

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ОНЛАЙН ЗАМОВЛЕНЬ В
ХАРЧОВІЙ КРАМНИЦІ. SOFTWARE IMPLEMENTATION OF AN ONLINE
ORDER SYSTEM IN A GROCERY STORE

Виконав: студент(ка) 4 курсу, групи КНТ-137
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Пилипчук.І.О.

(прізвище та ініціали)

Керівник Зеленьова.І.Я.

(прізвище та ініціали)

Рецензент Терещенко.Е.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА(КИ)

Пилипчука Ігора Олександровича
 (прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Програмна реалізація системи онлайн замовлень в харчовій крамниці

Software Implementation of an Online Order System in a Grocery Store,

керівник проекту (роботи) Зеленьова Ірина Яківна, к.н.т., доцент
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проекту (роботи) 17 травня 2021 року

3. Вихідні дані до проекту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми. 3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація, тестування та експериментальне дослідження програми. 5. Технічне завдання. 6. Текст програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	Прийняв виконане завдання
Основна частина	Зеленьова І.Я., доцент		
Нормоконтроль	Андрєєв М.О., асистент		

7. Дата видачі завдання 15.03. 2021 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка архітектури програми.	4 тиждень	Розділ 2
4	Розробка програми.	5–6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми.	7 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8 тиждень	Додатки
7	Захист роботи.	9 тиждень	

Студент (ка) _____ **І.О.Пилипчук**
(підпис) (прізвище та ініціали)Керівник проекту (роботи) _____ **І.Я.Зеленьова**
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
83 с., 5 табл., 39 рис., 2 дод., 25 джерел.

ОНЛАЙН ЗАМОВЛЕННЯ, ХАРЧОВА КРАМНИЦЯ, ІНТЕРНЕТ-
МАГАЗИН, ДОСТАВКА ЗАМОВЛЕНЬ.

Об'єктом дослідження є інформаційне забезпечення вебсайту систем онлайн замовлень.

Предметом дослідження є об'єктно-орієнтований аналіз та проектування онлайн систем замовлень в харчовій крамниці.

Метою даної роботи є розробка і реалізація системи онлайн замовлень в харчовій крамниці (інтернет-магазину).

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, технологія розробки ASP.NET, мова програмування C#, СУБД MS SQL Server 2018, персональний комп'ютер з процесором Intel Core 2 Duo під управлінням операційної системи Microsoft Windows 7 та вище.

Результати. Створено програмний вебзастосунок для оформлення онлайн-замовлень та їх доставки у харчовій крамниці.

Висновки. Розроблено веб-орієнтовану програмну систему для онлайн замовлень харчової крамниці, яка автоматизує та прискорює процес замовлень та доставки товарів онлайн харчової крамниці.

ABSTRACT

Explanatory note to the final qualifying work of the bachelor: 83 pages, 39 figures, 5 tables, 2 appendixes, 25 sources.

ONLINE ORDERING, FOOD STORE, ONLINE STORE, ORDER DELIVERY.

The object of research is information support for the website of online order systems.

The subject of the research is Object-Oriented Analysis and design of online order systems in a food store

The purpose of this work is to develop and implement an online order system in a food store (online store).

Materials, methods and technical means: structural and Object-Oriented Programming, Development Technology ASP.NET, C # programming language, MS SQL Server 2018 DBMS, personal computer with Intel Core 2 Duo processor running Microsoft Windows 7 and higher operating system.

Results. A software web application has been created for placing online orders and delivering them to a food store.

Conclusions. A web-based software system for online food store orders has been developed, which automates and speeds up the process of ordering and delivering goods to an online food store.

ЗМІСТ

Перелік умовних позначень, символів, скорочень і термінів	7
Вступ	8
1 ОГЛЯД АНАЛОГІВ І АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ СИСТЕМИ ДЛЯ ОНЛАЙН ЗАМОВЛЕНЬ ХАРЧОВОЇ КРАМНИЦІ.....	10
1.1 Аналогічні сучасні системи	10
1.2 Основні теоретичні положення розробки систем онлайн-замовлень	13
1.3 Програмні засоби для розробки	21
1.4 Постановка задачі розробки і дослідження системи для онлайн замовлень в харчовій крамниці.....	34
2 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ ДЛЯ ОНЛАЙН ЗАМОВЛЕНЬ В ХАРЧОВІЙ КРАМНИЦІ.....	38
2.1 Загальна структура програми.....	38
2.2 Модуль реєстрації та авторизації.....	41
2.3 Модуль онлайн-замовлень	45
2.4 Модуль адміністрування веб-програми.....	47
3 ДОСЛІДЖЕННЯ СИСТЕМИ І РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ ...	48
3.1 Розробка тестових сценаріїв для дослідження вебзастосунку	48
3.2 Дослідження та тестування вебпрограми.....	54
3.3 Рекомендації щодо вдосконалення вебпрограми.....	58
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	63
Додаток А Текст програми.....	66
Додаток Б Слайди презентації	76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
CDMA	– Code Division Multiple Access;
IDE	– Integrated Development Environment,
PGA	– Programmable-Gain Amplifier;
PsoC	– Programmable System-on-Chip;
RTL	– Register Transfer Level;
SAR	– Successive Approximation Register;
SoC	– System-on-Chip;
TRM	– Technical Reference Model;
UDB	– Universal Digital Block;
АЦП	– аналого-цифровий перетворювач;
BC	– вбудована система;
BIC	– велика інтегральна схема;
МК	– мікроконтролер;
ПЗ	– програмне забезпечення;
ПСнК	– програмована система на кристалі;
ЦАП	– цифро-аналоговий перетворювач;
⊗	– операція нечіткого логічного перетинання.

ВСТУП

Електронний магазин – це прикладна система, побудована з використанням технології системи електронної торгівлі. Подібно звичайному магазину, електронний магазин реалізує наступні основні функції: представлення товарів (послуг) покупцеві, обробку замовлень, продаж і доставку товарів.

Інтернет-магазин об'єднує елементи прямого маркетингу з образом відвідування традиційного магазину. Відмінною рисою інтернет-магазинів в порівнянні зі звичайною формою торгівлі є те, що інтерактивний магазин може запропонувати значно більшу кількість товарів і послуг, і забезпечити споживачів значно більшим обсягом інформації, необхідним для прийняття рішення про покупку. Крім того, за рахунок використання комп'ютерних технологій можлива персоналізація підходу до кожного з клієнтів, виходячи з історії його відвідувань магазину і зроблених раніше покупок [1-4].

Основні проблеми реалізації інтернет-магазину лежать на стику технологій Інтернету і традиційної комерційної діяльності. У звичайній торгівлі покупець звик до того, що є можливість оцінити товар візуально, визначити його якість і характеристики. В електронній торгівлі він такої можливості позбавлений. Найчастіше візуальної інформації досить, але тут вступають в дію емоційні та психологічні чинники. Часто існують проблеми з доставкою товарів, особливо якщо їх ціна невелика.

За останніми даними аудиторія в Інтернеті стрімко зростає, а продажі через Інтернет у великих містах досягають до 25%, при цьому фахівці підкреслюють тенденцію до зростання продажів саме через Інтернет. Щорічно кількість інтернет-магазинів збільшується, так як це дійсно прибутково і зручно для покупця, не кажучи про економію бюджету і часу. Інтернет-магазин працює цілодобово і може продавати певні товари в автоматичному режимі без участі продавця. До переваг так само можна віднести те, що не треба закуповувати товар заздалегідь, а це суттєва економія, на складських приміщеннях. Потрібно

лише домовитися з постачальниками, і в потрібний момент, просто викупити товар, який у вас замовлять. У порівнянні зі звичайним магазином, територія продажів якого обмежується населенням міста або району, територія охоплення інтернет-магазину збільшується на всю країну або і на інші країни, адже товар можна доставляти не тільки кур'єрською службою, а й поштою [5-6].

У роботі використані методи об'єктно-орієнтованого аналізу та проектування .

1 ОГЛЯД АНАЛОГІВ І АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ СИСТЕМИ ДЛЯ ОНЛАЙН ЗАМОВЛЕНЬ ХАРЧОВОЇ КРАМНИЦІ

1.1 Аналогічні сучасні системи онлайн-замовлень

"Система онлайн замовлень" компанії Роберт БОШ Україна. "Система онлайн замовлень" призначена для автоматизації процесів здійснення замовлень товарів, їх пересилання і відстеження.

Система складається з наступних модулів: Профіль, Баланс, Каталог, Замовлення, Кошик, Пошта.

Модуль Профіль надає користувачам наступну інформацію:

- інформація про компанію;
- адресна інформація;
- інформація про користувачів системи.

Модуль Баланс надає користувачеві системи наступну інформацію:

- загальна сума заборгованості за замовлену продукцію;
- прострочена заборгованість.

Модуль Каталог надає інформацію про номенклатуру товарів, які користувач може замовити у компанії і дозволяє вибрати необхідні товари і додавати їх в «кошик» замовлення.

Модуль Кошик – модуль, який дозволяє додавати товари в "кошик замовлення", переглядати замовлені позиції, коригувати поточне замовлення, переглядати зміну вартості замовлення при різних знижках і безпосередньо створювати замовлення і відсилати його в компанію.

Модуль Замовлення – в цьому модулі користувачеві надається повна інформація про зроблені замовлення.

Модуль Пошта – в цьому модулі користувач відправляє і приймає повідомлення компанії.

В системі доступні наступні ролі для співробітників компанії: "Адміністратор системи".

В системі доступні наступні ролі для корпоративних клієнтів: "Клієнт".

Функціональні можливості ролі "Клієнт":

- оформлення нових замовлень;
- перегляд вже сформованих замовлень;
- отримання інформації про статус замовлення-відмова або прийом до обробки;
- перевірка стану свого балансу в системі;
- зміна інформації про компанію;
- роздруківка звітів.

Рішення для управління замовленнями на базі Comindware Business Application Platform з опорою на типові рішення і з урахуванням вимог бізнесу. Система зазвичай охоплює всі етапи обробки замовлення - від укладення договору і до підписання Акту виконаних робіт, включаючи весь супутній документообіг, управління людськими ресурсами - і дозволяє клієнту відстежувати статус замовлення в особистому кабінеті. Завдяки використанню low-code платформи автоматизація бізнес-процесів обробки і виконання замовлень відбувається з мінімальними початковими витратами і в найкоротші терміни.

Унікальне рішення для бізнесу оперативно будується на основі набору типових об'єктів і функцій, необхідних для швидкого впровадження системи управління замовленнями на підприємстві, автоматизації прийому замовлень і їх обробки, і ефективного управління рахунками та іншими супутніми документами в компанії.

Бізнес-об'єкти:

- замовлення (проекти);
- майданчики (об'єкти);
- етапи, роботи;
- ресурси (виконавці).

Бізнес-процеси:

- від звернення до комерційної пропозиції;
- від договору до акту здачі робіт;

- інші бізнес-процеси.

Ключова функціональність:

- особистий кабінет клієнта;
- можливість гнучкого перепланування робіт і ресурсів;
- можливість скасування, зміни складу робіт "на льоту".

Перераховані вище типові елементи рішення для управління замовленнями можуть бути змінені "на льоту" і підлаштовані під актуальні потреби і завдання компанії, що робить процес впровадження онлайн системи управління замовленнями гладким і адаптивним.

Гнучка система управління замовленнями на базі low-code платформи Comindware є надійним помічником в питанні забезпечення якості виконання замовлень і забезпечує:

- ефективне управління часом співробітників;
- прозорість маршрутизації замовлень;
- налагоджений процес комунікації між відділами компанії;
- автоматизовану обробку рахунків та інших супутніх документів;
- управління змінами та скасуванням замовлень;
- зручність відстеження статусу замовлення клієнтом.

З кастомізованою системою управління замовленнями всі бізнес-процеси при обробці і виконанні замовлень стають узгодженими, що виключає появу помилок і значно підвищує ефективність роботи. У рішення для управління замовленнями входить також мобільна версія, доступна на смартфонах і планшетах під управлінням iOS і Android.

Система допомагає у вирішенні нетривіальної задачі оптимального розподілу завдань по замовленнях між наявними в розпорядженні співробітниками, а також забезпечує облік залученості кожного співробітника в повсякденні завдання, що не входять в процес виконання замовлень.

Управління замовленнями вимагає прозорості виконання щоденних операцій і зручності перевірки статусу замовлення для клієнтів. Система

Comindware дозволяє створити портал самообслуговування з особистими кабінетами для клієнтів, партнерів і внутрішніх користувачів.

У режимі реального часу відстежуйте замовлення і керуйте якістю послуг, що надаються.

