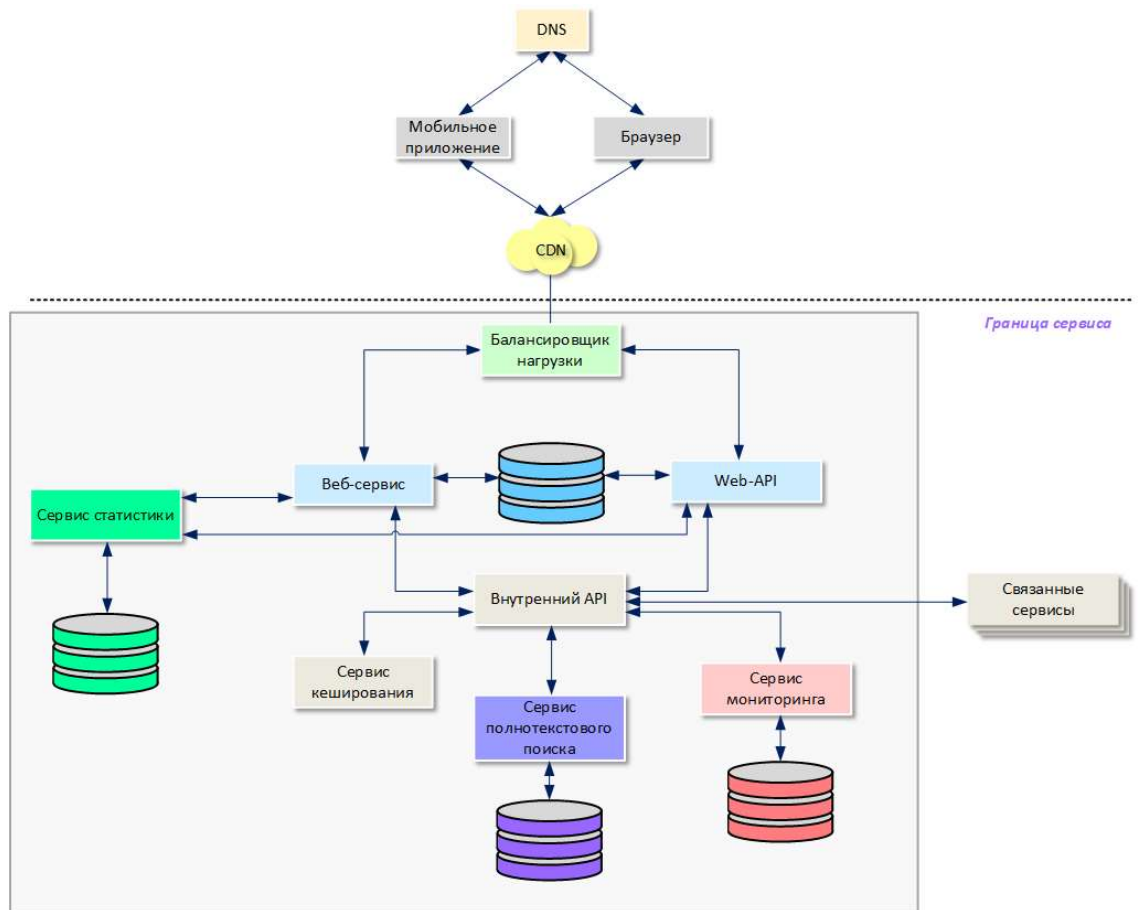


Рисунок 1.1 – Функціонування веб-сервісу [1]

На малюнку вище можна побачити функціонування веб-сервісу на найнижчому рівні за допомогою схеми.

1.1.2 Веб архітектура

Архітектура - це опис системи на найвищому рівні [1].

У будь-якому типовому веб-додатку є два різних коди (підпрограми), які працюють пліч-о-пліч. Це:

- Клієнтський код – код, який знаходиться у браузері та реагує на певний вхід користувача.

- Код на стороні сервера – код, який знаходиться на сервері і відповідає на запити HTTP.

- Веб-розробник (команда), що розробляє веб-додаток, вирішує, що робитиме код на сервері щодо коду в браузері. Для написання серверного коду використовуються C#, Java, JavaScript, Python, PHP, Ruby тощо.

Будь-який код, який може відповідати на запити HTTP, може виконуватися на сервері. Код на стороні сервера відповідає за створення сторінки, яку запитав користувач, а також за зберігання різних типів даних, включаючи профілі користувачів і введені користувачем дані. Його ніколи не бачить кінцевий користувач.

Для написання клієнтського коду використовується комбінація CSS, HTML і JavaScript. Цей код аналізується веб-браузером. На відміну від коду на стороні сервера, код на стороні клієнта можна побачити, а також змінити користувачем. Він реагує на введення користувача.

Код на стороні клієнта спілкується лише через HTTP-запити і не може зчитувати файли безпосередньо із сервера.

Веб-сервер: веб-сервер, також відомий як серверний компонент або компонент на стороні сервера, обробляє бізнес-логіку та обробляє запити користувачів, направляючи запити на правильний компонент і керуючи всіма операціями програми. Він може виконувати та контролювати запити від широкого кола клієнтів.

Сервер бази даних: сервер бази даних надає необхідні дані для програми. Він обробляє завдання, пов'язані з даними. У багаторівневій архітектурі сервери баз даних можуть керувати бізнес-логікою за допомогою збережених процедур.

Робота сучасних веб-додатків в браузері:

У сучасних додатках між клієнтом та сервером спілкування будується саме на даних, а не на відрендерених шматках розмітки. Найчастіше для такого спілкування вибирають JSON.

Зараз більшість програм працює так:

1.Клієнт робить первинний запит на сервер.

2.Сервер відповідає HTML-сторінкою, іноді з набором якихось даних, впроваджених у вигляді JS-об'єкта в кінці сторінки.

а.Сторінка може бути або намальована на сервері і тоді клієнту прийде готовий HTML.

б.Або малюванням займатиметься сам клієнт за допомогою якоїсь бібліотеки, наприклад, React. У цьому випадку від сервера надходить просто набір необхідних даних.

3.Користувач робить якусь дію, наприклад, просить відсортувати таблицю.

4.Клієнт у відповідь на цю дію вирішує, який запит надіслати на сервер, будує запит і надсилає його.

5.Сервер приймає цей запит, обробляє його та відправляє на клієнт порцію нових даних.

6.Клієнт приймає дані та перемальовує частину сторінки по них сам. Тобто він уже не замінює один шматок розмітки на інший готовий, а малює розмітку сам.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

Переваг такого спілкування (коли передаються лише дані) є декілька:

- Сервер та клієнт стають незалежними один від одного. Сервер може нічого не знати про пристрій сторінок, йому достатньо лише вміти працювати з БД та обробляти дані (первинне малювання може бути зроблене самим сервером за допомогою SSR).

- Кількість інформації, яку доводиться передавати та приймати, менша — а це зменшує обсяг трафіку.

- Логіка програми на сервері може бути простіше, тому саме логіка і клієнт стають менш залежними один від одного щодо формату даних [2].

1.2 Огляд існуючих рішень

1.2.1 Клієнтський сервіс

Які сервіси доставки товарів додому вам знайомі? Серед маркетів з доставкою продуктів одним з найпопулярніших останнім часом стає інтернет-магазин ОВО. Це чудовий спосіб отримати свіжі продукти вчасно. Але у переліку пропозицій є інші, бо попит на подібні послуги зростає, відповідно з'являється більше супермаркетів онлайн.

Серед найбільш відомих сервісів доставки продуктів в Україні виділяють такі:

1.Glovo - Іспанська фірма працює у 140 містах 20 країн світу.

Доступно доставка всього, майже в 20 містах України.

2.Zakaz.ua - Доставка продуктів із супермаркетів.

Замовлення доступні у 8 містах та з конкретних супермаркетів.

3.kabanchik.ua - Сервіс пошуку виконавців для будь-яких завдань.

Доступна практично в усіх містах України, дуже часто використовується в цілях доставки продуктових товарів з різних магазинів.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

Більш детальноше розглянемо сервіси на прикладі найкращої за рейтингом організації – Glovo.

Зареєструватися в Glovo ^

FACEBOOK

ИЛИ

Имя _____

Адрес эл. почты _____

Пароль _____

Зареєструватися с помощью эл. почты

Уже зарегистрированы? [Войти](#)

Рисунок 1.2 – Регістраційна форма додатку

Після реєстрації на сайті або заповнення форми адреси нас вітає головна сторінка, на якій ми можемо вибирати що завгодно.

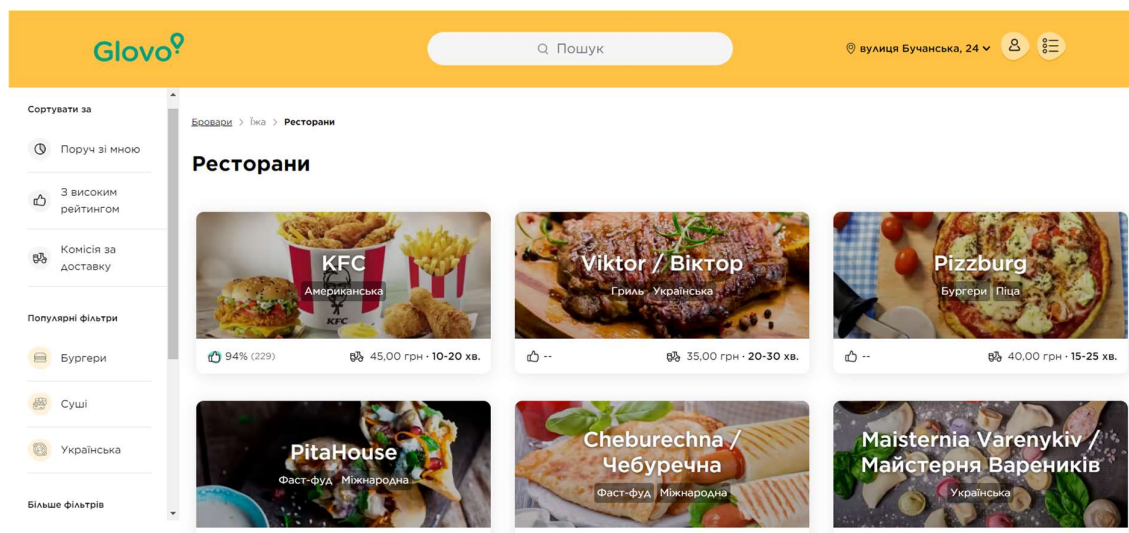


Рисунок 1.3 – Веб-інтерфейс головної сторінки Glovo

									Арк.
									11
Зм.	Арк.	№ докум.	Підпис	Дата					

В верхній шапці (header) знаходиться пошук, де ми можемо знайти будь-який потрібний нам товар, та адреса яку вводим при реєстрації та значок особистого кабінету з власною інформацією.

Ми можемо побачити slideblock з лівої сторони на якому знаходяться усі можливі сортувальні функції для зручнішого вибору продуктів. В центрі бачимо всі доступні, відсортовані магазини.

Процедура виконання замовлення:

Після завантаження програми відкрийте розділ «Управління способами оплати» та додайте дебетову/кредитну картку. Потім, вкажіть повну адресу доставки. Для цього потрібно перейти на головний екран. При цьому, можете додати промо-код, якщо, звичайно, він у вас є. Після цього потрібно буде вибрати, що саме вам потрібно замовити. Вартість доставки залежить, насамперед від відстані. Чим менша відстань, тим нижча вартість. До речі, виконання замовлення можна відстежувати у реальному часі.

Особливо примітно наявність послуги Glovo Prime. Завдяки цій послугі, ви можете отримати доступ до БЕЗКОШТОВНОЇ ДОСТАВКИ з деяких закладів. А також можливість брати участь в ексклюзивних промо-акціях. Однак, майте на увазі, що це послуга платна і доступна тільки в країні, де ви вперше оформили підписку Glovo Prime.

1.2.2 Адміністративна панель

Адміністративна панель (CMS) - панель порталу або інтернет магазину, яка допомагає здійснювати управління проектними параметрами, додавати, прибирати сторінки, видозмінювати ресурсний дизайн, редагувати контент.

Для цього нам не потрібні навички програміста, тому що як правило, такі панелі мають чудову структуру та є інтуїтивно зрозумілими для звичайного користувача, завдяки графічному інтерфейсу.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

В нашому випадку ми можемо редагувати інформацію продуктових товарів, змінювати картинки, додавати нові товари або видаляти застарілі, присутні функції додавання нових магазинів.

Переваги CMS:

1. Однією з основних переваг CMS є її спільна природа. Декілька користувачів можуть увійти в систему та внести свій вклад, запланувати або керувати вмістом для публікації. Оскільки інтерфейс зазвичай базується на браузері, до CMS може отримати доступ будь-яка кількість користувачів.
2. Другою важливою перевагою CMS є те, що вона дозволяє людям, які не мають технічних знань, які не знають мов програмування, легко створювати та керувати власним веб-контентом. Редактори типової платформи керування вмістом дозволяють користувачам вводити текст і завантажувати зображення без знання HTML або CSS (мов програмування).