1.2 Основні теоретичні положення розробки систем онлайн-замовлень

Електронна комерція (англ. e-commerce, e-comm,) стала невід'ємною частиною нашого суспільства. У міру інтенсивного зростання числа користувачів мережі Інтернет вона стає все більш потужним інструментом бізнесу, засобом реалізації товарів і послуг. Інтернет став основним джерелом інформації для величезного числа потенційних покупців.

Електронна комерція - сфера світової економіки, що включає в себе фінансові та торгові транзакції, які проводяться за допомогою комп'ютерних мереж, а також всі види бізнес-процесів, пов'язані з проведенням таких транзакцій.

Електронна комерція надає величезні можливості для розвитку своєї справи. Однак для того щоб використовувати її дійсно ефективно мало бути талановитим підприємцем, точно так же, як і мало вміти орієнтуватися в безмежному віртуальному просторі. Необхідно ще володіти рядом високотехнологічних інструментів для ведення електронного бізнесу. Одним з них і є інтернет-магазин.

Інтернет-магазин є одним з різновидів електронної комерції. Продаж власних товарів і послуг через Інтернет – основний спосіб комерційного використання сайтів для більшості виробничих і торгових підприємств. Компанія створює сайт і розміщує на ньому інформацію про свої товари і послуги, ціни і гарантії для покупців. Інтернет-магазини стають одним з необхідних і ефективних інструментів збільшення числа продажів

підприємства, збільшення обороту і підвищення іміджу, успішного розвитку. Існує кілька визначень» що таке інтернет-магазин".

Інтернет-магазин (англ. online shop або e-shop) - спеціалізований вид бізнесу, заснований на дистанційному обслуговуванні і дає можливість продавцеві і покупцеві здійснювати операції з купівлі-продажу того чи іншого виду товарів за допомогою мережі Інтернет.

Інтернет-магазин - це інтерактивний веб-сайт, що рекламує товар або послугу, що приймає замовлення на покупку, пропонує користувачеві вибір варіанту розрахунку, способу отримання замовлення і виписує рахунок на оплату.

Інтернет-магазин - це сукупність програм, що працюють на Web-сайті, які дозволяють покупцеві Дистанційно вибрати товар з каталогу і оформити його замовлення.

Інтернет-магазин є посередником між покупцем і складом виробника, якому достатньо мати ресурс в Інтернеті і телефон. Магазин віртуального типу не потребує представницькому офісі, складських приміщеннях, не несе витрати на утримання приміщень, власник магазину може управляти продажами навіть зі своєї квартири. Саме тому ціни в таких магазинах мінімальні. Куплені речі привозять споживачеві безпосередньо зі складу виробника.

Інтернет-магазини можна класифікувати за такими видами:

- по моделі бізнесу;
- за обсягами продажів;
- за видами продажів;
- за способами отримання доходу;
- за товарним асортиментом;
- по відношенню з постачальниками;
- за представленими товарами в каталозі.

По моделі бізнесу:

- чисто онлайновий магазин;

– поєднання офлайнного бізнесу з онлайнним (коли інтернет-магазин був створений на основі вже діючої реальної торгової структури). Цей вид інтернет-магазину характеризуються відмінними рисами класичного бізнесу: склад, куплені партії товару, співробітники, бухгалтер, юрист і т. д.;

– на аутсорсингу. При цьому виді інтернет-магазину користувач безпосередньо не займається прийомом, доставкою, зберіганням і комплектацією замовлень, а передає на аутсорс сторонньої компанії, вирішуючи лише організаційні бізнес питання. Така модель звільняє масу часу і в той же час, дозволяє отримувати досить значущий прибуток (близько 70% від фактичної виручки інтернет-магазину);

– продаж по системі дропшипінг (коли товару немає, а сайт магазину «продає» продукцію постачальників, відправляючи замовнику готові замовлення або купуючи у нього від імені свого покупця).

За обсягами продажів:

– роздрібний продаж (продажі на одного клієнта йдуть в невеликих обсягах, найчастіше по одній одиниці товару);

– оптовий продаж (продажі йдуть у великих обсягах, часто клієнт є постійним, що говорить про систематичність покупок).

За видами продажів:

– B2B (Бі-ту-бі або «бізнес для бізнесу») - продаж або надання послуг іншим комерційним підприємствам;

– B2C (Бі-ту-СІ або "бізнес для клієнта") - продаж або надання послуг кінцевому споживачеві.

За способами отримання доходу:

– продаж через сайт товарів і послуг від виробника або офіційного представника. Продаж власних товарів і послуг через Інтернет – основний спосіб комерційного використання сайтів для більшості виробничих і торгових підприємств. Компанія створює сайт і розміщує на ньому інформацію про свої товари і послуги, ціни і гарантії для покупців.

– продаж товарів і послуг за партнерською програмою (афіліатські мережі). Основна ідея партнерських програм - це участь в чужих продажах. Суть подібних програм полягає в тому, що продавець (власник «партнерки») згоден розділити частину доходу від продажу товарів або послуг з власником Того сайту (партнером, або афіліатом), який надіслав йому покупця;

– продаж інформації (контенту). Для продажу інформації вже немає необхідності у витратах на папір, тиражі, точки продажів. Будь-яка веб-сторінка може бути доступна мільйонам користувачів з усіх країн світу. Основний спосіб реалізації цього виду інтернет-магазину-платні електронні розсилки і сайти з платним доступом. Оплата за інформацію відбувається таким же чином - в он-лайн, як і при оплаті матеріальних товарів в стандартному інтернет-магазині.

Види інтернет-магазинів по товарному асортименту: книги, аудіо та відеокасети, CD, DVD, комп'ютерна та побутова техніка, Мобільні телефони, меблі, одяг, взуття, годинники та інші товари.

По відношенню з постачальниками:

- мають власний склад (наявність реальних товарних запасів);
- працюють за договорами з постачальниками (відсутність значних власних запасів).

Види інтернет-магазинів по представленим товарам в каталозі:

- інтернет-вітрини;
- інтернет-магазини;
- онлайн-аукціони.

Кожен з видів інтернет-магазину включає в себе кілька груп, досить чітко визначають сенс існування магазину і його аудиторію.

Інтернет-вітрина-скоріше це рекламний сервер. На вітрині викладають інформацію про товари, яку постійно оновлюють. Витрати на її створення та адміністрування можуть бути досить низькими, а практична користь такої вітрини очевидна. Але це ще не торгівля. Потенційний покупець, відвідавши вітрину, повинен зателефонувати на фірму, оплатити товар, домовитися про

доставку. Тому інтернет-вітрина виправдана в тих випадках, коли покупця треба познайомити зі складною продукцією, на вивчення якої в торговому залі у нього піде занадто багато часу.

Інтернет-вітрина може бути розміщена де завгодно-на власному сервері, на сервері провайдера, на сервері, що надає безкоштовні сторінки. Для роботи з вітриною досить мати підключення через телефонну лінію і мінімум навичок роботи з HTML.

Каталог інтернет-магазину буде містити багаторівневе логічне дерево розділів і підрозділів, всередині яких будуть знаходитися списки з картками товарів. Він повинен бути добре структурованим і дуже логічним, щоб в різноманітності товарів відвідувачі знайшли те, що їм дійсно потрібно. Кінцевою точкою переміщення по каталогу повинна стати картка товару, що містить його опис, ознака наявності у продажу і ціну. Список в каталозі сортується за ознакою наявності товару на складі. Актуальні товари знаходяться на початку списку, зняті з продажу-в кінці.

Онлайн-аукціон, або Інтернет-аукціон - це аукціон, що проводиться за допомогою мережі Інтернет. Він є видом інтернет-магазину, так як має в своєму функціоналі прийом онлайн-платежів. Момент закінчення Інтернет-аукціону заздалегідь призначається самим продавцем при постановці товару на торги. Після закінчення Інтернет-аукціону покупець повинен перевести гроші продавцеві за безготівковим розрахунком (рідше — готівкою, наприклад — при отриманні товару особисто), а продавець зобов'язаний вислати товар покупцеві поштою, нерідко в будь-яку точку країни проведення або всього світу. Межі можливої пересилання товару вказуються самим продавцем заздалегідь [5-6].

Головна особливість Інтернет-магазину - велика частина взаємодії продавець-покупець відбувається в он-лайн. Багато етапів покупки і продажу досить успішно проходять дистанційно. Це є великою перевагою. Але в той же час, саме в цьому і полягає одна з труднощів, яку треба подолати будь-якому інтернет-магазину - вселити довіру відвідувачу і розташувати його до здійснення покупки.

Відвідувач заходить на сайт і набирає кошик товарів, після чого відбувається оформлення замовлення. Якщо Відвідувач вже купував товари на цьому сайті, досить ввести логін, пароль і уточнити адресу доставки. Якщо не купував - необхідно буде додатково ввести контактні дані. Потім вибирається спосіб доставки і формується кінцева вартість Замовлення (товари + доставка).

Наступним етапом є вибір форми оплати. Як правило, відбувається перехід на сайт платіжної системи, де користувач авторизується і оплачує покупку.

Після оплати відвідувач повертається на сайт і отримує підтвердження - пройшла оплата чи ні. Якщо ні - зазвичай пропонується повторити спробу або зв'язатися з менеджером інтернет-магазину для отримання допомоги.

Залежно від обраного способу доставки в термін, вказаний на сайті або повідомлений менеджером, приходить замовлення - на пошту або за адресою. У разі доставки кур'єром іноді можна вказати зручні дні і час доставки.

Способи оплати покупок в інтернет-магазині:

- банківська карта — безготівковий вид розрахунку, який часто піддається нарікань в плані безпеки;
- банківський переказ-оплата за замовлення проводиться банківським платіжним дорученням на розрахунковий рахунок магазину;
- електронні гроші-безготівковий вид розрахунку;
- термінали моментальної оплати-оплата проводиться у вуличних платіжних терміналах;
- SMS-платежі. Інтернет-магазин формує запит коштів, передає його в одну зі спеціалізованих систем електронних платежів і відправляє туди користувача. Користувач вже у взаємодії з платіжною системою відправляє SMS із зазначеними системою реквізитами і підтверджує платіж. Платіжна система після підтвердження відправляє магазину повідомлення про успішність переказу або навпаки;
- накладений платіж - товар оплачується на пошті при отриманні;

- готівковий розрахунок-товар оплачується покупцем кур'єру готівкою при отриманні товару;

- електронні каси-вид розрахунку, який об'єднує практично всі перераховані вище способи оплати. При виборі такого способу оплати користувачеві пропонується на вибір найбільш зручний спосіб переказу грошей від пластикової картки до терміналу і мобільного телефону.

Інтернет-магазин має низку переваг у порівнянні зі звичайними магазинами:

- інтернет-магазин працює 24 години на добу, 365 днів на рік, без перерви на обід, без вихідних і святкових днів;

- необов'язково мати в наявності товар. Але в цьому випадку обов'язково мати мережу постачальників, що працюють за принципом «точно вчасно». Це гарантує доставку товару в заздалегідь обумовлені мінімальні терміни;

- не потрібно приміщення для продажів. Вітрини, стелажі, прилавки, квадратні метри і орендна плата – на всьому цьому можна істотно заощадити. На чому заощадити не можна-так це на якісно виконаному дизайні сайту для інтернет-магазину, оскільки саме зовнішній вигляд веб-сторінок буде замінювати покупцям і стелажі, і вітрини;

- термін і вартість створення інтернет-магазину незрівнянно нижче, ніж звичайного магазину;

- свобода пересувань продавця. Оскільки інтернет-магазин-це точка продажів у віртуальному просторі, ви можете керувати ним з будь-якого куточка земної кулі, де є доступ в інтернет;

- з інтернет-магазином можна отримати можливість розширити географію свого бізнесу аж до світових ринків. Із застереженням, що зміст сайту буде зрозуміло навіть іноземцям;

- можна не наймати продавців, адміністраторів, менеджерів, товарознавців, касирів, охоронців і ще кого б то не було. Професійно грамотно

створений інтернет-магазин може працювати повністю автономно. Практично без обслуговування;

- інтернет-магазин дає вам можливість самим розподіляти свій робочий час;

- для створення інтернет-магазину не потрібно отримання численних дозволів і ліцензій. Його не перевіряє пожежний інспектор, санепідемстанція та інші служби.

Однак крім переваг ведення торгівлі через інтернет-магазин є і істотні ризики:

- атака хакерів. У «кращому випадку» - внесуть плутанину в дані про замовлення, відправлення, розрахунки, і буде необхідно пояснюватися з усіма своїми клієнтами на предмет «Вибачте, це наша помилка». У гіршому випадку можуть вкрати базу даних з номерами кредитних карт клієнтів. Однак в Інтернеті є засоби забезпечують безпеку електронної торгівлі, тому є можливість інтегрувати в інтернет-магазин новітні захисні розробки;

- "баги". Це помилки в програмному забезпеченні, зроблені самими розробниками. Помилки є завжди. У будь-якому програмному забезпеченні для інтернет-торгівлі. Це може привести до того, що в певний момент, за певних умов магазин починає «глючити» або взагалі «відмовиться» працювати. Проблему з багами допомагає вирішити чітко функціонуюча служба технічної підтримки при компанії-розробника;

- купівля «кота в мішку». Купуючи інтернет-магазин, клієнт ніколи відразу не зрозуміє власником якого «добра» відтепер він є;

- клієнти легко приходять і йдуть. На щастя, в інтернет-магазин не треба «йти» - важливо знати адресу. Але, на жаль, щоб його покинути, також досить одного клацання «мишею».