Коли компанія використовує CMS для публікації своїх веб-сторінок, вона зменшує залежність від розробників інтерфейсу для внесення змін до веб-сайту, що робить публікувати нові веб-сторінки швидше та легше [19].

1.2.3 Огляд існуючих панелей управління

1.2.3.1 HostCMS

Розробляються ці панелі з 2003 року, за такий доволі довгий період розробки досягли чудових показників та мають велику популярність серед користувачів. Все це обумовлено тим, що з цих систем можна керувати як звичайними сайтами так і інтернет магазинами, все необхідне тут є.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13



Рисунок 1.4 – Вхід до адміністративної панелі

Спершу нас вітає сторінка авторизації користувача. Структура сторінки входу полягає в написанні імені користувача, та паролю. В данному прикладі також є галочка (Прив'язати сесію до IP-адресу) за допомогою якої сайт зберігає всі вписані нами данні та в випадку виходу з сайту ми можемо вернутися назад без написання данних.

Після процедури входу ми потрапляємо на головну сторінку.

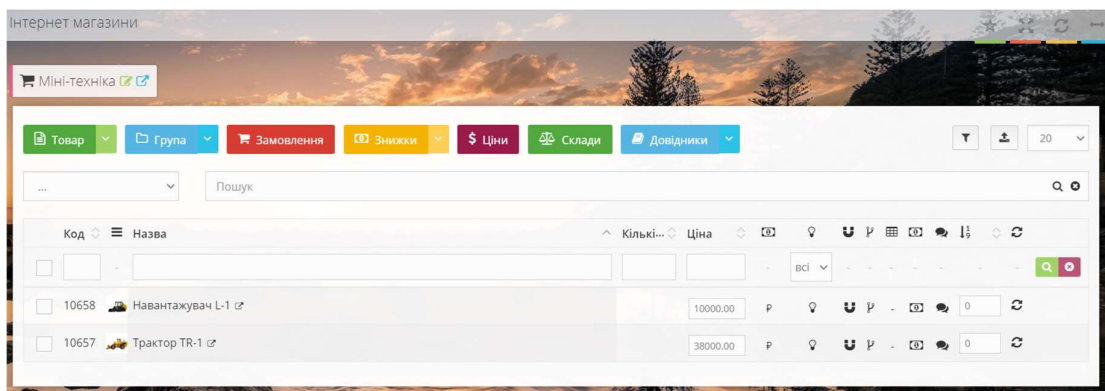


Рисунок 1.5 – Адміністративна панель HostCMS

З вище розглянутого рисунку бачимо панель управління інтернет магазином з мінімалістичним дизайном.

В таких центрах зазвичай використовується звичайна структура:

1. Каталог товару – каталог в якому ми можемо додати товар, змінити його властивості, продивитися аналітику та всі заклази клієнтів.
2. Список товарів які знаходяться на продажі. В цьому розділі, видаляємо товари, дивимосся статистику кожного з товарів, встановлюємо ціни.
3. Блок фільтрації товару або продуктів за допомогою яких, можемо швидко знайти необхідні нам речі.
4. Власний кабінет. Деякі адміністративні панелі дозволяють користувачу мати власний кабінет та коригувати власну інформацію [3].

1.2.3.2 SimplaCMS

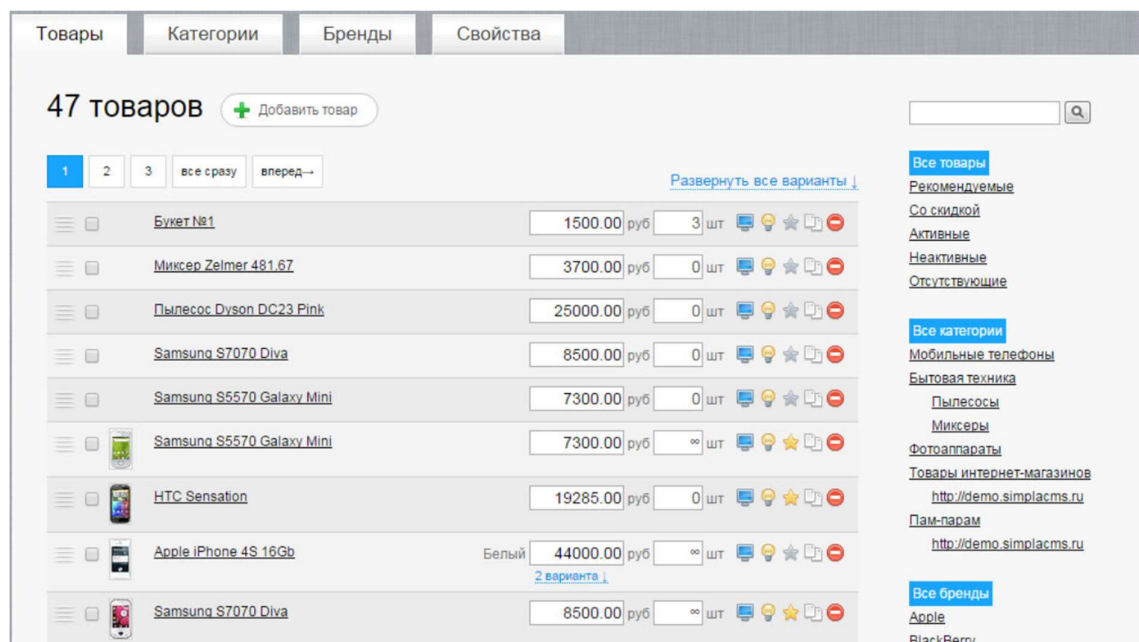


Рисунок 1.7 – Адміністративна панель SimplaCMS

Панель Simpla розробляється з 2010 року, та використовується виключно для інтернет магазинів, тому має дуже простий, зрозумілий інтерфейс та не використовує зайвих речей.

Переваги сервісу Simpla:

1. До товару можна додавати характеристики, потім використовувати їх у фільтрі. Доступне сортування за ціною/назвою.
2. Можна задавати знижку товару (тільки для конкретного товару). Знижки в залежності від суми замовлення, знижки одразу на всі товари реалізовані за купонами (наприклад, купон на 10% знижку на будь-який товар).
3. Можна виконувати варіанти товарів (наприклад, білий/чорний). Картка товару залишається одна, але в ній з'являється вибір можливого варіанту.
4. На головній сторінці можна показати "Новинки", "Хіти продажів". Також за замовчуванням налаштовано блок «Переглянуті товари».

Недоліки:

1. Порівняння товарів немає, неможливо вподобати товар та додати його до улюблених.
2. Немає можливості залишати відгуки.
3. Звичайний, непримітний дизайн [4].

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

ВИСНОВОК ДО ПЕРШОГО РОЗДІЛУ

В першому розділі детально було розглянуто поняття веб-сервісу, проаналізовано та показано графічно веб-архітектуру, її суть, основні поняття та структуру. За допомогою одного з найпопулярнішого веб-сервісу доставки продуктів – Glovo, було визначено основні переваги та недоліки таких сервісів.

Одним з досягнень даного розділу – аналіз та порівняння адміністративних панелей управління, системи, яка виконує одну з найголовніших ролей в побудові та використання веб – сервісу, в результаті чого затверджено найголовніші функції для подальшої розробки власної системи. Таким чином, переглянувши існуючі системи, доведено що розробка вибраної теми є актуальною та корисною в використанні.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

РОЗДІЛ 2. ФУНКЦІОНАЛЬНИЙ ВИБІР ДЛЯ РОБОТИ НАД ВЕБ СЕРВІСОМ

2.1 Frontend частина. Огляд та аналіз функціоналу для клієнтської частини

Основою кожного веб-сайту, веб-сервісу та інших програмних систем являється саме модель або макет який працює з клієнтом. Так як HTML та CSS є такими інструментами, розглядаємо їх безвиборчо.

2.1.2 HTML

Це мова гіпертекстової розмітки сторінки. Він використовується для того, щоб дати браузеру зрозуміти, як відображати завантажений сайт.

Є одним із найголовніших пристроїв для створення веб-застосунків та роботи в інтернеті [16].

Мова складається з тегів - це своєрідні команди, які перетворюються на візуальні об'єкти в браузері користувача. Наприклад, тег `` використовується для розміщення зображень на сторінці. Він має обов'язковий атрибут `src`, в якому вказується посилання на файл [13].

Ось так виглядатиме основний код на прикладі мого проекту:

```
<!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="stylesheet" href="/css/style.css">
  <title>diplom</title>
```

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

</head>

<body>

</head>

Принцип роботи розмітки наступний:

- Ви вводите адресний сайт або сторінки в адресний рядок.
- Браузер робить запит на цю адресу і отримує файл у форматі.html.
- Код отриманого файлу перетворюється на візуальні об'єкти.
- При цьому браузер слідує правилам перетворення, які трохи відрізняються в залежності від системи, версії та типу оглядача. Якщо в коді припущено помилку, то на сторінці може з'явитися щось несподіване.

Простими словами, HTML це каркас сайту. У ньому можуть бути прописані:

- 1.посилання;
- 2.таблиці;
- 3.зображення;
- 4.блоки;
- 5.абзаци;
- 6.форми;
- 7.заголовки тощо.

Також є обмежені можливості щодо зміни зовнішнього вигляду:

- 1.змінити колір;
- 2.вказати фонове зображення;
- 3.змінити шрифт;
- 4.зробити текст жирним, курсивним, підкресленим, закресленим тощо [6].

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

2.1.3 CSS

CSS – Cascading Style Sheets – це каскадні таблиці стилів. По суті, мова, яка відповідає за опис зовнішнього вигляду HTML-документа. Переважна більшість сучасних веб-сайтів працюють на основі HTML+CSS.

Якщо HTML структурує контент на сторінці, CSS дозволяє відформатувати його, зробити більш привабливим для читача. Спочатку веб-розробники використовували виключно HTML - так було на зорі розвитку інтернет-технологій. За допомогою розмітки можна було виділити параграф, заголовок, змінити зображення тексту. А більшого й не потрібно.

Згодом цих можливостей стало мало — і з'явилася технологія форматування без зміни вмісту та структури документа. CSS вирішив проблему "зоопарку" тегів форматування, коли різні браузері підтримували різні теги. Їх уніфікація та єдина база спростили роботу з веб-документами та полегшили життя веб-майстрам.

CSS використовується для визначення стилів (правил) оформлення документів - включаючи дизайн, верстку та варіації макета для різних пристроїв та розмірів екрану. У такого способу форматування кілька переваг:

- 1.теги не дублюються;
- 2.документ простіше обслуговувати;
- 3.Зовнішній вигляд всього сайту можна змінити централізовано, а не коригувати форматування кожної сторінки.

Стилі можна розмістити всередині тега <HEAD> або використовувати окремий файл CSS.

Ось так можна прописати CSS як атрибут безпосередньо в HTML:

```
<div class="product_name" style = 'font-size: 18px; line-height: 25px; color: #F8B505; font-weight: bold'>Яблука</div>
```

У мови CSS відносно простий синтаксис. Спочатку прописується селектор – він вибирає певний елемент на сторінці. Потім, після фігурних

дужок, вказуються властивості зі значеннями - між ними ставиться двокрапка. Самі властивості відокремлюються один від одного крапкою з комою.

Що таке селектор у CSS? Це конструкція, яка дозволяє вибрати окремі чи однотипні елементи на сторінці, щоб їх стилізувати. З селектора починається кожен блок оголошень CSS:

```
.error_psw{
  text-align: center;
  border: 2px solid red;
}
```

Докладніше обговоримо питання пріоритетів. Їхня ієрархія працює таким чином: найвищий пріоритет властивостей, в кінці оголошення яких вказано !important. Потім йдуть інлайнові стилі, що прописуються в тезі через атрибут style. Ще нижче пріоритет стилів, заданих у тезі style у самому документі. Далі слідує стилі, підключені до документа як зовнішній CSS-файл з використанням тега <link> [16].

2.1.4 Препроцесор SCSS

SCSS — це каскадні таблиці стилів.

SCSS — це препроцесор, який дозволяє використовувати функції, які ще не є частиною ширшого стандарту CSS, і забезпечує кращі робочі процеси для підтримки ваших таблиць стилів.

Переваги:

Scss можна розділити крапкою з комою і виконувати в одному рядку.

За допомогою препроцесора SCSS ми можемо зменшити кількість повторень і переконатися, що ми пишемо чистий, підтримуваний код на майбутнє.

Scss може приймати css-код і працювати.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

Він є повністю сумісний із синтаксисом CSS, але підтримує всю потужність Sass.

Scss є розширенням синтаксису CSS. Це означає, що кожна дійсна таблиця стилів CSS є дійсним файлом SCSS з тим же значенням. Крім того, SCSS розуміє більшість хаків CSS і спеціальний синтаксис постачальника, наприклад старий синтаксис фільтрів IE. Цей синтаксис доповнюється функціями Sass, описаними нижче. Файли, що використовують цей синтаксис, мають розширення .scss.

Сухий (без повторень) код набагато краще, ніж мокрий код (писати кожен раз) [25].

2.2 Backend частина. Огляд та аналіз серверної частини

2.2.1 Мова PHP

На PHP працює близько 78.2% всіх веб-сайтів . Мова вперше була представлена у 1995 році, коли для створення динамічних сайтів існувало не так багато можливостей.

PHP користується дуже великою популярністю серед інших, завдяки своїй простоті та універсальності.

Виходячи з усього, для свого проекту я обрав саме цю мову програмування.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

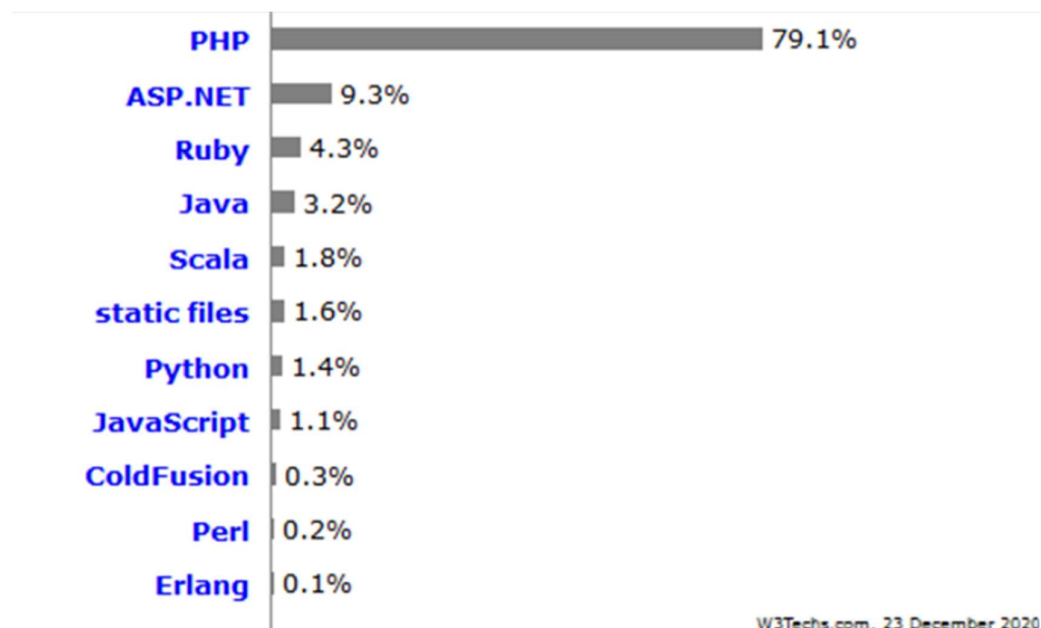


Рисунок 2.1 – Рейтинг серверних мов програмування [8]

PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor) - це поширена мова програмування загального призначення з відкритим кодом. PHP спеціально сконструйований для веб-розробок, і його код може впроваджуватися безпосередньо в HTML.

Проста відповідь, але що вона може означати? Ось приклад коду:

```
<?php
    session_start();
    if($_SESSION['profile'] ) {
        header("Location: profile.php");
    }
?>
```

Замість рутинного виведення HTML-коду командами мови (як це відбувається, наприклад, Perl або C), скрипт PHP містить HTML з вбудованим кодом. Код PHP відокремлюється спеціальними початковими та кінцевими тегамі <?php і ?>, які дозволяють "перемикатися" в "PHP-режим" і виходити з нього.

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері та генерують HTML, який надсилається клієнту. Якби у вас на сервері був розміщений скрипт, подібний до вищенаведеного, клієнт отримав би тільки результат його виконання, але не зміг би з'ясувати, який саме код зробив. Ви навіть можете налаштувати свій сервер таким чином, щоб звичайні HTML-файли оброблялися процесором PHP, тому клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта [17].

Переваги PHP:

1. Міжплатформенність: PHP не залежить від платформи. Вам не потрібно мати певну ОС, щоб використовувати її, оскільки вона працює на кожній платформі, будь то Mac, Windows чи Linux.

2. Open Source: PHP є відкритим вихідним кодом. Оригінальний код доступний для всіх, хто хоче розробити його. Це одна з причин, чому одна з її фреймворків, Laravel, настільки популярна.

3. Легко освоїти: PHP не важко вивчити для абсолютних початківців. Ви можете отримати це добре, якщо у вас вже є знання програмування.

4. PHP синхронізується з усіма базами даних: ви можете легко підключити PHP до всіх баз даних, реляційних і нереляційних. Тому він може швидко підключитися до MySQL, Postgress, MongoDB або будь-якої іншої бази даних.

5. Підтримуюча спільнота: PHP має дуже сприятливу онлайн-спільноту. Офіційна документація містить інструкції щодо використання функцій, і ви можете легко вирішити проблему, коли вона застрягла.

Також PHP як і багато інших мов програмування має свої недоліки:

1. Не підходить для створення настільних додатків або системних компонентів;

2. Має слабкі засоби для роботи з виключеннями;

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

3.Глобальні параметри конфігурації впливають на базовий синтаксис мови, що затрудняє настройку сервера і розвертання додатків;

4.Об'єкти передаються за значенням, що смущає багатьох програмістів, прив'язаних до передачі об'єктів по ссилку, як це робиться в більшості інших мов;

5.Веб-прикладні, написані на PHP, часто мають проблеми з безпекою [24].

2.2.2 Мова Javascript

JavaScript – це динамічна мова програмування, яка використовується для веб-розробки, у веб-додатках, для розробки ігор та багато іншого. Він дозволяє реалізувати динамічні функції на веб-сторінках, які неможливо зробити лише за допомогою HTML і CSS [9].

Багато браузерів використовують JavaScript як мову сценаріїв для виконання динамічних речей в Інтернеті. Кожного разу, коли ви бачите спадне меню «Показати при натисканні», додатковий вміст, доданий на сторінку, і динамічно змінювані кольори елементів на сторінці, зокрема деякі функції, ви бачите наслідки JavaScript.

Без JavaScript у вас в Інтернеті були б лише HTML і CSS. Саме вони обмежують вас кількома реалізаціями веб-сторінок. 90% (якщо не більше) ваших веб-сторінок будуть статичними, і ви матимете лише динамічні зміни, як-от анімації, які надає CSS.

HTML визначає структуру вашого веб-документа та вміст у ньому. CSS оголошує різні стилі для вмісту веб-документа.

HTML і CSS часто називають мовами розмітки, а не мовами програмування, оскільки вони, по суті, забезпечують розмітку для документів з дуже невеликим динамізмом.

JavaScript, з іншого боку, є динамічною мовою програмування, яка підтримує математичні обчислення, дозволяє динамічно додавати вміст HTML

до DOM, створює динамічні оголошення стилів, витягує вміст з іншого веб-сайту та багато іншого [21].

Як приклад, я візьму декілька рядків коду з свого проекту:

```
<script>
    function(){
        alert("Error")
    }
</script>
```

Завдяки тегу `<script></script>` ми можемо використовувати мову Javascript прямо в документі `.html`. Або в кращому випадку, потрібно підключати в окремий файл за допомогою тегу `<script src="/script.js"></script>` в документі `.html`

Переваги Javascript:

1.Швидкість - JavaScript має тенденцію бути дуже швидким, оскільки він часто запускається відразу в браузері клієнта. Поки він не вимагає зовнішніх ресурсів, JavaScript не сповільнюється через виклики до серверного сервера. Крім того, всі основні браузери підтримують компіляцію JIT (вчасно) для JavaScript, що означає, що немає необхідності компілювати код перед його запуском.

2.Простота - синтаксис JavaScript був натхненний Java і його порівняно легко вивчити порівняно з іншими популярними мовами, такими як C ++.

3.Популярність - JavaScript є скрізь в Інтернеті, і з появою Node.js все частіше використовується на серверній основі. Існує незліченна кількість ресурсів для вивчення JavaScript. Як StackOverflow, так і GitHub демонструють дедалі більшу кількість проектів, які використовують JavaScript, і привабливість, яку вона здобула за останні роки, очікується лише збільшення.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

4.Навантаження сервера - JavaScript працює на стороні клієнта, тому загалом зменшує попит на сервери, а простим програмам сервер може взагалі не знадобитися.

5.Розширені інтерфейси - JavaScript можна використовувати для створення таких функцій, як перетягування та компонентів, таких як повзунки, що все значно покращує користувальницький інтерфейс та досвід роботи на сайті.

6.Розширена функціональність - Розробники можуть розширити функціональність веб-сторінок, написавши фрагменти JavaScript для сторонніх доповнень, таких як Greasemonkey.

7.Універсальність - Є багато способів використовувати JavaScript через сервери Node.js. Якщо ви хотіли завантажити Node.js за допомогою Express, використовувати базу даних документів, як MongoDB, і використовувати JavaScript на інтерфейсі для клієнтів, можна розробити цілу програму JavaScript спереду назад, використовуючи лише JavaScript.

8.Оновлення - З моменту появи ECMAScript 5 (специфікація сценаріїв, на яку покладається JavaScript), ECMA International займається оновленням JavaScript щорічно. Наразі ми отримали підтримку браузера для ES6 у 2017 році і сподіваємось на підтримку ES7 у майбутньому.

Недоліки:

1.Захист на стороні клієнта - оскільки код JavaScript виконується на стороні клієнта, помилки та недогляди іноді можуть бути використані для зловмисних цілей. Через це деякі люди вирішують повністю вимкнути JavaScript.

2.Підтримка браузера - Хоча сценарії на стороні сервера завжди дають однакові результати, різні браузери іноді інтерпретують код JavaScript по-різному. У наш час відмінності мінімальні, і вам не слід

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

турбуватися про це, поки ви тестуєте свій сценарій у всіх основних браузерах [7].

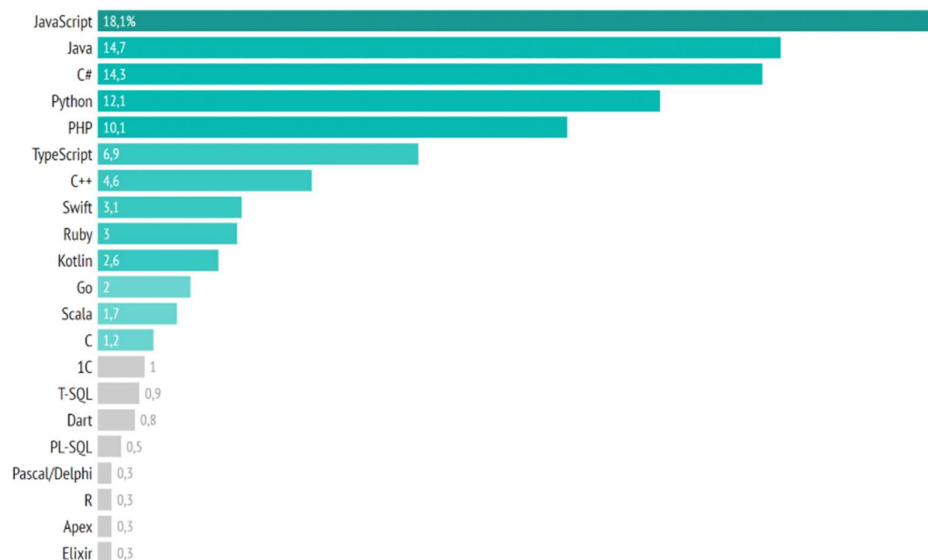


Рисунок 2.2 – Рейтинг мов програмування на 2021 рік [12]

2.2.3 База даних MySQL

Види баз даних:

1. Реляційна база даних - це тип сховища даних, що впорядковує дані таблиці, пов'язані друг з одним. Структурована мова запитів є основою цих систем, оскільки використовується для зв'язку та управління. СУБД мають зумовлену схему, тобто дані зберігаються в рядках (записах) та стовпцях (атрибутах) із суворю структурою. Кожен запис зазвичай містить значення для кожного атрибута, що призводить до чітких залежностей між різними точками даних.

2. Нереляційна база даних (БД) - це таблична база даних, яка використовує різні моделі даних для зберігання та управління. Тут, на відміну більшості традиційних СУБД, не використовується таблична схема рядків і стовпців, а застосовується модель, оптимізована під конкретні вимоги типу

даних, що зберігаються. Наприклад, дані можуть зберігатися як прості пари "ключ - значення", документи JSON або граф, що складається з ребер та верш [19].