Тим не менш, інтернет-магазини завойовують все більше довіри і популярність, і, безсумнівно, їх чекає велике майбутнє.

1.3 Програмні засоби для розробки систем онлайн-замовлень

1.3.1 Мова гіпертекстової розмітки HTML

Мова гіпертекстової розмітки, більш відома як HTML, є стандартною мовою, яка використовується для створення веб-сайтів. Як і все інше в технологічному світі, HTML дуже сильно змінився з моменту його створення в кінці 1980-х років [7-14].

HTML-код інструктує веб-браузер, як відображати контент. Він пояснює базову структуру веб-сторінки. Протягом десятиліть HTML вважався основою технології World Wide Web, а також каскадних таблиць стилів, або CSS, і JavaScript. Стандарти для HTML і CSS підтримуються консорціумом World Wide Web.

Як випливає з його назви, HTML не є технічно мовою програмування; це мова розмітки, яка використовується для організації даних з Інтернету. HTML-документи складаються з елементів HTML, представлених тегами, які написані всередині кутових дужок, для категоризації контенту.

HTML-документи, доставлені з веб-серверів, дозволяють браузеру знати, як представляти текст на додаток до вбудовування посилань, зображень та інтерактивних форм.

Останній великий перегляд HTML в кінці 1990-х років був виконаний завдяки спільних зусиль глобальних експертів з інтернаціоналізації мови та спрощення співпраці кодерів у всьому світі. В якості частини цього зусилля універсальний кодований набір символів був прийнятий як офіційний набір символів HTML. Цей крок дозволив браузерам краще представити безліч персонажів і акцентуацій, присутніх в людських мовах і діалектах.

В результаті тепер покращена індексація архівів для більш точного пошуку в Інтернеті на додаток до високоякісної друкарні. З введенням HTML4 веб-дизайнери також отримали більше контролю над швидкістю і порядком рендеринга контенту.

HTML5 є результатом співпраці між консорціумом World Wide Web або W3C, робочою групою Web Hypertext Application Technology або WHATWG. Організації об'єдналися в 2006 році, щоб зменшити залежність від плагінів, поліпшити обробку помилок. Отже, HTML5 значно спростив процес створення веб-додатків.

Завдяки HTML5 веб-сторінки тепер можуть зберігати дані локально в браузері користувача, що усуває необхідність використання файлів cookie HTTP. В результаті контент може бути доставлений швидше і безпечніше. HTML5 також спростив забезпечення узгодженості у всіх браузерах. Оскільки браузери традиційно покладалися на різні плагіни для відтворення мультимедійних файлів, вбудована підтримка відео та аудіо дозволяє розробникам уникнути проблем із сумісністю. Нові атрибути також дозволяють керувати відео, включаючи Опції відтворення, паузи і гучності.

Оскільки HTML4 був стандартом близько 15 років, багато кодерів все ще використовують його, і всі браузери будуть продовжувати підтримувати його протягом дуже довгого часу. Аналогічним чином, старі браузери можуть "пропускати" новий код HTML5 при рендерингу контенту. Практично всі сучасні браузери, включаючи Chrome, Firefox, Mozilla, Opera і Safari, тепер повністю підтримують специфікації HTML5. Оскільки HTML5 постійно розвивається, браузери поступово впроваджують підтримку нових функцій. На щастя, всі основні браузери постійно додають підтримку функцій у міру їх випуску.

Найважливішою відмінністю між старими версіями HTML та HTML5 є інтеграція відео та аудіо в специфікації мови. Крім того, HTML5 включає наступні оновлення:

- застарілі елементи, такі як center, font, і strike, були видалені;
- покращені правила синтаксичного аналізу дозволяють проводити більш гнучкий синтаксичний аналіз і Сумісність;
- додані нові елементи, video, time, nav, section, progress, meter, aside і canvas;

- додані нові атрибути `input`, включаючи адресу електронної пошти, URL, date I time;
- додані нові атрибути, включаючи `charset`, `async` і `ping`;
- додані нові API, які пропонують автономне кешування, підтримку перетягування та багато іншого;
- підтримка векторної графіки без допомоги таких програм, як Silverlight або Flash;
- підтримка MathML для кращого відображення математичних позначень;
- JavaScript тепер може працювати у фоновому режимі завдяки API JS Web;
- глобальні атрибути, такі як `tabindex`, `repeat` і `id`, можуть бути застосовані для всіх елементів;
- для розробників HTML5 також дає ряд переваг;
- деякі дані можуть зберігатися на пристрої Користувача, а це означає, Що додатки можуть продовжувати працювати належним чином без підключення до Інтернету;
- веб-сторінки можуть відображати більше шрифтів з більш широким набором кольорів, тіней та інших ефектів;
- об'єкти на сторінці можуть переміщатися у відповідь на переміщення курсору користувача;
- інтерактивні медіа, такі як Ігри, можуть працювати у веб-браузерах без необхідності додаткового програмного забезпечення або плагінів. Відтворення аудіо та відео також більше не вимагає додаткових плагінів;
- браузери можуть відображати інтерактивні 3D-графіки, використовуючи власний графічний процесор комп'ютера;
- обмежуючи необхідність у зовнішніх плагінах, HTML5 дозволяє швидше надавати більш динамічний контент.

Основним напрямком HTML5 було надання розробникам більшої гнучкості. HTML5 був задуманий з урахуванням декількох цілей.

Узгоджена обробка помилок.

У всіх браузерів є парсери для обробки синтаксично або структурно неправильного HTML-коду. Однак до недавнього часу для цього процесу не було письмового стандарту.

Підтримка додаткових можливостей веб-додатків.

Ще одна мета HTML5 полягала в тому, щоб браузери могли працювати як платформи додатків. У міру ускладнення веб-сайтів розробники повинні були знайти способи обійти "розширення браузера та інші серверні технології. HTML5 дає розробникам більше контролю над продуктивністю своїх сайтів.

Семантика розширених елементів.

Семантичні ролі деяких існуючих елементів були покращені. Нові елементи, такі як `section`, `header`, `article` і `nav`, можуть замінити більшість елементів `div`, що робить процес сканування помилок більш простим і зрозумілим.

Максимізована підтримка мобільних пристроїв.

HTML5 спрощує мобільну підтримку, обслуговуючи пристрої з низьким споживанням енергії, такі як смартфони та планшети.

Нове покоління розробників, знайде нові способи використання HTML5, а соціальні мережі, швидше за все, стимулюватимуть постійну еволюцію мови. W3C оголосила, що майбутні оновлення HTML5 визначатимуть пріоритети "основ програми", таких як інструменти конфіденційності. Одним з найбільш суперечливих пропозицій в HTML5 стало включення інструментів управління цифровими правами.

1.3.2 CSS3

Каскадні таблиці стилів (CSS) - це мова таблиць стилів, яка використовується для опису подання документа, написаного на мові розмітки HTML. Хоча CSS найчастіше використовується для встановлення візуального

стилю веб-сторінок та користувацьких інтерфейсів, написаних у HTML та XHTML, мова може бути застосована до будь-якого XML-документа, включаючи простий XML, SVG та XUL. Поряд з HTML та JavaScript, CSS-технологія, яка використовується більшістю веб-сайтів для створення візуально привабливих веб-сторінок, користувацьких інтерфейсів для веб-додатків та користувацьких інтерфейсів для багатьох мобільних додатків.

CSS розроблений в першу чергу для забезпечення поділу подання і Контенту, включаючи такі аспекти, як макет, кольори і шрифти. Цей поділ може поліпшити доступність контенту, забезпечити більшу гнучкість і контроль в специфікації характеристик презентації, дозволити декільком сторінкам HTML ділитися форматуванням, вказуючи відповідний CSS в окремому файлі .css і зменшуючи складність і повторення в структурному Контенті.

Поділ форматування і вмісту дозволяє представити одну і ту ж сторінку розмітки в різних стилях для різних методів рендеринга, таких як на екрані, у пресі, голосом (через мовний браузер або програмою читання з екрану) і на тактильній основі. CSS також може забезпечити відображення веб-сторінки по-різному в залежності від розміру екрану або пристрою перегляду.

Зміни в графічному дизайні документа (або сотнях документів) можуть бути застосовані швидко і легко, шляхом редагування декількох рядків у файлі CSS, які вони використовують, а не шляхом зміни розмітки в документах.

Специфікація CSS описує схему пріоритету, щоб визначити, які правила стилю застосовуються. У цьому так званому каскаді пріоритети (або ваги) обчислюються і присвоюються правилам, тому результати передбачувані.

Специфікації CSS підтримуються консорціумом World Wide Web (W3C). Тип інтернет-медіа (тип MIME) text / css зареєстрований для використання з CSS по RFC 2318 (березень 1998 р.). W3C використовує безкоштовну службу перевірки CSS для документів CSS.

CSS3 містить майже все, що включено в CSS2.1, попередню версію специфікації. CSS3 також додає нові функції, які допомагають розробникам

вирішити ряд проблем, пов'язаних з презентацією, не вдаючись до сценаріїв або додаткових зображень.

Нові функції в CSS3 включають підтримку додаткових селекторів, тіні, закруглених кутів, оновлені можливості компоновання, анімацію, прозорість і багато іншого.

У минулому для створення градієнтів, тіней і закруглених кутів веб-дизайнерам доводилося вдаватися до безлічі складних методів. Іноді потрібні додаткові HTML-елементи. У випадку градієнтів використання додаткових зображень було неминучим. CSS3 дозволяє включати ці та інші елементи дизайну в перспективне мислення, що призводить до безлічі переваг: більш чиста розмітка, Підтримуваний код, менша кількість сторонніх зображень і сторінки з більш швидким завантаженням.

CSS дозволяє легко застосовувати правила про форматування і макеті до елементів HTML, а потім повторно використовувати ці правила для декількох елементів і навіть сторінок.

1.3.3 ASP.NET

ASP.NET (Active Server Pages для .NET) - технологія створення веб-додатків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft .NET і розвитком старішої технології Microsoft ASP. На даний момент останньою версією цієї технології є ASP.NET 5.

ASP.NET зовні багато в чому зберігає схожість з більш старою технологією ASP, що дозволяє розроблювачам відносно легко перейти на ASP.NET. В той же час внутрішні пристрій ASP.NET істотно відрізняється від ASP, оскільки вона заснована на платформі .NET і, отже, використовує всі нові можливості, що надаються цією платформою.

Хоча ASP.NET бере свою назву від старої технології Microsoft ASP, вона значно від неї відрізняється. Microsoft повністю перебудувала ASP.NET, ґрунтуючись на Common Language Runtime(CLR), яка є основою всіх додатків

Microsoft .NET. розробники можуть писати код для ASP.NET, використовуючи практично будь-які мови програмування, що входять в комплект. NET Framework (C#, Visual Basic.NET і JScript. Net). ASP.NET має перевагу в швидкості в порівнянні зі скриптовими технологіями, так як при першому зверненні код компілюється і поміщається в спеціальний кеш, і згодом тільки виповнюється, не вимагаючи витрат часу на парсинг, оптимізацію, і т. д.

Переваги ASP.NET:

- компільований код виконується швидше, більшість помилок відловлюється ще на стадії розробки;
- значно покращена обробка помилок під час виконання запущеної готової програми, з використанням блоків try..catch;
- користувацькі елементи керування (controls) дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
- використання метафор, що вже застосовуються в Windows-додатках, наприклад, таких як елементи управління і події;
- розширюваний набір елементів управління і бібліотек класів дозволяє швидше розробляти програми;
- ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок VB.NET, Delphi.NET, Visual C#, J# і т. д.;
- можливість кешування всієї сторінки або її частини для збільшення продуктивності;
- можливість кешування даних, що використовуються на сторінці;
- можливість розділення візуальної частини та бізнес-логіки за різними файлами ("code behind»);
- розширювана модель обробки запитів;
- розширена подієва модель;
- розширювана модель серверних елементів управління;
- наявність master-сторінок для завдання шаблонів оформлення сторінок;
- підтримка CRUD-операцій при роботі з таблицями через GridView;

- вбудована підтримка AJAX;
- ASP.NET має перевагу в швидкості в порівнянні з іншими технологіями, заснованими на скриптах.