MySQL - одна з найпопулярніших реляційних баз даних. Спочатку випущена як рішення з відкритим вихідним кодом, MySQL тепер належить корпорації Oracle. Сьогодні MySQL є основою класичного серверного веб-оточення - стеків LAMP та LEMP (Linux, Apache/nginx, MySQL та Perl/PHP/Python). Написаний на C та C++ MySQL, добре працює з такими платформами, як Linux, Windows, MacOS, IRIX та іншими.

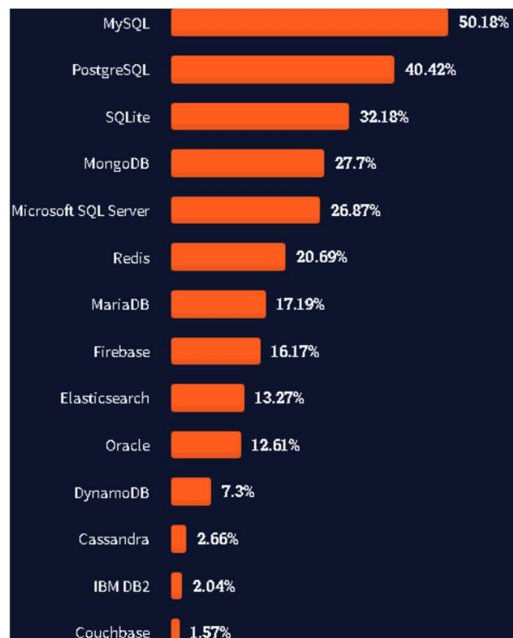


Рисунок 2.3 – Рейтинг баз даних на 2022 рік [5]

Це система управління реляційною базою даних з відкритим вихідним кодом (RDBMS) з моделлю клієнт-сервер.

СУБД — це програмне забезпечення або служба, що використовується для створення та керування базами даних на основі реляційної моделі.

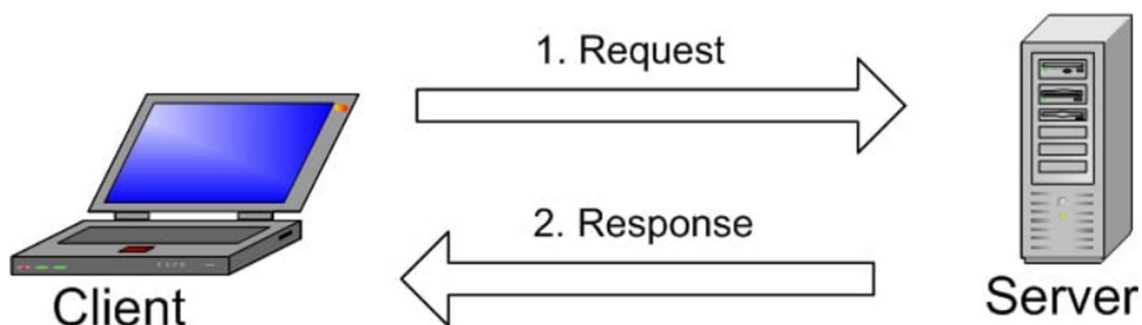


Рисунок 2.4 – Розглянуто зв’язок клієнт-сервер, запити до бази даних [14]

Зображення пояснює основну структуру структури клієнт-сервер. Один або кілька пристроїв (клієнтів) підключаються до сервера через певну мережу. Кожен клієнт може зробити запит із графічного інтерфейсу користувача (GUI) на своїх екранах, і сервер видаватиме потрібний результат, якщо обидві сторони розуміють інструкцію. Не будучи занадто технічним, основні процеси, що відбуваються в середовищі MySQL, однакові, а саме:

1. MySQL створює базу даних для зберігання та маніпулювання даними, визначаючи взаємозв’язок кожної таблиці.

2. Клієнти можуть робити запити, вводючи певні оператори SQL на MySQL.

3. Серверна програма відповідає запитаною інформацією, і вона з’явиться на стороні клієнта.

З боку клієнтів, зазвичай підкреслюють, який графічний інтерфейс MySQL використовувати. Чим легшим і зручнішим є графічний інтерфейс користувача, тим швидшим і легшим буде їхнє керування даними. Одними з найпопулярніших графічних інтерфейсів MySQL є MySQL WorkBench, SequelPro, DBVisualizer і інструмент адміністратора Navicat DB. Деякі з них безкоштовні, деякі є комерційними, деякі працюють виключно для macOS, а деякі сумісні з основними операційними системами. Клієнти повинні вибирати

графічний інтерфейс відповідно до своїх потреб. Для керування веб-базами, включаючи сайт WordPress, найочевиднішим є phpMyAdmin [19].

Переваги Mysql:

1. Гнучкий і простий у використанні

2. Ви можете змінити вихідний код, щоб відповідати вашим очікуванням, і вам не потрібно нічого платити за цей рівень свободи, включаючи варіанти оновлення до розширеної комерційної версії. Процес встановлення відносно простий і не повинен тривати більше 30 хвилин.

3. Висока продуктивність

4. Широкий набір кластерних серверів підтримує MySQL. Незалежно від того, зберігаєте ви величезні обсяги великих даних електронної комерції або займаєтеся серйозною бізнес-аналітикою, MySQL може допомогти вам плавно з оптимальною швидкістю.

5. Промисловий стандарт індустрії роками використовують MySQL, а це означає, що для кваліфікованих розробників є багато ресурсів. Користувачі MySQL можуть очікувати швидкого розвитку програмного забезпечення та позаштатних експертів, які бажають працювати за меншу зарплату, якщо вони коли-небудь знадобляться.

6. Безпечний. Ваші дані повинні бути вашою основною турботою при виборі правильного програмного забезпечення СУБД. Завдяки своїй системі привілеїв доступу та керування обліковими записами користувачів, MySQL встановлює високий рівень безпеки. Доступна перевірка на основі хоста та шифрування пароля.

Недоліки:

1. Недостатня надійність. У питаннях надійності деяких процесів по роботі з даними (наприклад, зв'язок, транзакції, аудит) MySQL поступається деяким іншим СУБД.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

2. Низька швидкість розробки.

Серед всіх відомих баз даних була обрана саме ця, як система, яка зарекомендувала себе на протязі багатьох років, будучи використаною у великій кількості відомих проектів [14].

2.3 UI/UX дизайн

Проект був створений за принципами UI/UX-дизайну, тому розглянемо його як одну з найбільш головних речей в інтерфейсі більш детально.

UI-дизайн та UX-дизайн – це два терміни, які найчастіше плутають і змішують у веб-дизайні та дизайні додатків. І це зрозуміло. Зазвичай їх поміщають разом в один термін, UI/UX-дизайн, і якщо дивитися з поверхні, здається, що вони описують одне і те ж. Часто важко знайти чіткі описи двох, які б не спускалися занадто далеко в жаргон.

«Інтерфейс користувача» (UI) в дизайні інтерфейсу користувача означає «користувацький інтерфейс». Інтерфейс користувача — це графічний макет програми. Він складається з кнопок, на які натискають користувачі, тексту, який вони читають, зображень, повзунків, полів для введення тексту та всіх інших елементів, з якими взаємодіє користувач. Сюди входить макет екрана, переходи, анімація інтерфейсу та кожна мікро-взаємодія. Будь-який візуальний елемент, взаємодія чи анімація мають бути розроблені.

«UX» означає «користувацький досвід». Взаємодія користувача з додатком визначається тим, як вони з ним взаємодіють. Чи є досвід гладким та інтуїтивним чи незграбним і заплутаним? Навігація додатком виглядає логічною чи довільною? Чи дає людям відчуття, що взаємодія з додатком ефективно вирішує поставлені завдання, чи це схоже на боротьбу? Досвід користувача визначається тим, наскільки легко чи важко взаємодіяти з

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

елементами інтерфейсу користувача, створеними дизайнерами інтерфейсу користувача.

У дизайні UX також є певна кількість ітераційного аналізу. Завдяки UX створюється каркасний рендеринг інтерфейсних взаємодій і отримуються відгуки користувачів. Дизайнери інтегрують це у свої проекти. Для UX-дизайнерів важливо мати цілісне розуміння того, як користувачі вважають за краще взаємодіяти зі своїми додатками [15].

Одним із головних задач UX/UI дизайну розробити інтерфейс для різних пристроїв. Адаптивний дизайн — дизайн вашого веб-сайту чи програми з плавним інтерфейсом для різних розмірів пристроїв — був популярним протягом останніх кількох років. Якщо ви ще не користуєтеся адаптивним дизайном, ви, напевно, чули, що це важливо, але чи дійсно це має значення? Дослідження продовжують доводити, що це так.

Однак більшість дизайнерських компаній, у тому числі ми спочатку, створювали веб-сайти так, щоб вони були чуйними, але розмір зображень і тексту іноді був пропорційним. Користувачі мають усі типи пристроїв з будь-яким розміром екрана, тому ви, звичайно, хочете, щоб ваш сайт або додаток виглядали добре всюди. Існує різниця між адаптивним дизайном і адаптивним дизайном, який виглядає добре. Тому, поставивши прапорець, подивіться на свій адаптивний дизайн на мобільному або планшетному пристрої та запитайте себе, чи гарний він [18].

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

ВИСНОВОК ДО ДРУГОГО РОЗДІЛУ

У вище розглянутому розділі було розроблено веб-сервіс, панель адміністрування та наглядно продемонстровано усі основні моменти з розробки. Задачу розділено на три етапи - backend та frontend частини роботи та аналіз дизайну сервісу.

Перший етап розробки виконаний за допомогою мови PHP, підключення до бази даних MySQL, написані головні принципи зв'язку серверу, показано головні скрипти.

В другому етапі розроблена клієнтська частина, а саме, веб-інтерфейс, функціонал сервісу.

Третій етап полягає в аналізі дизайну сайту, розглянуті принципи та основи UI/UX дизайну.

Наведено усі налаштування допоміжних програм за використанням яких велася розробка. Детально розібрано усі алгоритми рішень, методи та функції кодової частини програми.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

РОЗДІЛ 3. РОЗРОБКА ВЕБ-СЕРВІСУ, АДМІНІСТРАТИВНОЇ ПАНЕЛІ

В попередньому розділі було проаналізовано та вибрано саме ті мови програмування та компоненти, як будуть описані в цьому розділі більш детально та вже з власної розробки. Продемонстровано роботу локального серверу, налаштування та підключення до бази даних, виконання алгоритмів, необхідних скриптів.

3.1 Розробка серверної логіки. OpenServer

OpenServer - це портативний локальний сервер WAMP/WNMP, який має велику багатофункціональну програму та великий вибір компонентів, що підключається. OSPanel широко використовується з метою розробки, нарощування та тестування веб-проектів, а також для надання веб-сервісів у локальних масштабах.

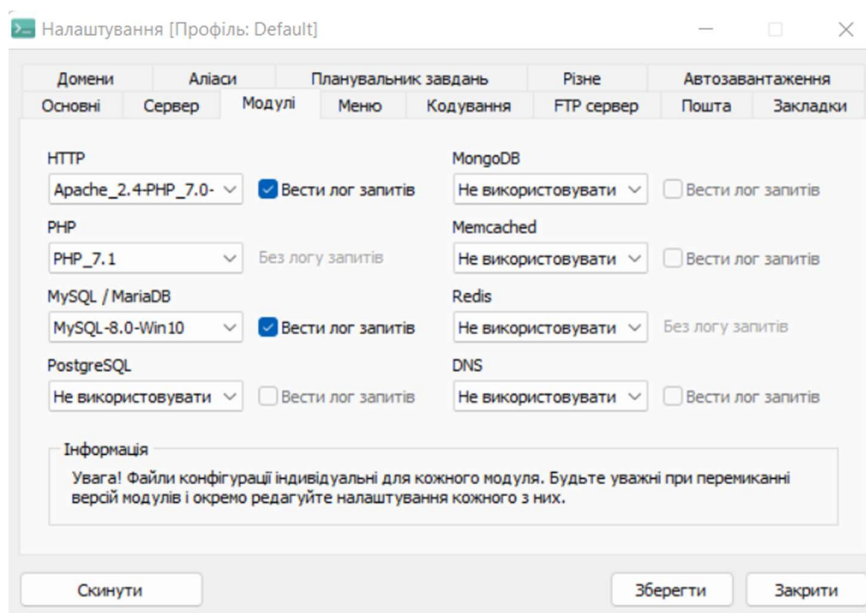


Рисунок 3.1 – Розглянемо панель налаштування OSPanel

Після установки серверу, потрібно занести деякі налаштування для подальшої коректної праці.