1.3.4 СУБД MS SQL Server

В даний час для побудови інформаційних систем застосовуються різні системи управління базами даних (СУБД), що розрізняються як своїми можливостями, так і вимогами до обчислювальних ресурсів. Все різноманіття застосовуваних СУБД, однак, можна звести до двох основних їх класів: персональні і розраховані на багато користувачів.

До першого класу відносяться СУБД, орієнтовані для роботи на персональному комп'ютері (dBASE, FoxPro, MS Access і т.п.). Спочатку вони підтримували роботу з даними тільки одного користувача. Вся СУБД такого класу виконується як єдина програма, таблиці бази даних представляються окремими файлами на диску того ж персонального комп'ютера.

СУБД другого класу спочатку створювалися для виконання на великих комп'ютерах і забезпечення паралельної роботи багатьох користувачів. Такі СУБД, як правило, складаються з ядра, постійно присутнього в пам'яті, (сервера) і великої кількості програм-агентів, які обслуговують запити кінцевих користувачів і прикладних програм (клієнтів). В цьому випадку і ядро СУБД, і дані знаходяться на одному і тому ж комп'ютері. Одна копія СУБД управляє однією копією даних. Єдина керуюча система дозволяє ефективно організувати одночасний доступ до даних багатьох агентів, запобігаючи конфліктам між ними. Помилка в роботі СУБД локалізована і може бути ефективно виправлена самої ж СУБД. При роботі в умовах мережі ядро СУБД виконує запити агентів на вибірку даних і передає по мережі тільки результати вибірки. Багатокористувацькі СУБД володіють також незаперечними перевагами в таких аспектах, як надійність, безпека, доступність. Багатокористувацькі СУБД

з самого початку своєї історії використовували в якості інтерфейсу запитів мову SQL, звідси сталося одне з їх альтернативних назв-SQL-сервери.

1.3.5 Веб-сервер MS IIS

Internet Information Services (IIS, раніше Internet Information Server) являє собою розширюваний веб-сервер, створений Microsoft для використання з ОС сімейства Windows NT. IIS підтримує HTTP, HTTPS, FTP, FTPS, SMTP і NNTP. Він є невід'ємною частиною ОС сімейства Windows NT, починаючи з Windows NT 4.0 , хоча може бути відсутнім в деяких виданнях (наприклад, Windows XP Home Edition), а не активний за замовчуванням.

Перший веб-сервер Microsoft був дослідницьким проектом European Microsoft Windows NT Academic Centre (EMWAC), який є частиною Единбурзького університету в Шотландії, і поширювався в якості безкоштовного веб-сервера. Так як сервер EMWAC не зміг впоратися з обсяг трафіку, що проходить через сайт Microsoft.com, то компанія Microsoft була змушена розробити свій власний веб-сервер, IIS.

Майже кожна версія IIS була випущена разом з новою версією ОС Microsoft Windows:

- IIS 1.0 був спочатку випущений в якості безкоштовного доповнення для Windows NT 3.51;
- IIS 2.0 був включений в Windows NT 4.0;
- IIS 3.0, який був включений з пакетом оновлень 2 в Windows NT 4.0, ввів активні серверні сторінки динамічного середовища сценаріїв;
- IIS 4.0 був випущений в рамках "Option Pack" для Windows NT 4.0;
- IIS 5.0 поставлявся з Windows 2000 і в ньому були введені додаткові методи аутентифікації, підтримка WebDAV протоколу, а також удосконалення ASP. У IIS 5.0 також була припинена підтримка протоколу Gopher;
- IIS 5.1 був випущений разом з Windows XP Professional, і був майже ідентична IIS 5.0 на Windows 2000;

- IIS 6.0, що входить до складу Windows Server 2003 і Windows XP Professional x64 Edition, була додана підтримка IPv6 і включена нова модель робочого процесу, що дозволило забезпечити більш високий рівень захисту, а також надійності;

- IIS 7.0 мав повністю перероблений дизайн, і поставлявся з Windows Vista і Windows Server 2008. IIS 7.0 мав нову модульну конструкцію, яка дозволяла зменшити ймовірність атак і збільшити продуктивність;

- IIS 7.5 був включений в Windows 7 і Windows Server 2008 R2. IIS 7.5 мав поліпшені WebDAV FTP модулі, а також адміністрування командного рядка в PowerShell. У ньому також представили TLS 1.1 і TLS 1.2 і підтримку інструментів і способів виділення bra для пулів додатків;

- IIS 8.0 доступна тільки в Windows Server 2012 і Windows 8. IIS 8.0 включає SNI (зв'язування SSL імен хостів, а не IP - адрес), централізовану підтримку сертифіката SSL, а також масштабування багатоядерних пума апаратних засобів, в числі інших нових функцій;

- IIS 8.5 включена в Windows Server 2012 R2 і Windows 8.1. Ця версія включає в себе динамічну активацію сайту, оптимізатор Logging, ETW ведення журналу;

- IIS 10 входить до Windows Server 2016 та Windows 10. Ця версія включає підтримку HTTP2.

Всі версії IIS до 7.0, що працюють на клієнтських операційних системах підтримують тільки 10 одночасних з'єднань до одного сайту.

IIS 6.0 і вище підтримують наступні автентифікації механізми:

- анонімна Перевірка автентичності;
- базова автентифікація доступу;
- дайджест встановлення автентичності доступу;
- вбудована перевірка автентичності Windows;
- перевірка автентичності UNC;
- Net Passport Authentication (видалено в Windows Server 2008 і IIS 7.0);

- перевірка автентичності сертифікатів.

IIS 7.0 має модульну архітектуру. Модулі, також звані розширення, можуть бути додані або видалені окремо, так що тільки модулі, необхідні для виконання певних функцій повинні бути встановлені. IIS 7 включає в себе власні модулі як частину повної установки. Ці модулі визначають індивідуальні особливості, які сервер використовує для обробки запитів і включають в себе наступні:

- модулі безпеки;
- модулі змісту;
- компресійні модулі;
- модулі кешування;
- модулі протоколювання та діагностики.

IIS 7.5 включає в себе наступні додаткові або розширені функції безпеки:

- зіставлення сертифікатів клієнтів;
- безпека ір;
- фільтрація запитів;
- URL авторизації.

Автентифікація трохи змінена між IIS 6.0 і IIS 7, в першу чергу в тому, що анонімний користувач, який був названий "IUSR_ {імя_комп'ютера}" є вбудованою обліковим запис в Vista і названий "IUSR". Примітно, що в IIS 7, кожен механізм автентифікації ізольований в свій власний модуль і може бути встановлений або видалений.

IIS 8.0 пропонує нові функції, орієнтовані на продуктивність і полегшує адміністрування:

- ініціалізація: функція, яка дозволяє адміністратору налаштувати певні програми для автоматичного запуску при запуску сервера. Це зменшує час очікування досвідчених користувачів, які мають доступ до сайту в перший раз після перезавантаження сервера;

- сторінка очікування: адміністратор може налаштувати сторінку очікування, яка буде відображатися відвідувачам сайту під час ініціалізації програми;

- підтримка ASP.NET 4.5: в IIS 8.0, ASP.NET 4.5 Включений за замовчуванням і IIS також пропонує кілька варіантів конфігурації для запуску його спільно з ASP.NET 3.5;

- централізована підтримка сертифіката SSL: функція, яка робить управління сертифікатами простіше, дозволяючи адміністратору зберігати і отримувати доступ до сертифікатів;

- Multicore масштабування NUMA: IIS 8.0 надає кілька варіантів конфігурації, які оптимізують продуктивність на системах NUMA, такі як запуск декількох робочих процесів в рамках одного пулу додатків і т. д.;

- підтримка протоколу WebSocket;

- Server Name indication (SNI): SNI є розширенням протоколу Transport Layer Security, який дозволяє пов'язувати кілька веб-сайтів з різними іменами хостів по одній IP – адресою;

- обмеження динамічної IP-адреси: функція, яка дозволяє адміністратору динамічно блокувати IP-адреси або діапазони IP, які відправляють на сервер велику кількість запитів.

IIS 8.5 має ряд удосконалень, пов'язаних з продуктивністю в великомасштабних сценаріях, наприклад тих, які використовуються комерційними хостинг-провайдерами і власними пропозиціями хмарних сервісів Microsoft. Він також має кілька додаткових можливостей:

- Idle робочих-сторінок відмови: функція зменшує час простою сайту, з метою щоб зменшити обсяг пам'яті;

- динамічна активація сайту: функція, яка реєструє прослуховування черги тільки тих сайтів, які отримали запити;

- покращений Logging: функція забезпечення збору змінних сервера, заголовків запитів і заголовків відповіді в журналах IIS;

- ETW протоколювання: постачальник ETW, який дозволяє збирати в режимі реального часу з використанням різних журналів подій інструментів трасування;
- автоматичний сертифікат Rebind: функція, яка визначає, коли сертифікат сайту був оновлений, і автоматично виконує повторну прив'язку сайту до нього.

1.3.6 Мова програмування C#

C # є елегантним і безпечним типом об'єктно-орієнтованої мови, який дозволяє розробникам створювати різні безпечні і надійні додатки, які працюють на платформі .NET Framework. Можна використовувати C # для створення клієнтських додатків Windows, веб-служб XML, розподілених компонентів, клієнт-серверних додатків, додатків баз даних, і багато чого багато чого іншого. Visual C # надає зручний редактор коду, дизайнерський користувальницький інтерфейс, інтегрований відладчик, і багато інших інструментів, щоб зробити розробку додатків, заснованих на мові C # i.NET Framework більш простою і зручною.

Мова C #. Синтаксис C # досить виразний, але це також простий і легкої вивчається мова програмування. Фігурні дужки з C # будуть миттєво впізнаваними для тих, хто знайомий з C, C ++ або Java. Розробники, які знають, будь-який з цих мов, як правило, в змозі почати продуктивно працювати в C # протягом дуже короткого часу. C # синтаксис спрощує багато складнощів C ++ і надає потужні функції, такі як обнуляються типи значень, перерахувань, делегатів, лямбда-виразів і прямого доступу до пам'яті, які не зустрічаються в Java. C # підтримує загальні методи і види, які забезпечують підвищену безпеку і продуктивність типів, і ітераторів, які дозволяють виконавцям КЛАСІВ визначати Користувальницькі поведінки ітерацій, які прості у використанні. Language-Integrated Query (LINQ) вирази роблять строго типізований запит більш легким і зрозумілим для розуміння.

Як об'єктно-орієнтована мова C # підтримує поняття інкапсуляції, успадкування та поліморфізму. Всі змінні і методи, в тому числі головний метод, точки входу додатки, інкапсулюються в визначеннях КЛАСІВ. Клас може успадковувати безпосередньо від одного батьківського класу, але може реалізувати будь-яку кількість інтерфейсів. Методи, які переписують віртуальні методи в батьківському класі вимагають перевизначення ключового слова як способу уникнути випадкового перевизначення. У с#, на структуру виділяється тип, який може реалізовувати інтерфейси, але не підтримувати успадкування.

На додаток до цих основних принципів об'єктно-орієнтованого C # дозволяє легко розробляти програмні компоненти через кілька інноваційних мовних конструкцій, в тому числі наступні:

- закриті методи підпису звані делегати, які дозволяють задавати повідомлення про події типу-сейф;
- властивості, які служать в якості засобу доступу для приватних змінних-членів;
- атрибути, які забезпечують декларативну метаданих про типи під час виконання;
- Language-Integrated Query (LINQ), надає вбудовані можливості запитів в різні джерела даних.

1.4 Постановка задачі розробки і дослідження системи для онлайн замовлень в харчовій крамниці

Об'єктом дослідження є інформаційне забезпечення веб-сайту систем онлайн замовлень.

Предметом дослідження є об'єктно-орієнтований аналіз та проектування онлайн систем замовлень в харчовій крамниці.

Метою даної роботи є розробка і реалізація системи онлайн замовлень в харчовій крамниці (інтернет-магазину).

Відповідно до мети роботи були поставлені і вирішені наступні основні задачі:

- виконати огляд аналогів і аналіз інструментів для розробки веб-додатки для онлайн замовлень продуктового магазину;
- розробити програмну частину веб-додатку для онлайн замовлень продуктового магазину;
- дослідити веб-додаток для онлайн замовлень продуктового магазину і надати рекомендації по його використанню.

Інформаційна система буде реалізована у вигляді веб-програми, яка повинна забезпечувати наступні функції [15-17]:

- надання інформації про товари;
- реєстрацію нових клієнтів;
- вибір клієнтом товару;
- оформлення замовлення на товар;
- оформлення доставки товару;
- управління профілю клієнтів;
- аналіз продажів по групах товарів;
- аналіз продажів по клієнтам;
- підтримка клієнтської бази даних;
- сегментація клієнтської бази.

Програма повинна забезпечувати взаємодію користувача з системою з використанням web-інтерфейсу.

Слід зазначити, що сама наявність розробленої ІС не гарантує виконання якісного обліку та аналізу всіх аспектів діяльності інтернет-магазину. Тому що, крім інструментарію для наповнення бази даних і сегментації клієнтів необхідно мати методику розробки директ-маркетингових пропозицій щодо збільшення лояльності клієнтів.

Перша версія системи виконує лише базові завдання. У майбутні випуски будуть включені додаткові функції, можливості та засоби, що забезпечують легкість і простоту використання.

У базовій версії система не матиме засобів, що дозволяють здійснювати зв'язок із зовнішніми по відношенню до наявної БД джерелами даних та інформації.

Інформаційна система повинна дозволяти оператору вводити всю необхідну інформацію. При роботі з системою співробітники мають можливість використовувати дані зберігаються в базі даних. Співробітники мають можливість перегляду, редагування та друку документів з бази даних. Всі дані зберігаються на сервері БД доступні по локальній мережі підприємства, на будь-якому комп'ютері, на якому встановлена клієнтська частина інформаційної системи.

Для функціонування інформаційної системи необхідно, як мінімум, технічне забезпечення, що задовольняє наступним вимогам:

а) сервер (додатків і бази даних):

- процесор: Intel Pentium Xeon від 2GHz;
- RAM: 2048 Мб;
- вільний дисковий простір: 72 Gb.

б) «Тонкий» клієнт:

- процесор: Intel Pentium IV від 2 GHz;
- RAM: 1024 Мб;
- вільний дисковий простір 500 Мб.

Телекомунікаційна інфраструктура повинна забезпечити оперативну інформаційну взаємодію сервера веб-додатків і сервера баз даних і забезпечувати їх стійку його роботу.

Клієнтські станції і сервери працюють в локальній обчислювальній мережі з пропускною здатністю не менше 10 Mbit/c (рекомендується 100 Mbit / c).

Розробка та підтримка системи буде здійснюватися під управлінням ОС Windows 8.1 та вище. Використовуване апаратне забезпечення: сервер БД під управлінням будь-якої серверної ОС Windows Server 2012, ПК-клієнти під управлінням ОС Windows 8.1 та вище, мережеве обладнання (СПД, мережеві

адаптери, Комутатори). На сервері повинна бути встановлена і запущена інформаційна система MS SQL Server 2018 Express Edition, на клієнтських машинах повинні бути встановлені програми-клієнти для підключення і роботи з Інтернет (Opera, IE і т.д.).

Вимоги до зберігання даних. Всі дані системи повинні зберігатися в структурованому вигляді під управлінням реляційної СУБД MS SQL Server 2018 Express Edition. Винятки становлять файли конфігурації., призначені для налаштування Програми. Такі файли зберігаються у файловій системі.

Вимоги до мов програмування. Для реалізації програмного продукту використовується мова програмування C#.

2 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ ДЛЯ ОНЛАЙН ЗАМОВЛЕНЬ В ХАРЧОВІЙ КРАМНИЦІ

2.1 Загальна структура програми

При виконанні проектів для більш повного відображення прийнятих проектних рішень рекомендується будувати діаграми компонентів для наступних уявлень [18-19]:

- 1) модель вихідного програмного коду системи;
- 2) модель виконуваного програмного коду системи;
- 3) модель артефактів системи, що поставляється замовнику.

На першій діаграмі слід відобразити розподіл класів файлів з вихідним кодом; на другий — представити за допомогою стереотипів розширення «compile» і (або) «build» трансформацію вихідного коду у виконуваний код, тобто показати взаємозв'язок між одними і тими ж артефактами, представленими в різних фізичних форматах; на третій — представити зв'язок зазначених програмних артефактів з інформаційними та іншими артефактами, тобто з базою даних і СУБД.

На рис. 2.1 та рис. 2.2 представлені приклади перерахованих раніше варіантів діаграм компонентів. У реальному проекті крім проблемних класів, операції яких реалізують бізнес-логіку програми, існує велика кількість допоміжних класів-це всілякі форми, що формують користувальницький інтерфейс [20-25]. З цієї причини модель вихідного коду може виявитися дуже об'ємною. Якщо врахувати, що в деяких системах програмування з кожною формою зв'язується не один, а кілька файлів, то для того, щоб діаграма не «потонула» в деталях, або обмежуються зазначенням основних асоційованих файлів, або не виносять допоміжні класи на діаграми.

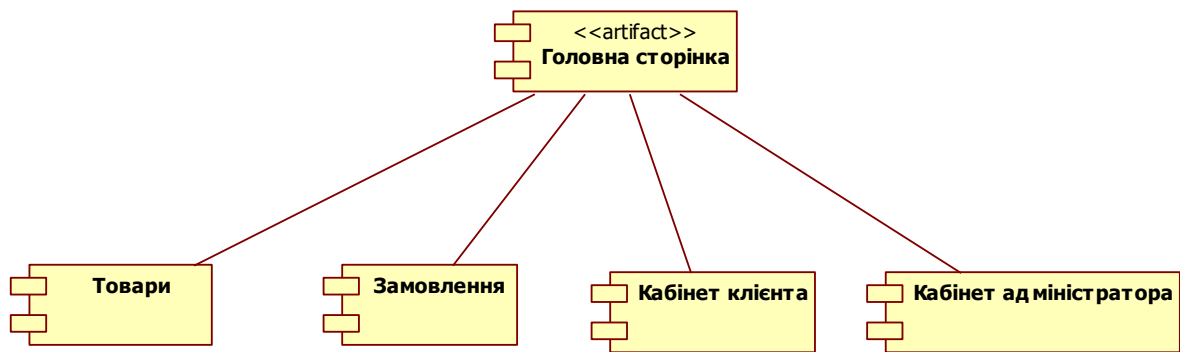


Рисунок 2.1-Діаграма компонентів модель вихідного програмного коду



Рисунок 2.2 – Діаграма компонентів - модель виконуваного програмного коду

Із рисунка 2.2 видно, що для сторінок порталу існують .cs сторінки з програмним кодом, який реалізують динамічну складову для aspx сторінок системи. Цей артефакт являє собою вихідний код створюваної програми.

На рис. 2.3 представлено найпростіше проектне рішення, в якому інтерфейс користувача відділений від бізнес-логіки. Цей компонент (його прийнято називати «товстим клієнтом») взаємодіє з базою даних через інтерфейси СУБД — стороннього компонента, що використовується в проєктованій системі.

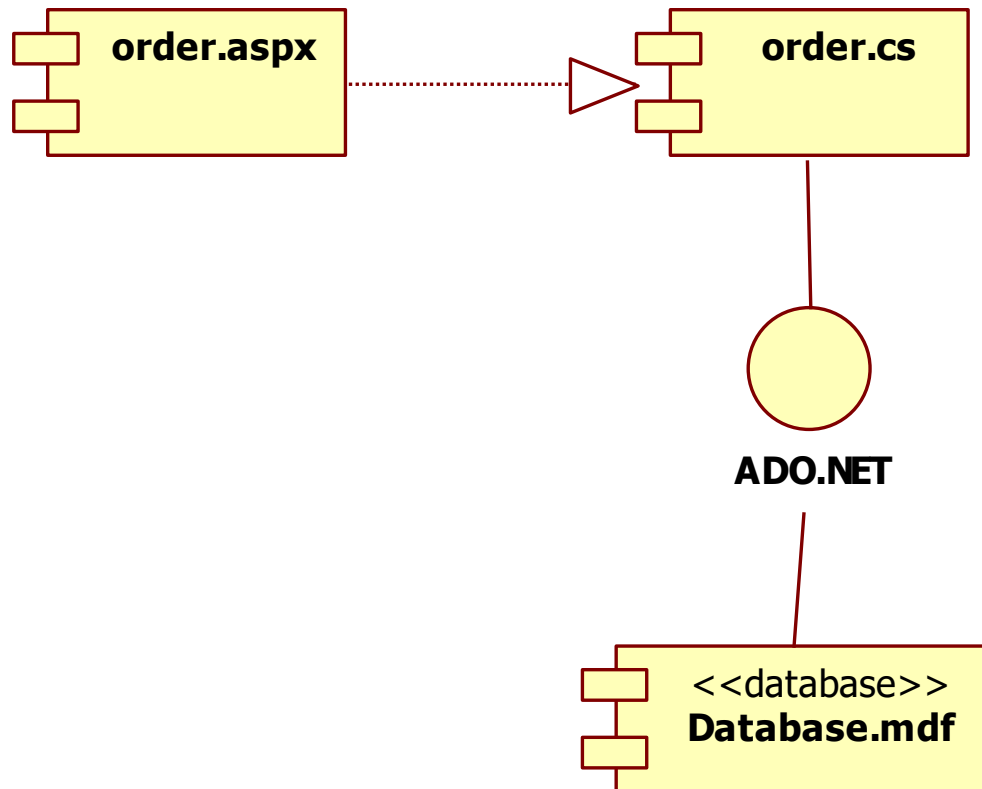


Рисунок 2.3 – Діаграма компонентів – модель артефактів системи з триланковою архітектурою

Розміщення артефактів програми по фізичним пристроям відображається діаграмою розгортання (рис. 2.4).

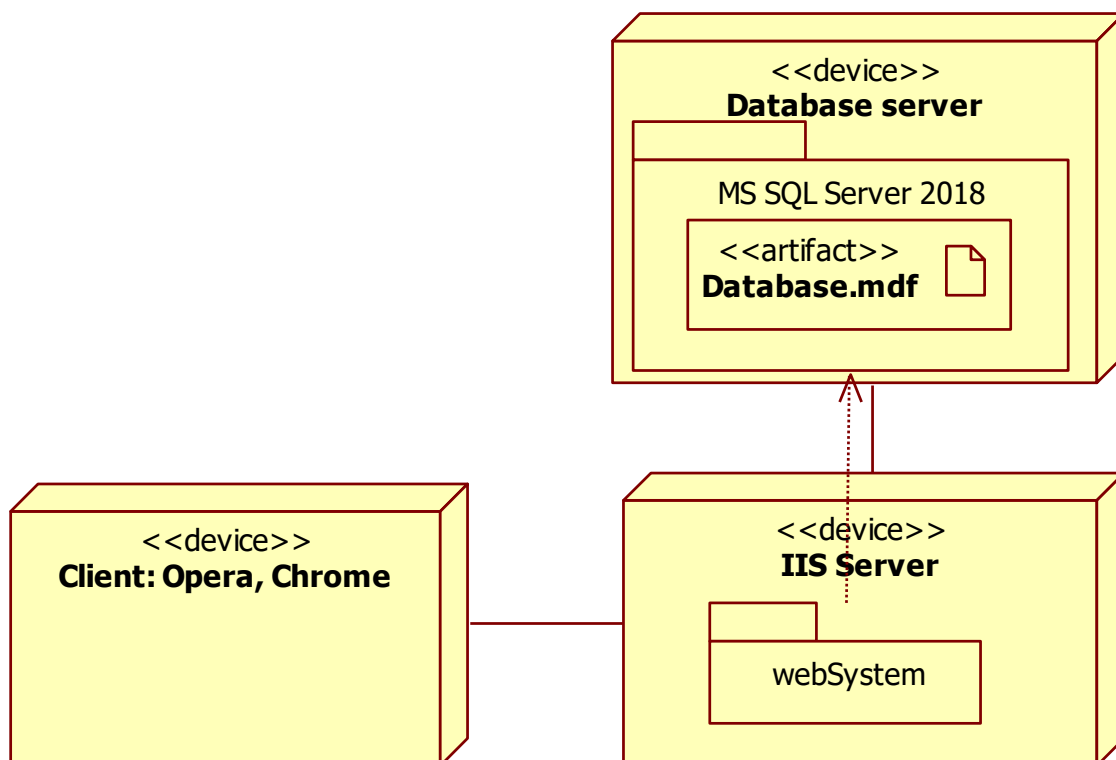


Рисунок 2.4 – Діаграма розгортання-артефакти розподілені по трьох комп'ютерах

Основними вузлами на цій діаграмі є комп'ютери, в яких розміщуються артефакти програми, основними з яких є фізичні втілення користувацького інтерфейсу, бізнес-логіки і бази даних.

2.2 Модуль реєстрації та авторизації

Модуль реєстрації та авторизації відповідає за реєстрацію нових користувачів с системі онлайн-замовлень та авторизацію їх в системі. З цією метою було розроблено форму авторизації та форму реєстрації.

Макет сторінки авторизації містить заголовок, в якому є випадające меню для доступу до функціоналу онлайн-системи (рис. 2.5), то форма реєстрації (рис. 2.6), які містить поля для введення логіну (email) та паролю.