Для роботи нам необхідно перейти в налаштуваннях до вкладки «Модулі», та обрати встановлену нами раніше версію PHP та MySQL. Після вдалого налаштування можемо відкривати та запускати сервер. Далі використовуємо phpMyAdmin.

phpMyAdmin — це безкоштовний програмний інструмент, написаний на PHP, призначений для адміністрування MySQL через Інтернет. phpMyAdmin підтримує широкий спектр операцій на MySQL і MariaDB. Часто використовувані операції (керування базами даних, таблицями, стовпцями, зв'язками, індексами, користувачами, дозволами тощо) можна виконувати через інтерфейс користувача, при цьому ви все ще маєте можливість безпосередньо виконувати будь-який оператор SQL.

Чому доцільно використовувати додаток phpMyAdmin:

1.Витончене обслуговування, без додаткових витрат, крім тих, які будуть витрачені на володіння самими системами керування базами даних.

2.Підтримує та діє гнучко для більшості часто використовуваних форматів файлів, що стане в нагоді для документації під час роботи над системами керування базами даних.

3.Окремі панелі для маніпуляцій з базою даних, редагування запитів SQL, відстеження статусу тощо.

4.Допомагає відображати всі активні плагіни з підключених баз даних.

Недостаток:

Підтримує лише бази даних такі як, MySQL, MariaDB [10].

					ІАЛЦ.467200.003 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Имя	Дата изменения	Тип	Размер
#source	11.05.2022 17:03	Папка с файлами	
css	24.05.2022 18:47	Папка с файлами	
img	06.06.2022 21:37	Папка с файлами	
js	05.06.2022 23:12	Папка с файлами	
vendor	24.05.2022 17:07	Папка с файлами	
index	24.05.2022 17:18	Исходный файл Р...	2 КБ
prepros	06.06.2022 21:54	Исходный файл С...	23 КБ
profile	06.06.2022 21:42	Исходный файл Р...	20 КБ
register	24.05.2022 17:18	Исходный файл Р...	3 КБ

Рисунок 3.2 – Остаточний вигляд каталогу файлів

З розглянутого рисунку вище можемо бачити що каталог знаходиться у папці /OpenServer/domains, потрібне для успішної роботи на локальному сервері. Папка CSS, в якій зібрані усі файли з розширенням .css. Папка img включає в себе усі картинки з розробленого дизайну, або завантажені файли. У папці vender ми можемо знайти усі файли необхідні для реалізації серверної бізнес-логіки. Всі інші файли виконані у форматі .php, мають головну роль в побудові проекту.

3.1.1 Підключення до бази даних

Підключення до бази даних MySQL відбувається за допомогою мови програмування PHP.

У файлі connect.php, змінній «\$conn» присвоюємо функцію mysql_connect(), яка встановлює з'єднання з сервером MySQL. Так як я використовую локальний хост, всі параметри є за замовчуванням. Адрес бази даних – «127.0.0.1», назва – «diplom», ім'я користувача – «root», пароль – «».

```

<?php
// $servername = "127.0.0.1";
// $database = "diplom";
// $username = "root";
// $password = "";

$conn = mysqli_connect('127.0.0.1', 'root', '', 'diplom');

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// echo "Connected success";
// mysqli_close($conn);
?>

```

Рисунок 3.3 – Файл connect.php

Після успішного підключення, ми можемо працювати з базою, додавати та редагувати різну інформацію.

3.1.2 Робота з базою даних

Для проекту, знадобилось створити 4 колонки інформації:

1. id – номер користувача.
2. name_company – назва компанії.
3. tel_number – номер телефону.
4. password – пароль.
5. email – електронна пошта.
6. photo - фото.

+ Параметры

	id	name_company	tel_number	password	email	photo
<input type="checkbox"/>	19	Кирилл Сушильников	+380958345258	123412	NULL	NULL





↑ Отметить все С отмеченными:    

Рисунок 3.4 – Табличка створених даних

На показаному вище скріншоті, можемо побачити що після заповнення форми реєстрації та відправлення даних в базу, пароль залишається незмінним, що є дуже небезпечним, та в таких випадках дуже швидко може бути зламаним.

Для вирішення цього питання, була використана функція md5(), як одна з найлегших функцій шифрування, яка підходить для початку роботи з невеличким сервісом.

Функція md5() обчислює хеш рядка MD5.

Функція md5() використовує алгоритм RSA Data Security, Inc. MD5 Message-Digest.



Рисунок 3.5 – Наслідок хешування функції md5()

Недоліки такого алгоритму виходять з того, що він є відносно простий та небезпечний для великих проєктів.

3.1.3 Розгляд форми реєстрації, авторизації

Насамперед розглянемо форму авторизації за допомогою використання PHP скриптів.

Слід починати з функції session_start(), так як ми власноруч запускаємо сесію для початку процесу, в якому користувач та сервер обмінюються даними. Якщо клієнт не здійснює ніяких дій на протязі певного періоду, сесія автоматично завершується.

Для роботи з сервером використовуються два HTTP-методи:

1. GET-запит. Використовується для зчитування даних з сайту. Наприклад для доступу на сторінку, яку хоче прочитати клієнт.
2. POST-запит. Метод відправки даних на сайт. Найчастіше його застосовують для передачі форм до серверу.

```
<?php
session_start();
require_once 'connect.php';

$name_company = $_POST['name_company'];
$tel_number = $_POST['tel_number'];
$password = $_POST['password'];
$password_conf = $_POST['password_conf'];

if ($password !== $password_conf) {
    $_SESSION['error'] = 'Паролі не співпадають';
    header("Location: ../register.php");
}else{

    $hash_password = md5($password);

    mysqli_query($conn, "INSERT INTO `users` (`id`, `name_company`,
`tel_number`, `password`) VALUES (NULL, '$name_company', '$tel_number', '$hash_password')");
    $_SESSION['error'] = 'Регістрація виконана';
    header("Location: ../index.php");
}

?>
```

Рисунок 3.6 – Код для відправлення форми

За допомогою методу запиту POST, надсилаємо данні користувача, а саме «пароль», «підтвердження паролю», «номер телефону», «назва компанії», до веб-серверу для їхнього подальшого зберігання.

Для зберігання даних сеансу використовуємо суперглобальну змінну `$_SESSION`, внаслідок чого вони зберігаються в файлі сесії на сервері, який був створений під час першого запуску сесії. Таким чином можемо використовувати данні сеансу разом з декількома запитами.

Функція `mysqli_query()` виконує запис до бази даних. Вираз `INSERT INTO` забезпечує нам додавання всіх наших записів до бази даних. Складаємо

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

коректний запис SQL-запит на вставку запису з усіма даними, а потім виконуємо його шляхом передачі запиту до функції `mysqli_query()`, та додаємо в таблицю. Першим параметром функції `mysqli_query()` завжди слугує ресурс підключення, який знаходиться в «conn», другим параметром є сам SQL-запит. Щоб переконатися в правильності роботи нашого запиту без помилок використовується функція `mysqli_error()`. В випадку успішної роботи, суперглобальна змінна вертає нас до сесії авторизації.

```
<?php
session_start();
require_once 'connect.php';

$tel_number = $_POST['tel_number'];
$password = md5($_POST['password']);

$user_check = mysqli_query($conn, "SELECT * FROM `users` WHERE
`tel_number` = '$tel_number' AND `password` = '$password'");
if (mysqli_num_rows($user_check) > 0){

    $profile = mysqli_fetch_assoc($user_check);

    $_SESSION['profile'] = [
        'id' => $profile['id'],
        'tel_number' => $profile['tel_number'],
        'name_company' => $profile['name_company']
    ];

    header("Location: ../profile.php");
}
else{
    $_SESSION['error'] = 'Невірний логін або пароль';
    header("Location: ../index.php");
}
?>
```

Рисунок 3.7 – Код сесії авторизації

Інша частина операція при роботі з базами даних в PHP - це отримання записів з таблиць (запити типу SELECT). Спочатку ми заново записуємо запит до SQL, який використовує оператор SELECT. Потім запускаємо функцію

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

`mysqli_query()`, щоб отримати дані таблиці. Вибираємо з таблиці «`tel_number`» та «`password`». Функція `Mysqli_num_rows ()` повертає кількість рядків у наборі результатів. За допомогою перевірки `if(){}else{}` визначаємо, чи є задані параметри в нашій таблиці. Якщо перевірка успішна, виконується вхід до системи.

Функція `mysqli_fetch_assoc ()` використовується для повернення асоціативного масиву, де наступний рядок набору результатів представляє результат з параметром результату, де кожен ключ у масиві представляє назву одного зі стовпців набору результатів [23].

Функція `header ()` — це вбудована функція в PHP, яка використовується для надсилання необробленого заголовка HTTP. HTTP — це функції, які обробляють дані, надіслані клієнту або браузеру через веб-сервер, перш ніж надсилати будь-який інший вихід. Функція заголовка PHP () надсилає заголовок HTTP клієнту або браузеру. Перш ніж надсилати HTML, XML, JSON або інші дані браузеру або клієнту, вихідні дані надсилаються із запитом (особливо HTTP-запитом), який сервер оброблює як інформацію заголовка. Заголовок HTTP надає необхідну інформацію про об'єкт оснований в тексті повідомлення, а саме про запит і відповідь.

Ця функція приймає три параметри, як описано нижче:

1. `$header`: цей параметр зберігає рядок заголовка. Існує два типи викликів заголовків. Перший заголовок починається з рядка "HTTP/", який використовується для знаходження коду статусу HTTP для надсилання. Ще одна умова заголовка — «Розташування:». Це необхідний параметр.

2. `$replace`: це необов'язковий варіант. Це означає, що заголовок має бути зміненим на попередню назву або додати іншу назву. Значенням за замовчуванням є `True` (замінено). Якщо заміна дорівнює `False`, вона змушує використовувати декілька заголовків одного типу.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

3. \$ http_response_code: це необов'язковий параметр. Він примушує код відповіді HTTP до встановленого значення [11, 22].

В кінці, для примусово виходу із сесії, скасовуємо поточну суперглобальну змінну, за допомогою:

```
<?php
    session_start();
    unset($_SESSION['profile']);
    header("Location: ../index.php");
?>
```

Функція unset () є вбудованою функцією в PHP, яка використовується для заміни встановленої змінної. Поведінка функції залежить від різних умов. Якщо функція викликається з визначеної користувачем функції, вона скасовує значення, пов'язане зі змінними в ній, а значення, встановлене поза нею, залишається.

Це означає, що ця функція знімає лише локальну змінну. Якщо ми хочемо скасувати встановлення глобальної змінної всередині функції, то для цього потрібно використовувати масив \$GLOBALS [11].

3.2 Реалізація клієнтської частини

3.2.1 Модель сервісу

В основі входу в адміністративну панель лежить мова розмітки HTML разом з скриптами PHP для зв'язності серверу. Під час розробки клієнтської частини головна увага приділяється розробці інтерфейсу, зручності у користуванні подальшим юзерам. Саме в цьому лягає суть цієї частини.

Мова розмітки документів складається з тегів, які складають цілу систему та атрибутів.

```

<div class="wrapper">
  <form action="vendor/signin.php" method="post" enctype='multipart/form-data'>
    <h1>Вхід</h1>
    <label>Введіть номер телефону</label>
    <input type="tel_number" name='tel_number' required>
    <label>Введіть пароль</label>
    <input type="password" name='password' id="" required>
    <div class="registration">Ще немає профілю? <span style="font-weight:bold;color: black;">
      <a href="./register.php">Зареєструйтеся</a></span>
    </div>
    <button type='submit'>Зайти</button>
  <?php
    if ($_SESSION['error']) {
      echo '<p class="error_psw"> ' . $_SESSION['error'] . ' </p>';
    }
    unset($_SESSION['error']);
  <?>
</form>
<div class="logo"></div>
<div class="img_back"></div>
</div>

```

Рисунок 3.4 – Форма входу мовою HTML

Теги складається з полів, які разом утворюють авторизаційну форму. В головній формі, тегу <form>, приймаємо атрибут типу method='post', через запити POST на сервері. Також слід писати атрибут enctype='multipart/form-data' через те, що в нашій формі будуть присутні поля для відправки файлів (input type="file"). Це є одним із найскладніших методів кодування які існують (методи: application/x-www-form-urlencoded, 'multipart/form-data', text/plain), але не в перевагу він дозволяє включати цілі файли у дані.

3.2.2 Використання CSS, SASS, SCSS

CSS – каскадна таблиця стилів, мова опису зовнішнього виду документа, за рахунок якої відбувається покращення інтерфейсу та налаштування сайту для більш кращого сприйняття очами користувача продукту.

SCSS (Sassy CSS) - це більш просунута версія CSS. Завдяки препроцесору CSS можемо легко писати код, який дозволяє нам використовувати цикли, функції, імпорт, змінні та математичні операції, що робить написання CSS більш потужним.

									Арк.
									44
Зм.	Арк.	№ докум.	Підпис	Дата					

В своєму проєкті для опису зовнішнього вигляду документу саме препроцесор SCSS, так як цей метод є зручнішим та більш швидшим в використанні.