Рисунок 2.5 – Заголовок сторінки авторизації

Рисунок 2.6 – Форма реєстрації

Програмний код, що реалізує інтерфейс сторінки реєстрації представлено на рисунку 2.7.

```

1  <%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Registr.aspx.cs" Inherits="
2
3  <asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
4      <!-- Main -->
5      <article id="main">
6
7          <header class="special container">
8              <span class="icon fa-check"></span>
9              <h2>Авторизація в системі онлайн замовлень</h2>
10
11          </header>
12
13          <!-- One -->
14          <section class="wrapper style4 special container small">
15
16              <!-- Content -->
17              <div class="content">
18                  <form>
19                      <div class="row half no-collapse-1">
20                          <div class="6u">
21                              <asp:Label ID="Label1" runat="server" Text="Email"></asp:Label> </br>
22                              <asp:TextBox ID="TextBox1" runat="server" ></asp:TextBox>
23
24                          </div>
25                          <div class="6u">
26                              <asp:Label ID="Label2" runat="server" Text="Пароль"></asp:Label> </br>
27                              <asp:TextBox ID="TextBox2" runat="server" TextMode="Password"></asp:TextBox>
28
29                          </div>
30                      </div>
31
32                      <div class="row">
33                          <div class="12u">
34                              <ul class="buttons">

```

Рисунок 2.7 – Фрагмент програмного кода для сторінки авторизації

Якщо клієнт ще немає свого акаунта в системі, то він повинен пройти реєстрацію. Для цього використовують сторінку реєстрації (рис.2.8). Вона містить поля для вводу email, прізвища клієнта, телефону та пароля. Макет сторінки реєстрації представлено на рисунку 2.9.



Рисунок 2.8 – Макет сторінки реєстрації нового клієнта

Рисунок 2.9 – Макет форми реєстрації

Фрагмент програмного коду для сторінки реєстрації представлено на рисунку 2.10.

```

NewUser.aspx  MasterPage.master  Registr.aspx
1 <% Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="NewUser.aspx.cs" Inherits="Default2" %>
2
3 <asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
4 <!-- Main -->
5 <article id="main">
6
7 <header class="special container">
8 <span class="icon fa-check-circle"></span>
9 <h2>Реєстрація нового клієнта в системі онлайн замовлень</h2>
10
11 </header>
12
13 <!-- One -->
14 <section class="wrapper styled special container small">
15
16 <!-- Content -->
17 <div class="content">
18 <form>
19 <div class="row half no-collapse-1">
20 <div class="6u">
21 <asp:Label ID="Label1" runat="server" Text="Email"></asp:Label> </br>
22 <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
23
24 </div>
25 <div class="6u">
26 <asp:Label ID="Label2" runat="server" Text="Прізвище імя клієнта"></asp:Label> </br>
27 <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
28
29 </div>
30 </div>
31 <div class="row half no-collapse-1">
32 <div class="6u">
33 <asp:Label ID="Label5" runat="server" Text="Пароль"></asp:Label> </br>
34 <asp:TextBox ID="TextBox5" runat="server" TextMode="Password"></asp:TextBox>
35

```

Рисунок 2.10 – Фрагмент програмного коду для сторінки реєстрації

2.3 Модуль онлайн-замовлень

Для олайн-замовлень потрібно перейти з головної сторінки на сторінку Каталог (рис.2.11) на сторінку Каталогу товарів. Макет головної сторінки з меню переходу представлено на рис.2.11, а макет сторінки Каталогу товарів представлено на рисунку 2.12.

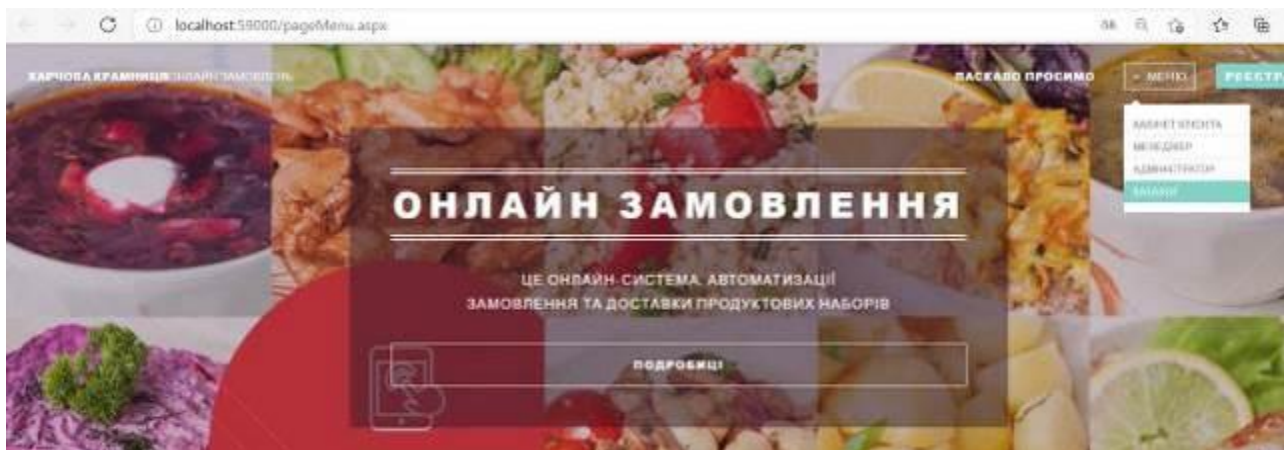


Рисунок 2.11 – Макет головної сторінки з меню

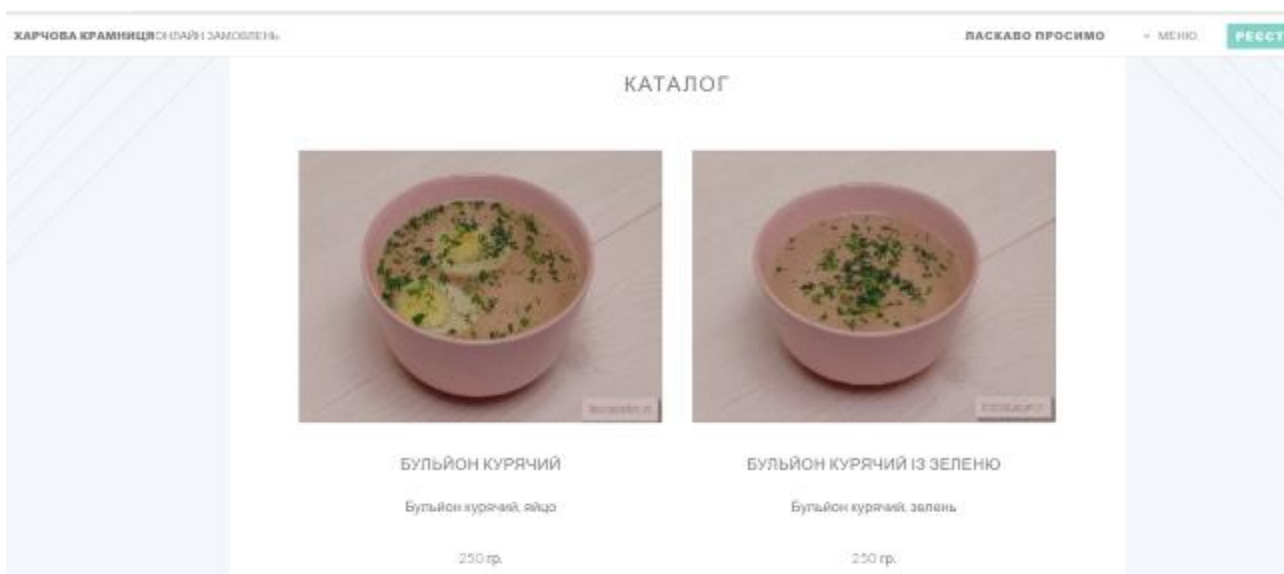


Рисунок 2.12 – Макет сторінки Каталог товарів

Для оформлення замовлення необхідно перейти на сторінку оформлення замовлень, макет якої представлено на рисунку 2.13. Програмний код сторінки оформлення замовлень представлено на рисунку 2.14. Сторінка оформлення

замовлень містить поля для введення номера замовлення, прізвища замовника, адресу та телефон замовника, а також поле для приміток.

Рисунок 2.13 – Макет форми оформлення замовлення

```

Default.aspx  GenerateSign.aspx  pageManager.aspx  pageMenu.aspx*  SmartContract.aspx  NewUser.aspx  MasterPage.master
1  <% Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="SmartContract.aspx.cs" Inherits="Default2" %>
2
3  <asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
4      <!-- Main -->
5      <article id="main">
6
7          <header class="special container">
8              <span class="icon fa-check-circle">/span>
9              <h2>Оформлення доставки замовлення</h2>
10
11          </header>
12
13          <!-- One -->
14          <section class="wrapper style4 special container small">
15
16              <!-- Content -->
17              <div class="content">
18                  <form>
19                      <div class="row half no-collapse-1">
20                          <div class="6u">
21                              <asp:Label ID="Label1" runat="server" Text="Номер замовлення">/asp:Label> </div>
22                              <asp:TextBox ID="TextBox1" runat="server">/asp:TextBox>
23
24                          </div>
25                          <div class="6u">
26                              <asp:Label ID="Label2" runat="server" Text="Прізвище замовника">/asp:Label> </div>
27                              <asp:TextBox ID="TextBox2" runat="server">/asp:TextBox>
28
29                          </div>
30                      </div>
31
32                      <div class="row half no-collapse-1">
33                          <div class="6u">
34                              <asp:Label ID="Label3" runat="server" Text="адреса доставки">/asp:Label> </div>
35                              <asp:TextBox ID="TextBox3" runat="server">/asp:TextBox>

```

Рисунок 2.14 – Фрагмент програмного коду для сторінки оформлення замовлень

2.4 Модуль адміністрування веб-програми

Модуль адміністрування веб-програми містить сторінку для відслідковування та зміни статусу замовлень. Ця сторінка дозволяє адміністратору (менеджеру) відслідковувати замовлення клієнтів та проставляти статус їх виконання. Макет сторінки відслідковування замовлень клієнтів представлено на рисунку 2.15. На рисунку 2.16 представлено фрагмент програмного коду для сторінки для адміністрування замовлень клієнтів.

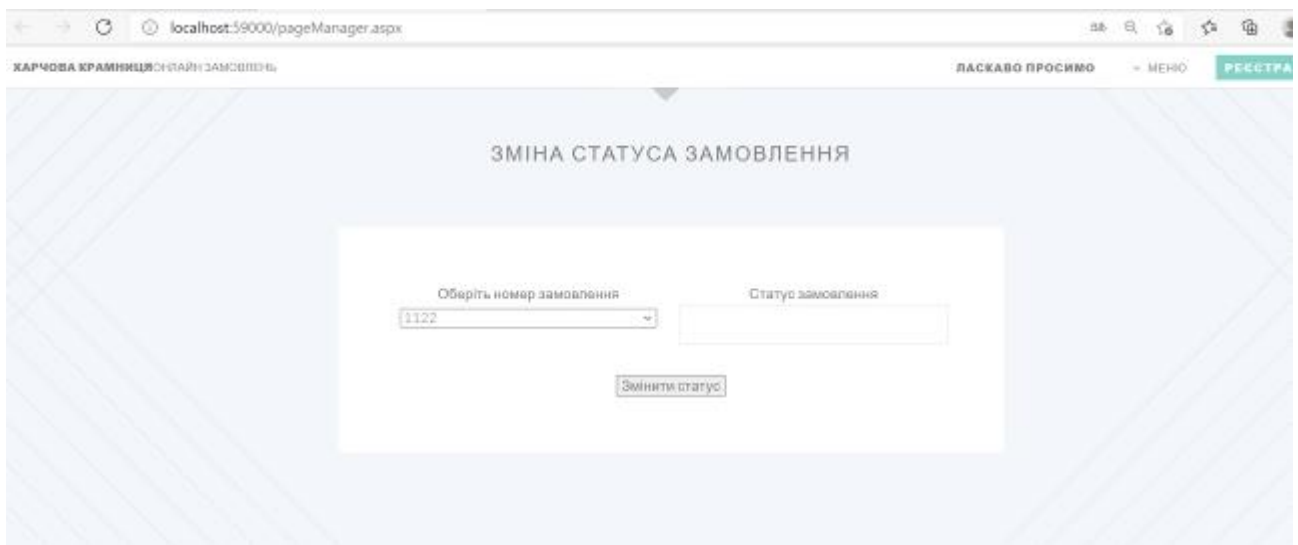


Рисунок 2.15 – Макет сторінки адміністрування замовлень клієнтів онлайн-крамниці

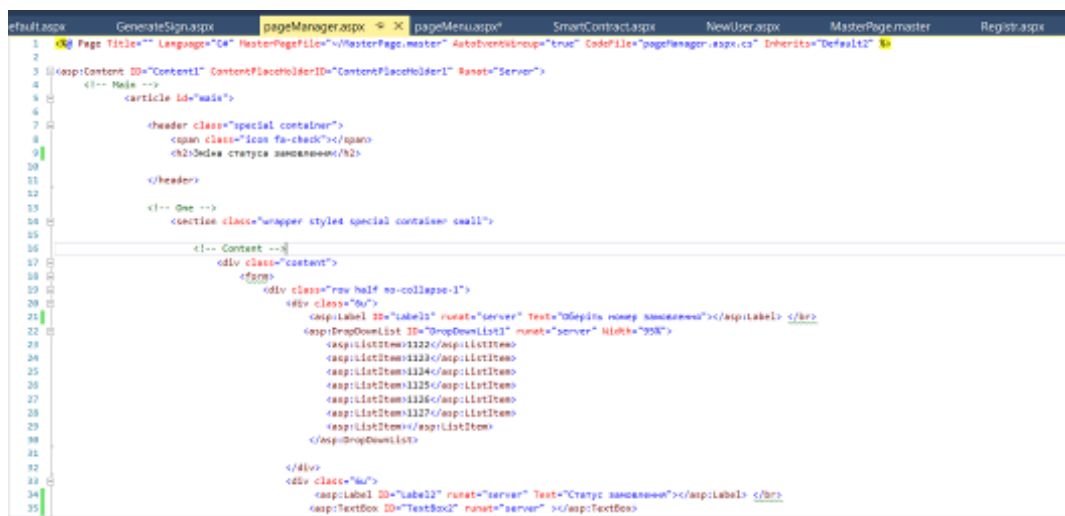


Рисунок 2.16 – Фрагмент програмного коду сторінки

У другому розділі було розроблено опис програмного застосунку. Описано основні модулі системи онлайн-замовлень в харчовій крамниці: Модуль реєстрації та авторизації, Модуль онлайн-замовлень, Модуль адміністрування. Наведено макети сторінок цих модулів та фрагменти програмного коду.