Так як браузер читає лише CSS-код, необхідно було обрати програмне забезпечення яке компілює код препроцесору в потрібний нам. Для таких задач чудово підійшов компілятор Prepros.

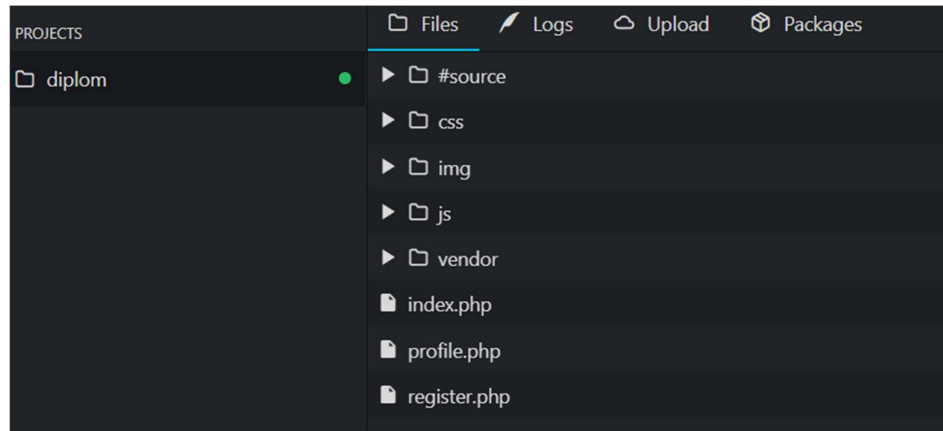


Рисунок 3.5 – Загальний вигляд Prepros

Перед роботою з програмою, в налаштуваннях потрібно було вибрати саме той інструмент, з яким ми будемо працювати.

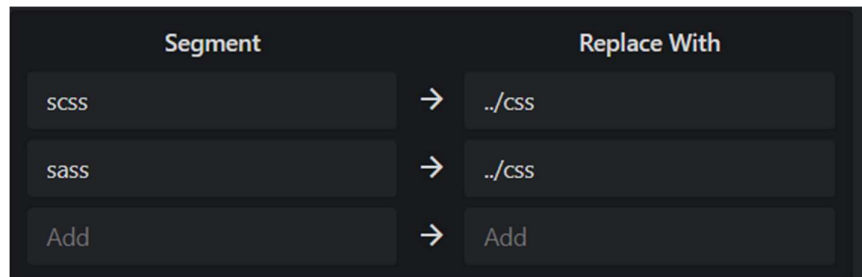


Рисунок 3.6 – Налаштування, вказівка шляху

Розроблений код мовою SCSS для одного із блоків програми:

```

.zakaz_block{
padding: 15px 20px;
background: #F4F4F4;
border-radius: 3px;
display: flex;
flex-direction: row;
.zakaz_block-info{
margin-top: 10px;
margin-left: 20px;
padding-left: 20px;
border-left: 1px solid #E6E6E6;
width: 216px;
.block-info_main-info{
margin-top: 30px;
margin-bottom: 40px;
.block-info_main-info_item{
font-weight: 700;

border-bottom: 1px solid #B6B6B6;
p{font-size: 14px;
line-height: 17px;
color: #B6B6B6;
margin-top: 2.5px;
margin-bottom: 10.5px;
}
}
.item_top{
margin-top: 10px;
}
}
}
}
}

```

Та скомпільованийий в мову CSS:

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

```

.zakaz_block {
padding: 15px 20px;
background: #F4F4F4;
border-radius: 3px;
display: -webkit-box;
display: -ms-flexbox;
display: flex;
-webkit-box-orient: horizontal;
-webkit-box-direction: normal;
-ms-flex-direction: row;
flex-direction: row;
}
.zakaz_block .zakaz_block-info {
margin-top: 10px;
margin-left: 20px;
padding-left: 20px;
border-left: 1px solid #E6E6E6;
width: 216px;
}
.zakaz_block .zakaz_block-info .block-info_main-info {
margin-top: 30px;
margin-bottom: 40px;
}
.zakaz_block .zakaz_block-info .block-info_main-info .block-info_main-info_item {
font-weight: 700;
border-bottom: 1px solid #B6B6B6;
}
.zakaz_block .zakaz_block-info .block-info_main-info .block-info_main-info_item p {
font-size: 14px;
line-height: 17px;
color: #B6B6B6;
margin-top: 2.5px;
margin-bottom: 10.5px;
}

```

Як висновок бачимо, що використання препроцесорів мови таблиць CSS, є набагато кращим, компактнішим та швидшим, ніж звичайна верстка. Причиною для цього є вміння препроцесору заощаджувати місце шляхом верстання дочірніх елементів прямо всередині батьківського блоку, як приклад, та використовувати менше повторень через наслідування, імпортування файлів, використання міксінів.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

3.2.3 Використання Javascript

Без використання мови програмування Javascript майже неможливе створення повномірного веб-сайту. Завдяки йому створюється динамічний та інтерактивний веб-контент, а саме програми та браузері. Саме використання цієї мови програмування забезпечує сайти усім функціоналом, анімаціями, математичні обрахування в реальному часі та інше. Детально було розглянуто в розділі вище.

Для забезпечення адміністративної панелі вкладками на сторінці профілю, було розроблено алгоритм перемикання.

```
function changeTabs(evt, tabName) {
    var i, mainSection, tabButtons;

    mainSection = document.getElementsByClassName("main_section");
    for (i = 0; i < mainSection.length; i++) {
        mainSection[i].style.display = "none";
    }

    tabButtons = document.getElementsByClassName("slidebar_buttons-link");
    for (i = 0; i < tabButtons.length; i++) {
        tabButtons[i].className = tabButtons[i].className.replace(" active", "");
    }

    document.getElementById(tabName).style.display = "block";
    evt.currentTarget.className += " active";
}
```

Створена функція changeTabs() передає два аргументи, завдяки першому ми знаходимо необхідну нам назву класу, та завдяки другому додаємо новий клас «active» для створення ефекту «активний» вибраної нами вкладки.

getElementsByClassName – метод який повертає масивоподібний об'єкт усіх дочірніх елементів, які відповідають усім названим іменам класів.

getElementById – метод подібний до попереднього, але вертає лиш ті елементи які відповідають виключно унікальним назвам тегів.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

ВИСНОВОК ДО ТРЕТЬОГО РОЗДІЛУ

У вище розглянутому розділі було розроблено веб-сервіс, панель адміністрування та наглядно продемонстровано усі основні моменти з розробки. Задачу розділено на два етапи - backend та frontend частини роботи.

Перший етап розробки виконаний за допомогою мови PHP, підключення до бази даних MySQL, написані головні принципи зв'язку серверу, показано головні скрипти.

В другому етапі розроблена клієнтська частина, а саме, веб-інтерфейс, функціонал сервісу.

Наведено усі налаштування допоміжних програм за використанням яких велася розробка. Детально розібрано усі алгоритми рішень, методи та функції кодової частини програми.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

РОЗДІЛ 4. ТЕСТУВАННЯ ВИКОНАНОЇ ПРОГРАМИ. ДЕМОНСТРАЦІЯ РОБОЧИХ ПРОЦЕСІВ

4.1 Інтерфейс та навігація. Авторизація та Реєстрація

Сервіс представляє адміністративну панель, яка реалізована на локальному сервері. Реалізація сайту розроблялася з принципом використанням UI/UX – дизайну. Першим кроком на сайті нас зустрічає вікно авторизації, ціль якої введення даних користувача (адміністратора магазину).

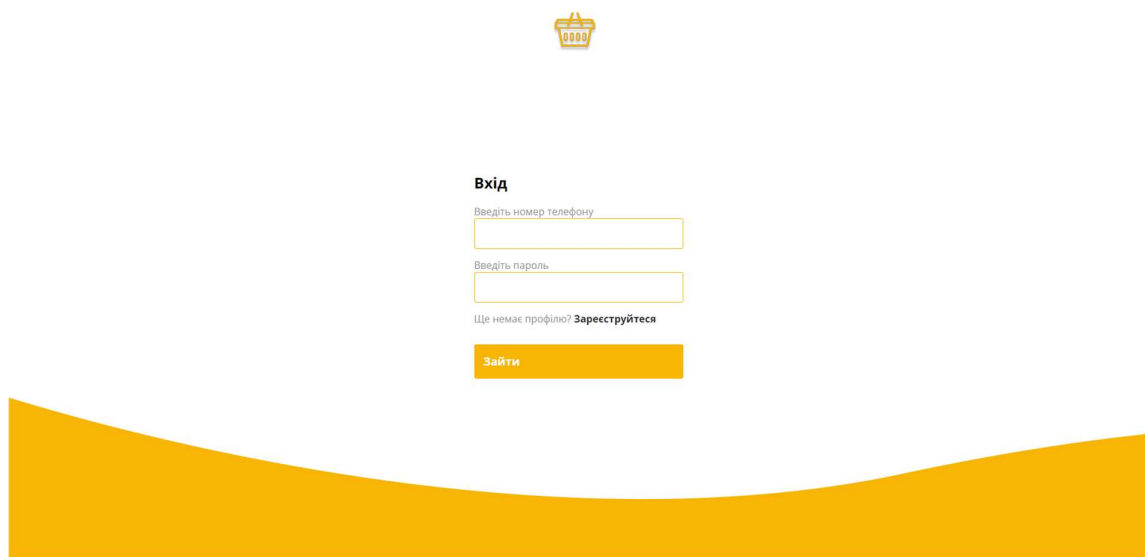


Рисунок 4.1 – Вікно авторизації

На вікні вище, ми бачимо форму заповнення даних, які надсилаються на сервер та перевіряються за допомогою перевірок чи правильними вони є, та чи є користувач з такими даними в базі.

У випадку, якщо користувач в перший раз потрапляє на наш сервіс, необхідно натиснути на кнопку «Зареєструйтеся», яке виконано як перехід на інше посилання. Вслід за цим користувач опиняється на сторінці реєстрації.

Реєстрація

Назва компанії

Номер телефону

Пароль

Підтвердження паролю

Вже є профіль? [Війти](#)

Зареєструватись

Я приймаю умови користування цієї оферти

Рисунок 4.2 – Вікно реєстрації

На сторінці реєстрації ми бачимо форму для якої необхідно заповнити поле «Номер телефону», «Пароль», «Підтвердження паролю». Заповнення поля «Назва компанії» не є необхідним. Також можемо побачити кнопку «Ввійти» це на випадок якщо користувач раптом згадав що в нього є акаунт, після натискання він буде перенаправлений на форму авторизації.

Галочка «Я приймаю умови користування цієї оферти» є необхідна, і тільки після прочитання її та натискання на неї, реєстрація буде продовжена.

Після завершення успішної реєстрації, нас знову зустрічатиме вікно авторизації, який заповнюємо вже новоствореними даними.

Під формою ми бачимо повідомлення про те, що реєстрація пройшла успішно.



Вхід

Введіть номер телефону

Введіть пароль

Ще немає профілю? [Зареєструйтеся](#)

Зайти

Регістрація виконана!

Рисунок 4.3 – Успішне виконання реєстрації



Вхід

Введіть номер телефону

Введіть пароль

Ще немає профілю? [Зареєструйтеся](#)

Зайти

Невірний логін або пароль

Рисунок 4.4 – Повідомлення про невірно введені дані

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

У випадку неправильно введені одного з полів «Пароллю» або «Номеру телефону», виводиться повідомлення про похибку.

У разі правильного введення усіх даних, вони звіряються з тими що є у нашій базі даних і після цього ми потрапляємо в особистий кабінет адміністратора.

Рисунок 4.5 – Головна сторінка панелі адміністратора

На першому екрані можемо побачити головну панель навігації (navigation). На панелі з самого верху бачимо логотип сайту, нижче – назву компанії та номер телефону, введеного при реєстрації. Головним навігатором панелі є чотири вкладки, які оформлені анімаційно (як і всі кнопки на сайті) та мають статус «активно» при натисканні на одну із них.

Чотири навігаційні кнопки (Tab):

1. Додати товар. В цьому вікні адміністратор спроможний додавати новий товар до каталогу товарів.
2. Список товарів. Вікно призначене для перегляду всіх раніше доданих товарів та їхнього редагування або видаленню.
3. Закази. Сторінка в якій ми переглядаємо усі заклази замовлені клієнтами.

Зм.	Арк.	№ докум.	Підпис	Дата

4. Додати магазин. Вікно в якому адміністратор може додати будь-який свій магазин.

Самою останньою на панелі являється кнопка виходу (exit). При натисканні на неї, користувач заново виходить з сесії свого акаунту, та буде направлений на вікно авторизації.