З ДОСЛІДЖЕННЯ СИСТЕМИ ДЛЯ ОНЛАЙН ЗАМОВЛЕНЬ В ХАРЧОВІЙ КРАМНИЦІ І РЕКОМЕНДАЦІЇ ПО ВИКОРИСТАННЮ

3.1 Розробка тестових сценаріїв для дослідження веб-застосунку

У вузькому розумінні тестування – це процес визначення відповідності об'єкта тестування заданим специфікаціям (характеристикам, функціям), який полягає в опрацюванні програмою послідовності різноманітних контрольних наборів тестів з відомими результатами. Тести підбираються так, щоб вони охопили найрізноманітніші типи можливих ситуацій. У контексті розробки, відповідно, тестування розуміється як процес визначення відповідності продукту первісним специфікаціям, які були задані технічним завданням.

Якщо розглядати тестування в широкому сенсі, то його можна характеризувати як процес експериментального аналізу функціональності деякої досліджуваної системи. При цьому постулюється, що всякий випадок тестування можна визначити як дослідження. Наприклад, тестування з метою виявлення структурних дефектів (дефектів конструкції) можна визначити як дослідження системи на предмет наявності дефектів конструкції.

Всяке тестування має на увазі двох дійових осіб: суб'єкт і об'єкт тестування. Суб'єктом тестування (тобто джерелом активності виступає "Тестувальник" (tester) – спеціально призначена особа, в обов'язки якої входить виконання тестування. Під об'єктом тестування розуміється досліджувана система. При цьому в ролі тестованої системи може виступати як весь продукт, так і окремі його модулі, складові частини.

При тестуванні програмного забезпечення, в загальному випадку, об'єктом тестування є версія розроблюваного програмного продукту.

Зрозуміло, що під версією мається на увазі не тільки «фінальна» (кінцева) версія, але і проміжні версії, одержувані в ході розробки.

При проведенні тестування завжди має місце «зовнішній» вплив «на систему з боку тестувальника, який також називають» тестовим впливом".

Тестування полягає в здійсненні тестувальником спрямованого впливу на тестовану систему, який передбачає отримання деякої очікуваної реакції, яку називають «тестовою реакцією», поява якої має підтвердити відповідність системи конкретної окремо взятої специфікації. Отримання зворотної (щодо очікуваної) або неспецифічної реакції дозволяє говорити про невідповідність системи специфікації і приймати рішення про повернення Програмного Продукту на доопрацювання з метою усунення дефекту.

Як і в моделюванні, в тестуванні поняття системи є ключовим, оскільки наше уявлення про досліджуваний об'єкт є первинним стосовно експерименту. Іншими словами, характер експерименту залежить від наших знань про структурний склад і характер зв'язків між компонентами тестованої системи

Необхідно відзначити, що програмні продукти в силу різних причин мають модульну структуру і рівневу організацію, і, як правило, функціонують не відокремлено, а в рамках деякої інфраструктури (спільно з іншими додатками, програмними комплексами, а також в рамках програмних середовищ). Вищевказане призводить до виникнення різних рівнів тестування:

- компонентне або модульне тестування перевіряє функціональність і шукає дефекти в частинах програми, які доступні і можуть бути протестовані окремо в силу своєї автономності (модулі програми, об'єкти, функції і т.д.). Ознакою, що утворює даний рівень, є те, що тестування окремих модулів можна виконувати на ранніх етапах розробки, поки робота над іншими модулями триває. Один з найефективніших підходів до компонентного (модульного) тестування – це підготовка автоматизованих тестів до початку

основного кодування ПЗ. Це називається розробкою від тестування або підходом тестування спочатку;

- інтеграційне тестування призначене для перевірки зв'язків між компонентами Програми, а також дослідження взаємодії програми з середовищем, в рамках якої воно буде виконуватися;

- системне тестування спрямоване на дослідження функціональних і нефункціональних особливостей системи в цілому. При цьому виявляються дефекти, такі як неправильне використання ресурсів системи, непередбачувані комбінації даних рівня користувача, несумісність з оточенням, непередбачені сценарії використання, відсутня або невірна функціональність, незручність використання і т.д.;

- приймальне тестування проводиться на фінальному етапі розробки і спрямовані на з'ясування того, чи відповідає система приймально / здавальним критеріям до вимог замовника. Приймальне тестування виконується відповідно до плану приймальних робіт. Рішення про проведення приймального тестування приймається, коли продукт досяг необхідного рівня якості і замовник ознайомлений з Планом приймальних робіт або іншим документом, де описаний набір дій, пов'язаних з проведенням приймального тестування, дата проведення, Відповідальні і т.д.

Методи приймального тестування:

- тестування замовником самотійно. Це ризиковано в тому плані що у замовника може не бути, а Завантаження за поточними завданнями може розтягнути процес приймання;

- тестування третьою стороною (аудит). Наймається спеціалізована компанія на тестування або підписується договір з конкурентом постачальника на надання послуг аудиту;

- спільне тестування за сценаріями з замовником. Постачальник допомагає готувати пакет матеріалів для приймального тестування, готує команду замовника до методичного приймального тестування, контролює хід приймального тестування і терміни його виконання.

Присутність інженера з тестування з боку виконавця допоможе краще зафіксувати розбіжності, зауваження та виявлені дефекти.

Фаза приймального тестування триває до тих пір, поки замовник не виносить рішення про відправку програми на доопрацювання або реліз програми.

Незважаючи на те, що приймання знаходиться в кінці етапу (а в невеликих проектах і в кінці проекту) – готуватися до неї потрібно заздалегідь і перший прогін потрібно робити трохи раніше – щоб визначитися з повнотою і якістю робочого набору артефактів приймання, привчити до нього замовника, заздалегідь виявити можливі проблеми в приймальних тестах або в продукті.

Залежно від цілей тестування, виділяють три основних види тестування програмного забезпечення:

а) функціональні види тестування пов'язані з дослідженням зовнішньої поведінки системи, тобто виконуваних нею функцій. До них належить:

1) функціональне тестування, спрямоване на перевірку коректності виконуваних системою функцій і може бути присутнім на всіх рівнях тестування;

2) тестування безпеки, яке спрямоване на перевірку безпеки системи, а також оцінку цілісності підходу до захисту програми від несанкціонованого доступу та захисту конфіденційних даних;

3) тестування взаємодії, яке спрямоване на оцінку можливості програми взаємодіяти із зовнішніми компонентами або системами, а також включає тестування сумісності та інтеграційне тестування;

б) нефункціональні види тестування, які спрямовані на перевірку всіх нефункціональних особливостей системи. Сюди відносяться тести специфічних для програмних продуктів характеристик:

1) тестування установки направлено на перевірку інсталяції системи, а також процесів настройки, вилучення та оновлення програмного забезпечення;

2) тестування зручності використання, яке пов'язане з оцінкою ступенем зручності використання, а також зрозумілості і привабливості користувальницького інтерфейсу програми;

3) тестування на відмову і відновлення, яке пов'язане з оцінкою засобів забезпечення відмовостійкості і надійності системи;

4) конфігураційне тестування, спрямоване на перевірку функціональності системи при всіх можливих конфігураціях програмного забезпечення та обладнання, підтримуване системою;

5) тестування продуктивності (навантаження), яке складається з таких методів тестування як:

- тестування навантаження, що полягає в дослідженні реакції системи на функціонування в умовах навантаження (мається на увазі навантаження в межах норми);

- стресове тестування, що передбачає дослідження поведінки системи при функціонуванні в умовах перевантаження (навантаження, що перевищує штатну);

- тестування стабільності і надійності, спрямоване на вивчення поведінки системи в умовах нормального навантаження при тривалому функціонуванні;

- об'ємне тестування, що використовується для оцінки поведінки системи за умови збільшення обсягу даних обробляються додатком.

На всьому протязі процесу створення програми Від макетування форм, до написання програмних алгоритмів, проводилося тестування відображень розміщених елементів і їх взаємодія між собою. Завжди тестувалися ділянки коду при додаванні того чи іншого функціоналу на коректність роботи і отримання кінцевого результату.

Головним критерієм тестування програмного забезпечення є розмежування на те – як працює функція на саме справі і те, який очікуваний результат закладається після виконання функції.

У нашому програмному продукті закладено кілька основних функцій:

- авторизація користувача в системі;
- реєстрація нового клієнта;
- перегляд даних про товари онлайн-крамниці;
- реєстрація нового замовлення товару.

У таблиця 3.1 наведені тестові сценарії.

Таблиця 3.1 – Тестові сценарії для перевірки функціоналу системи онлайн замовлень

Функціонал	Очікуваний результат	Отриманий результат
Авторизація користувача в системі	При завантаженні програми відкриється вікно авторизації. Воно містить поля введіть логін, введіть пароль. Після заповнення полів необхідно натиснути кнопку Увійти. Якщо логін і пароль співпадають, то користувач отримає доступ до основної сторінки програми. В іншому випадку виводиться повідомлення про помилку.	Функція відпрацьовує відповідно до очікувань
Реєстрація нового клієнта	Для реєстрації нового клієнта необхідно натиснути кнопку Реєстрація. Після цього відкриється сторінка реєстрації з полями для введення. Після заповнення форми реєстрації натиснути кнопку Реєстрація.	Функція відпрацьовує відповідно до очікувань
Перегляд даних про товари онлайн-крамниці	Після авторизації в системі користувачеві доступно головне вікно програми. При натисканні на кнопку Каталог відкриється сторінка зі списком товарів онлайн-крамниці	Функція відпрацьовує відповідно до очікувань
Реєстрація нового замовлення товару	Після авторизації в системі користувачеві доступно головна сторінка програми. При натисканні на кнопку Додати на сторінці Каталог користувач має можливість заповнити форму замовлення. Після заповнення форми необхідно натиснути кнопку Замовити.	Функція відпрацьовує відповідно до очікувань

3.2 Дослідження та тестування веб-програми

У попередньому розділі були описані тестові сценарії для тестування основного функціоналу системи онлайн-замовлень. Виконаємо тестування системи за допомогою тестових сценаріїв (таблиця 3.2-3.5, рисунки 3.2-3.8).