The screenshot shows a web form for adding a product. It is divided into three main sections:

- Головна інформація (Main information):** Contains a large text area for an image with a 'Вибрати файли' (Select files) button and 'Файл не вибран' (File not selected) text. Below are input fields for 'Назва' (Name) and 'Ціна' (Price), and a list of units with checkboxes: 'Кілограм' (Kilogram), 'Грам' (Gram), 'Упаковка' (Packaging), and 'Штука' (Piece).
- Харчова цінність (Nutritional value):** Contains four input fields, each labeled 'Калорійність' (Caloric value).
- Загальна інформація (General information):** Contains three input fields labeled 'Країна виробник' (Country of origin), 'Торгова марка' (Trade mark), and 'Час придатності' (Shelf life). A yellow 'Додати товар' (Add product) button is located at the bottom right of this section.

Рисунок 4.6 – Вигляд форми для заповнення інформації продуктів

Для заповнення форми необхідними полями для заповнення є наступні:

1. Назва продукту
2. Його ціна
3. Вибір із варіантів подачі товару (Кілограм, Грам, Упаковка, Штука)

Після заповнення всієї необхідної інформації натискаємо кнопку «Додати товар», яка є інтерактивною з використанням псевдокласу `::hover`, внаслідок чого має змінювати свій колір на інший при наведенні.

4.2 Огляд списку товарів

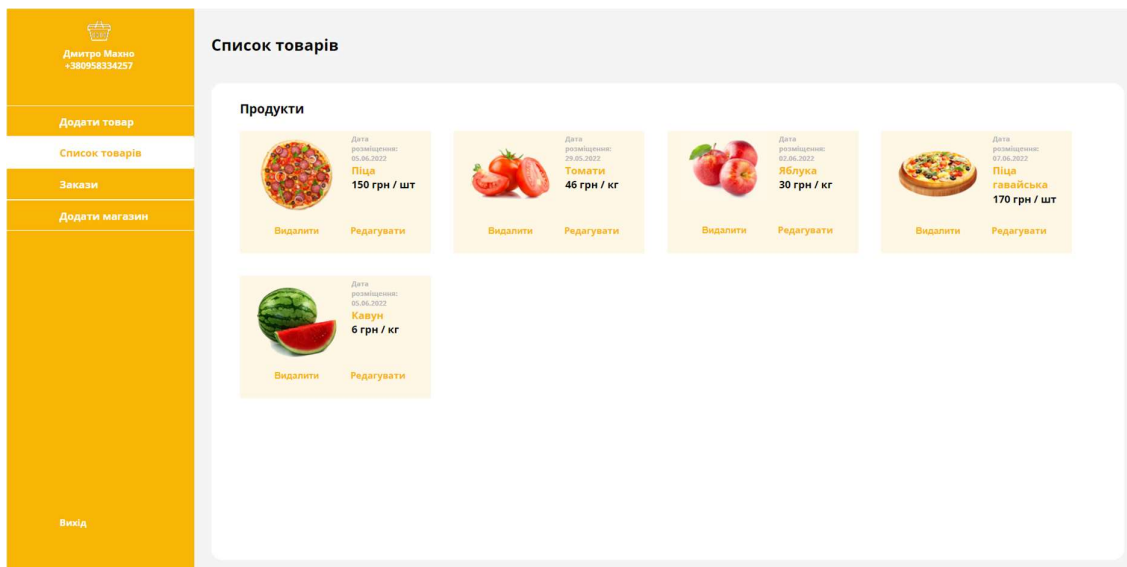


Рисунок 4.7 – Список доданих товарів

У другому навігаційному вікні бачимо список всіх раніше доданих товарів, в кожному віконці знаходиться головна інформація щодо товару: ціна, дата розміщення в базу, назва. До кожного товару є можливість редагувати його інформацію, або ж зовсім видалити за бажанням.

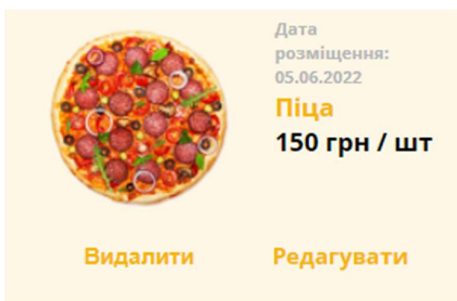


Рисунок 4.8 – Віконце товару

При натиску на кнопку «редагувати», сервіс повертає нас до вікна з формою заповнення товари, в якому змінюємо важливу нам інформацію. При натисненні кнопки видалити, вибране віконце зникає.

Зм.	Арк.	№ докум.	Підпис	Дата

4.3 Вікно замовлень

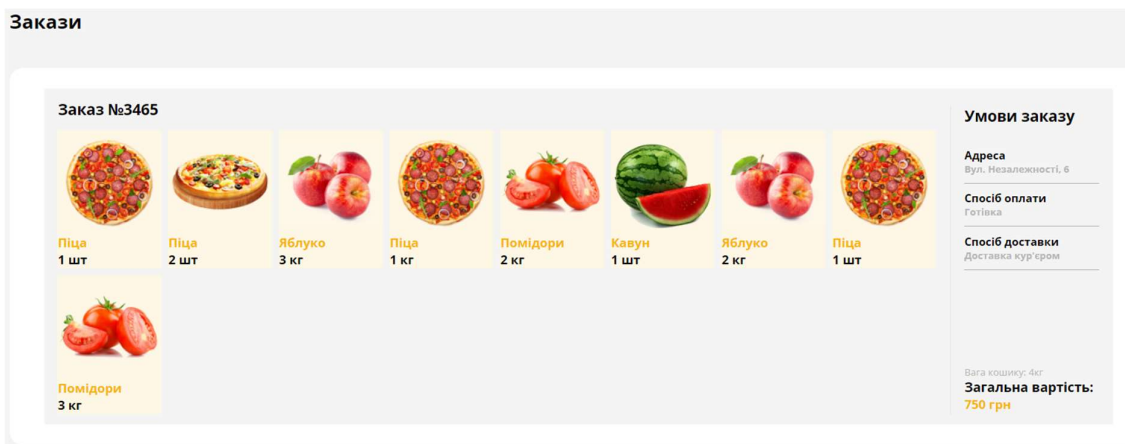


Рисунок 4.9 – Вікно замовлень

В розділі «Замовлення» бачимо список всіх замовлень від клієнтів, які нам поступають. Віконця товару мають дуже щільну інформацію, задля забезпечення вільного місця. З правої сторони маємо інформаційну панель з усіма даними замовлення, які були вказані з мобільного додатку користувача, приклад наведено нижче. За допомогою такої панелі адміністратор може слідкувати за всією інформацією в замовленнях, та бути впевненим в їхньому завершенні.

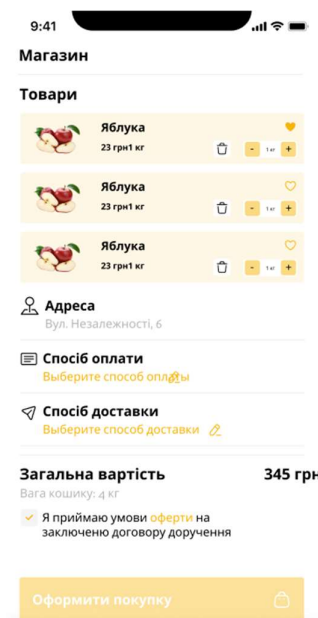


Рисунок 4.10 – Приклад мобільного додатку користувача

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

На прикладі мобільного додатку, бачимо що клієнт обирає товар, зручні йому умови доставки та за натиском кнопки підтверджує свій заказ, який згодом адміністратор бачить у своїй панелі заказів. На серверному рівні це відбувається так:

1. Клієнт вибирає потрібний йому заказ, натискає кнопку «Оформити заказ», потім надсилається HTTP-запит на сервер
2. Підтвердження від магазину щодо заказу, запит повертається знову до клієнта.
3. Клієнт отримує підтвердження, та очікує на прибуття продуктових товарів.

4.4 Вікно додавання магазину

Додати магазин

Додавання магазину

Назва компанії

Номер телефону

Додати документи +

Додати

Я приймаю умови користування цієї оферти

Рисунок 4.11 – Приклад форми додавання магазину

При необхідності користувач, якщо має два або більше своїх магазинів з різними продуктами, він може додати їх, та керувати цими магазинами.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

Форма має майже нічим не відрізняється від авторизації, за умови що тут додано кнопку «Додати документи». Кнопка має анімаційний вид та псевдоклас :hover. За допомогою псевдоелементу :after додано «+».

Після заповнення всіх полів, та відправлення їх, буде додано можливість виставляти товари на продаж з іншого магазину.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

ВИСНОВОК ДО ЧЕТВЕРТОГО РОЗДІЛУ

Виходячи з результатів розгляду даного розділу, можна сказати що проект веб-сервісу доставки продуктових товарів був повністю продемонстрований, описані всі можливі функції, показаний динамічний та інтерактивний контент. Детально розглянуті можливі подальші дії користувача.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

ВИСНОВКИ

В результаті виконання даної бакалаврської роботи за завданням розробки веб-сервісу для організації доставки продуктових товарів, роботу було успішно завершено. В процесі розробки проекту було залучено велику кількість програмних компонентів, мов програмувань, середовищ, допоміжних додатків.

Створена сучасна адміністративна панель з новітнім дизайном, яка дозволяє користувачу керувати власним магазином онлайн без різних проблем. Є можливості щодо додавання нових продуктів, керування базою продуктів, видаляти старі та керувати заказами. Завдяки сервісу, організація (магазин) може створити власний бізнес в інтернеті, просуваючи та продаючи продуктові товари онлайн.

Система має власну базу даних, яка організована підключенням через серверну основу, використано мову програмування Javascript для встановлення всього функціоналу та динамізації. Внаслідок чого були отримані нові знання щодо програмування та поліпшені старі навички.

В майбутньому такий веб-сервіс можна оновлювати та додавати новітні функції, так як ще є багато корисних потребностей в цьому напрямку. Проект побудован в такому стилі, щоб дійсно стати одним з конкурентоздібних сервісів та стати спроможним в реальному ринку.

Ознайомитись детальніше з програмною реалізацією проекту можна за наступним посиланням: <https://github.com/killreal321/bachelor-s-work>.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архитектура веба: основы для начинающих разработчиков. *Tproger*. URL: <https://tproger.ru/translations/web-architecture-101/>.
2. Клиент-серверна архітектура [Електронний ресурс] – Режим доступу до ресурсу: <https://doka.guide/js/web-app-works/>.
3. HostCMS. *HostCMS*. [Електронний ресурс] – Режим доступу до ресурсу: <http://demo.hostcms.ru/>.
4. DemoSimplaCms.[Електронний ресурс] – Режим доступу до ресурсу:<http://demo.simplacms.ru/>.
5. Драч В. Анализ популярных реляционных систем управления базами данных (2022 г.). *Сайт Владимира Драча*. URL: <https://drach.pro/blog/hi-tech/item/196-popular-relational-dbms-2022>.
6. Интерактивные онлайн-курсы HTML Academy. *HTML Academy*. [Електронний ресурс] – Режим доступу до ресурсу: <https://htmlacademy.ru/tutorial/php/mysql>.
7. Переваги та недоліки JavaScript. *hackit-ukraine*. [Електронний ресурс] – Режим доступу до ресурсу: <https://hackit-ukraine.com/627-the-advantages-and-disadvantages-of-javascript>.
8. Goodsn. Самые популярные языки программирования бэкенда: для чего они подходят лучше всего и какие компании их используют. *Хабр*. [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/skillbox/blog/534684/>.
9. Megida D. What is JavaScript? A Definition of the JS Programming Language. *freeCodeCamp.org*. [Електронний ресурс] – Режим доступу до

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ресурсу: <https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/>.

10. phpMyAdmin. *phpMyAdmin*. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.phpmyadmin.net>.

11. PHP | unset() Function - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/php-unset-function/>.

12. RushenRoshen. Рейтинг языков программирования 2021: доля Python падает, а TypeScript обошел C++, в лидерах JavaScript, Java, C#. *Хабр*. URL: <https://habr.com/ru/post/543346/>.

13. WebForMySelf - Как создать сайт с нуля, самому и бесплатно. *WebForMySelf - публикация тематических материалов по веб-разработке и сайтостроению*. [Електронний ресурс] – Режим доступу до ресурсу: <https://webformyself.com/>.

14. What is MySQL: MySQL Explained For Beginners. *Hostinger Tutorials*. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hostinger.com/tutorials/what-is-mysql>.

15. What is UI design? What is UX design? UI vs UX: What's the difference [Електронний ресурс] – Режим доступу до ресурсу: <https://uxplanet.org/what-is-ui-vs-ux-design-and-the-difference-d9113f6612de>.

16. HTML и HTML5. Описание тегов по основным разделам. *HTML5BOOK.RU - HTML, CSS, JavaScript и jQuery*. [Електронний ресурс] – Режим доступу до ресурсу: <https://html5book.ru/html-html5/>

17. Гущин Д. Стандарты, стилі та правила оформлення коду PHP. *КовельPost: Актуальні новини міста Ковель*. URL: <https://kovelpost.com/blogs/9229>.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

18. UX/UI design principles to help create an engaging experience. *Fresh Consulting*. URL: <https://www.freshconsulting.com/insights/blog/7-uiux-principles-fresh-web-experiences/>.
19. Реляционные СУБД – сравнение MySQL и SQL сервер. *Руководства Hostinger*. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.hostinger.com.ua/rukovodstva/reljacionnye-subd-sravnienie-mysql-i-sql-server/>.
20. Выбор CMS для интернет-магазина: сравниваем 4 движка. *Продвижение сайтов от IPS.RU*. [Электронный ресурс] – Режим доступа до ресурсу: <https://1ps.ru/blog/sites/vyibiraem-cms-dlya-internet-magazina-1/#simpla>.
21. Основы асинхронного програмування на JavaScript. *Codeguida*. URL: <https://codeguida.com/post/445>.
22. Simplilearn. Header in PHP: The Ultimate Guide to Header Function. *Simplilearn.com*. URL: <https://www.simplilearn.com/tutorials/php-tutorial/header-in-php#>.
23. mysqli_fetch_assoc. *Mathematical and Computer Sciences- Heriot-Watt University*. [Электронный ресурс] – Режим доступа до ресурсу: [https://www.macs.hw.ac.uk/~hwloidl/docs/PHP/function\(mysqli-fetch-assoc.html#](https://www.macs.hw.ac.uk/~hwloidl/docs/PHP/function(mysqli-fetch-assoc.html#)
24. Why Use PHP in 2022? Main Advantages and Disadvantages. *Web and Mobile App Development Company | Light IT*. URL: <https://light-it.net/blog/why-use-php-main-advantages-and-disadvantages/>
25. Какая разница между SCSS и Sass? - программирование. *Ответы на вопросы пользователей по программированию на русском языке stackoverflow*. URL: <https://ask-dev.ru/info/10892/whats-the-difference-between-scss-and-sass>

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

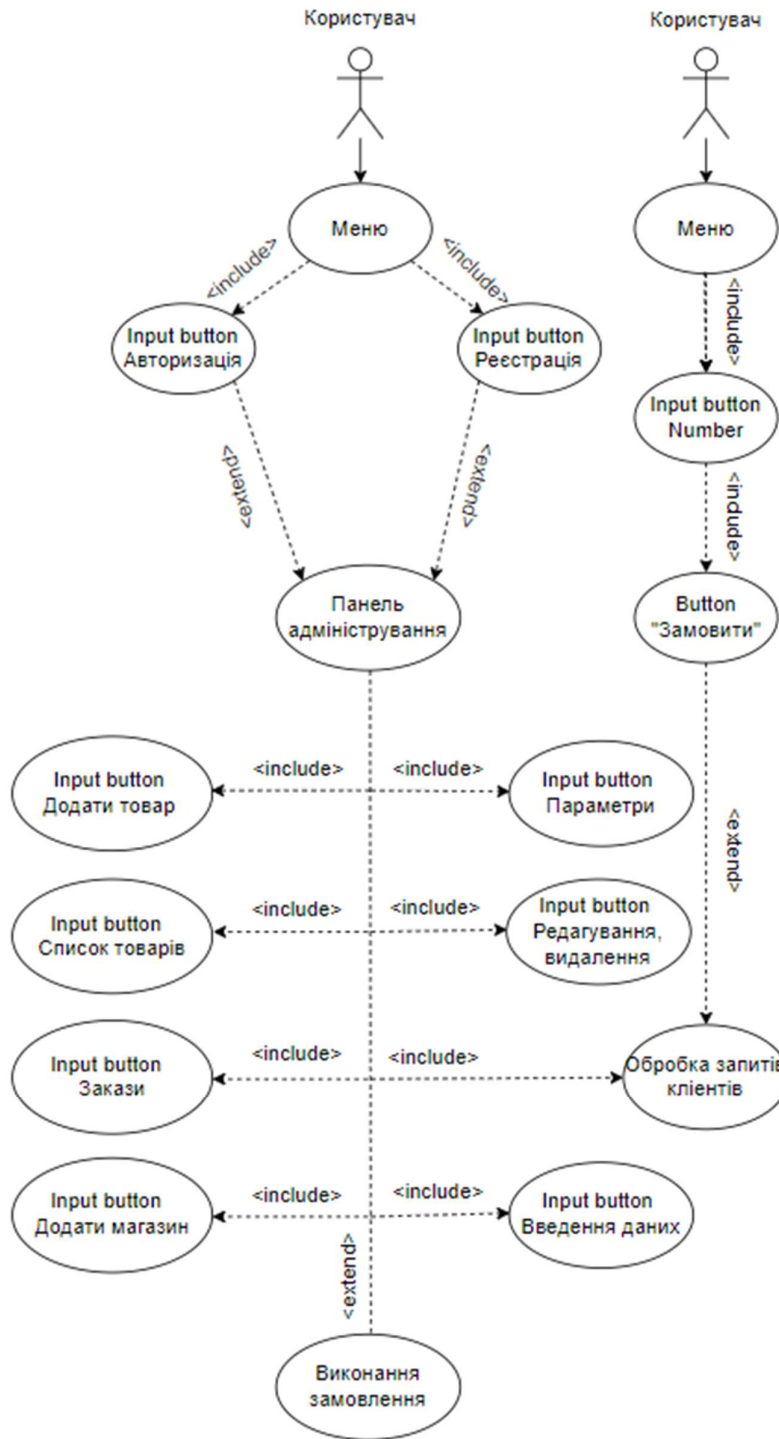
ДОДАТОК 1

Веб-сервіс для організації процесу доставки продуктових
товарів

Структурна схема системи
ІАЛЦ.467200.004 Д1

Аркушів 1

Київ 2022 р



ІАЛЦ.467200.004 Д1				
	№ докум.	Підпис	Дата	
Розробив	Сушільников К.Д.			
Перевірив	Кочура Ю.П.			
Н. Контр.	Сімоненко В. П.			
Затвердив				
Веб-сервіс для організації процесу доставки продуктивних товарів			Літ.	Аркуш
Структурна схема системи			1	1
НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-83				

ДОДАТОК 2

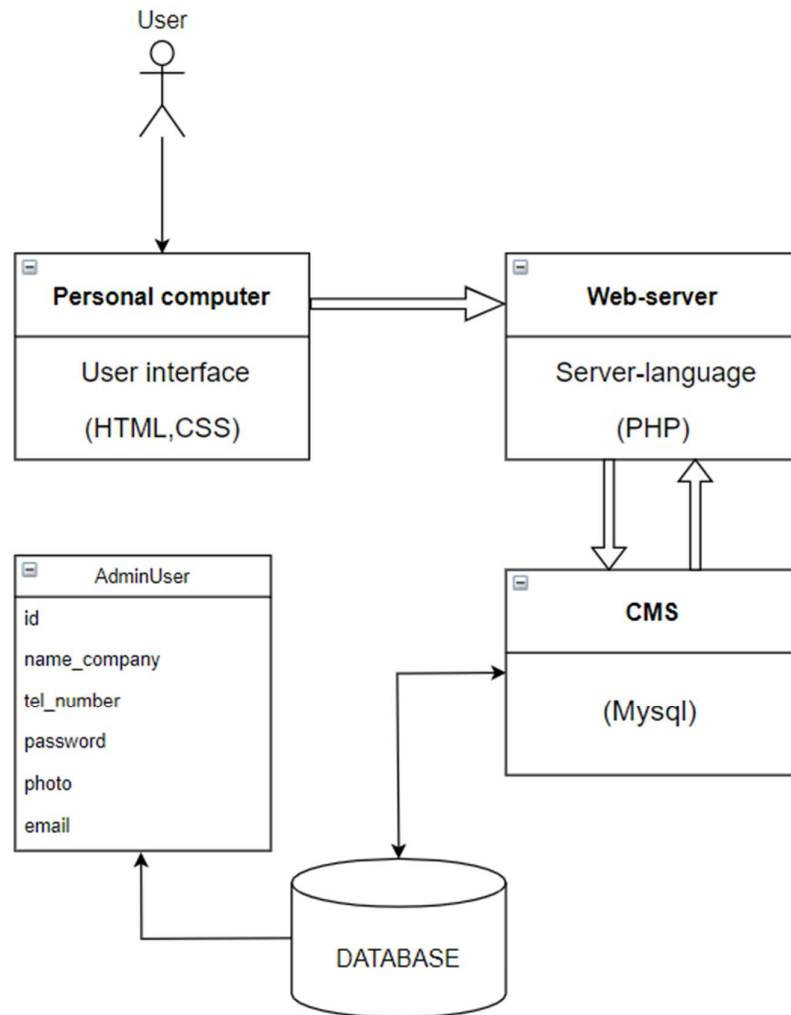
Веб-сервіс для організації процесу доставки продуктових
товарів

Функціональна схема (діаграма класів)

ІАЛЦ.467200.005 Д2

Аркушів 1

Київ 2022 р



					ІАЛЦ.467200.005 Д2		
		№ докум.	Підпис	Дата			
Розробив	Сушильников К.Д.				Літ.	Аркуш	Аркушів
Перевірив	Кочура Ю.П.					1	1
Н. Контр.	Сімоненко В. П.				НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-83		
Затвердив							
Веб-сервіс для організації процесу доставки продуктивних товарів							
Функціональна схема (діаграма класів)							

ДОДАТОК 3

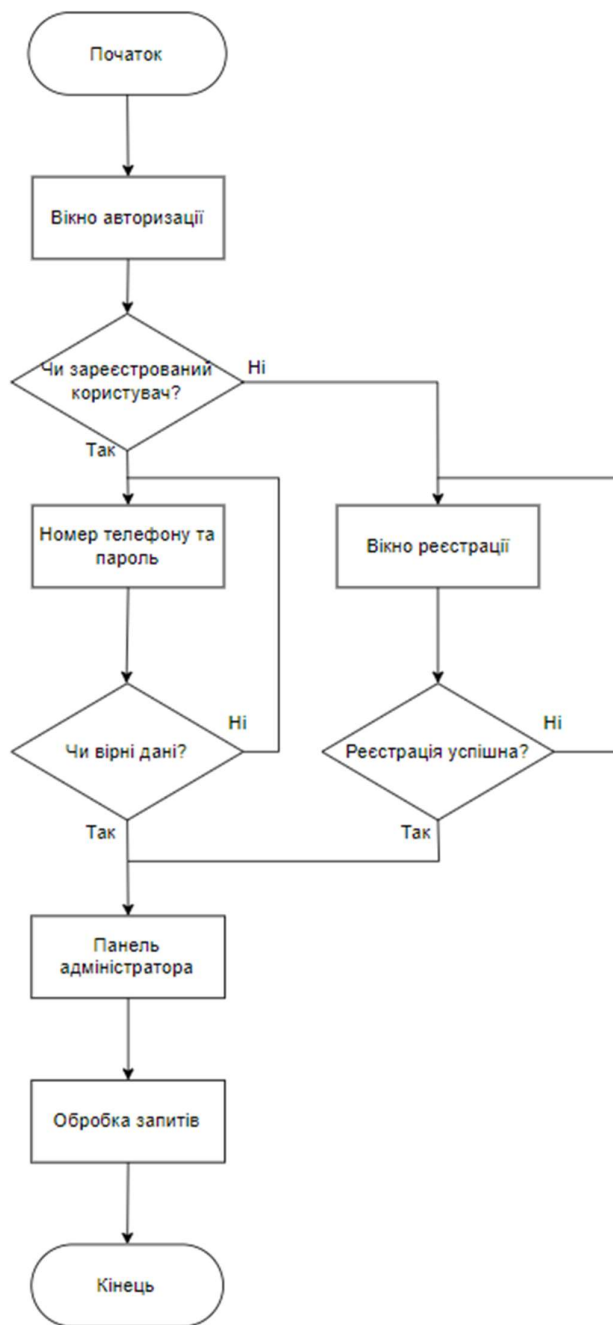
Веб-сервіс для організації процесу доставки продуктивних
товарів

Алгоритм дій

ІАЛЦ.467200.006 ДЗ

Аркушів 1

Київ 2022 р



					ІАЛЦ.467200.006 ДЗ		
		№ докум.	Підпис	Дата			
Розробив	Сушільников К.Д.				Літ.	Аркуш	Аркушів
Перевірив	Кочура Ю.П.					1	1
Н. Контр.	Сімоненко В. П.				НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-83		
Затвердив							

Веб-сервіс для організації процесу доставки продуктивних товарів
Алгоритм дій