Таблиця 3.2 – Тестовий сценарій функції: Авторизація користувача в системі

Функціонал	Очікуваний результат	Отриманий результат
Авторизація користувача в системі	При завантаженні програми відкриється вікно авторизації (рис. 3.1). Воно містить поля введіть логін, введіть пароль. Після заповнення полів необхідно натиснути кнопку Увійти. Якщо логін і пароль співпадають, то користувач отримає доступ до основної сторінки програми. В іншому випадку виводиться повідомлення про помилку (рис. 3.2).	Функція відпрацьовує відповідно до очікувань

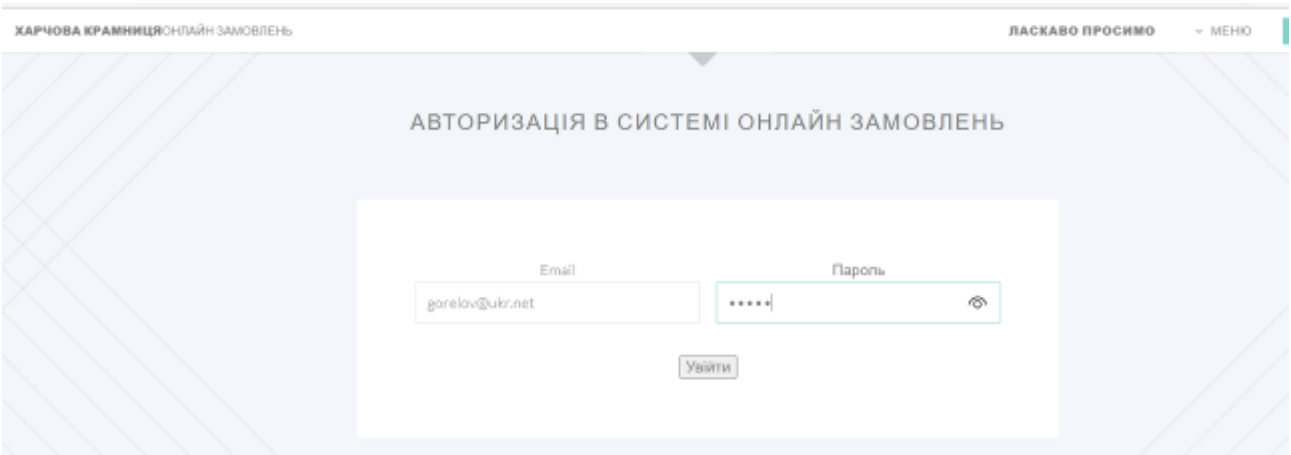


Рисунок 3.1 – Сторінка авторизації



Рисунок 3.2 – Повідомлення про помилку

Таблиця 3.3 – Тестовий сценарій функції: Реєстрація нового клієнта

Реєстрація нового клієнта	Для реєстрації нового клієнта необхідно натиснути кнопку Реєстрація. Після цього відкриється сторінка реєстрації з полями для введення (рис. 3.3). Після заповнення форми реєстрації натиснути кнопку Реєстрація. Якщо реєстрація успішна, то буде виведено повідомлення (рис. 3.4)	Функція відпрацьовує відповідно до очікувань
---------------------------	---	--

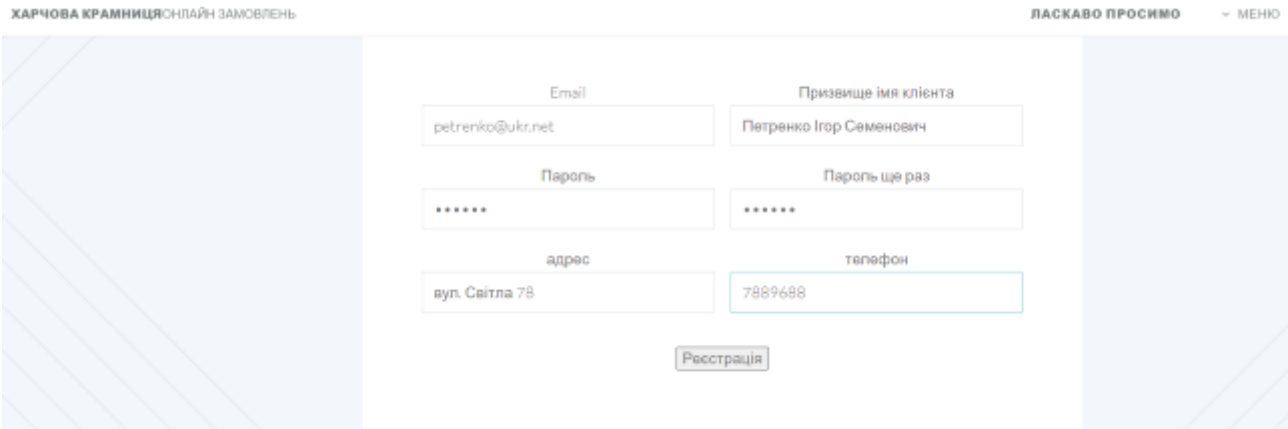


Рисунок 3.3 – Сторінка реєстрації



Рисунок 3.4 – Повідомлення про успішну реєстрацію

Таблиця 3.4 – Тестовий сценарій функції: Перегляд даних про товари онлайн-крамниці

Перегляд даних про товари онлайн-крамниці	Після авторизації в системі користувачеві доступно головне вікно програми. При натисканні на кнопку Каталог (рис. 3.5) відкриється сторінка зі списком товарів онлайн-крамниці (рис. 3.6)	Функція відпрацьовує відповідно до очікувань
---	---	--



Рисунок 3.5 – Вибір пункту меню Каталог

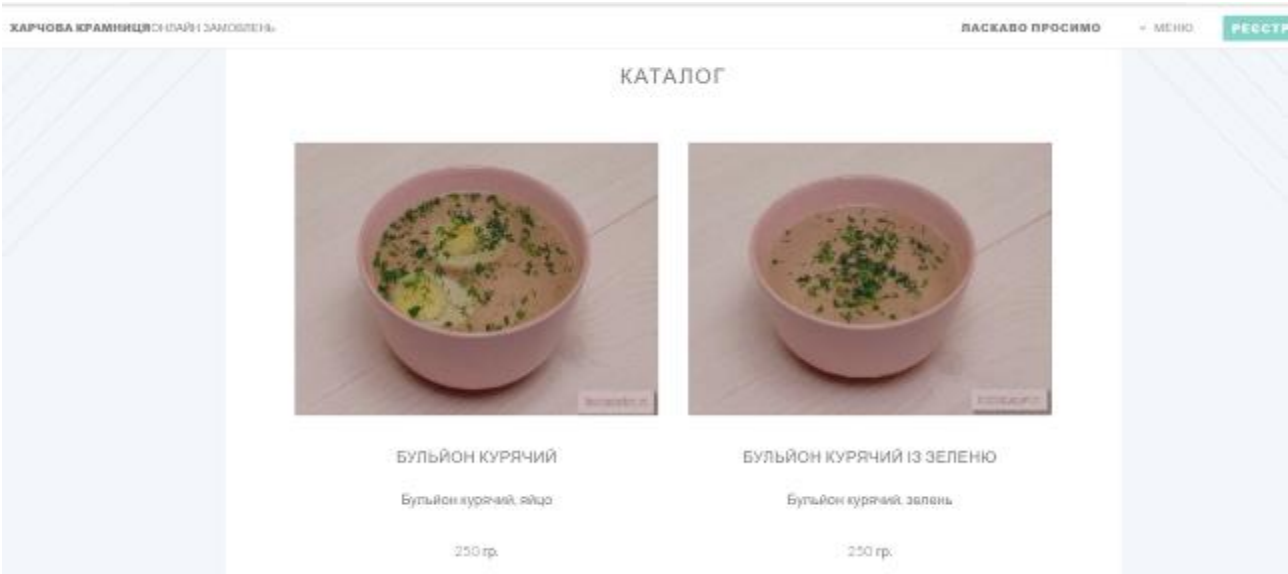


Рисунок 3.6 – Сторінка Каталог

Таблиця 3.5 – Тестовий сценарій функції: Реєстрація нового замовлення товару

Реєстрація нового замовлення товару	Після авторизації в системі користувачеві доступно головна сторінка програми. При натисканні на кнопку Додати на сторінці Каталог користувач має можливість заповнити форму замовлення (рис. 3.7). Після заповнення форми необхідно натиснути кнопку Замовити (рис. 3.8).	Функція відпрацьовує відповідно до очікувань
-------------------------------------	---	--

ОФОРМЛЕННЯ ДОСТАВКИ ЗАМОВЛЕННЯ

Номер замовлення 111	Прізвище замовника Петренко Ігор Семенович
адреса доставки вул.Світла 78	телефон 7889688
Примітки домофон	

[Замовити](#)

Рисунок 3.7 – Сторінка замовлення

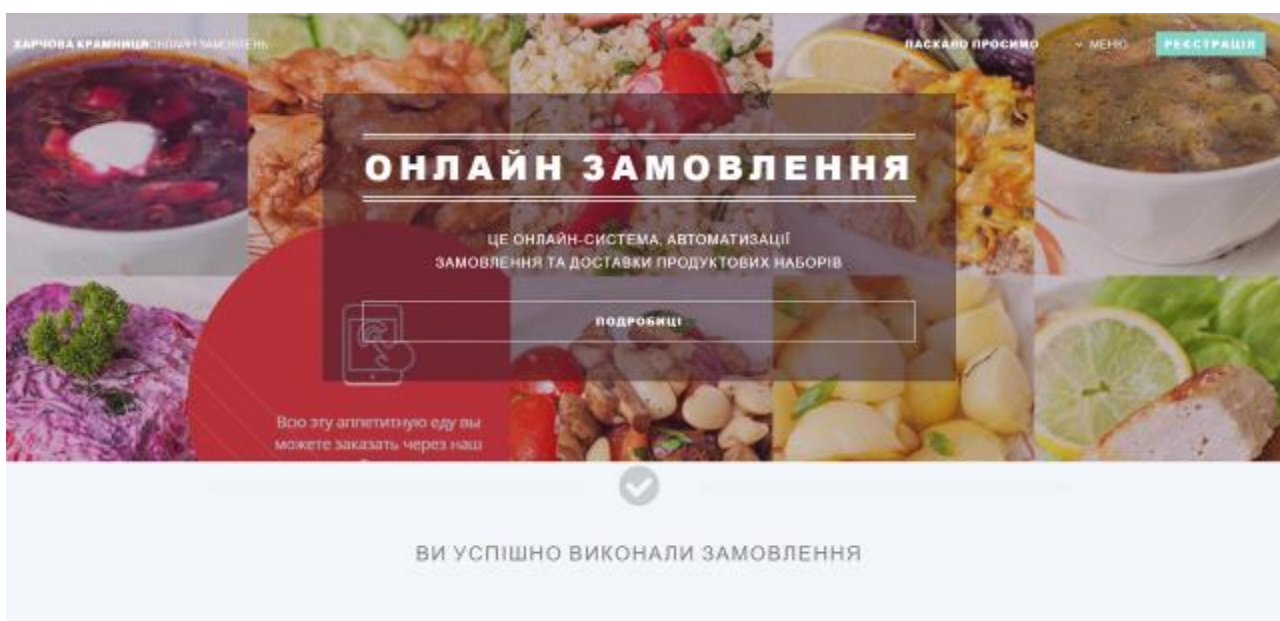


Рисунок 3.8 – Повідомлення про реєстрацію замовлення

3.3 Рекомендації щодо вдосконалення веб-програми

Грунтуючись на досвіді аналогічних інтернет-магазинів, припускаємо, що основна аудиторія покупців складеться вже через 6 місяців, тобто через 2 квартали. Саме на цьому етапі доцільно припиняти оплачувати ефективне SEO-

просування і формування подальшої маркетингової діяльності на поєднанні більш дешевих методів.

Стратегія просування сайту-це частина загальної стратегії компанії, тому для її складання важливо визначити:

- що просуваєте бренд або продукт;
- потрібен трафік або впізнаваність;
- цільову аудиторію;
- конкурент.

Загальні налаштування сайту активізують просування і посприяють його подальшої розкрутки, а сервіс Google Analytics допоможуть відстежувати результати просування сайту:

- необхідно встановити сервіси Google Analytics;
- налаштувати цілі в Google Analytics;
- додати сайт в панель вебмастера Google Webmasters Tools;
- створити файл robots.TXT і XML-карту і завантажити їх в кореневий каталог сайту;
- налаштувати генерацію адрес сторінок за принципами людино зрозумілого URL;
- налаштувати Canonical URL, закрити дублі сторінок від індексації;
- зменшити час завантаження сторінок сайту;
- налаштувати внутрішню перелінковку сайту (max 100 посилань на сторінки);
- додати https протокол, для більшої довіри до сайту.

Семантичне ядро сайту - це перелік ключових слів і словосполучень, які описують основну спрямованість і тематику сайту. Від смислового поля сайту залежить Технічна оптимізація, підбір цільових сторінок, і формування контенту. Складання семантичного ядра сайту забезпечить ефективність просування і приверне цільову аудиторію.

Семантичне ядро сайту: продукти, готові страви, доставки, замовлення. Одночасно аналізуються ключові слова конкурентів.

Технічний аудит-це важливий крок у просуванні сайту, на якому виявляються і усуваються недоробки, що перешкоджають успішній оптимізації сайту. При проведенні технічного аналізу:

- максимальна вага html-коду 100-200 кілобайт;
- зменшення швидкість завантаження сторінки до 3-5 секунд;
- флеш не повинен містити важливий для індексації контент;
- перевірити , чи відсутні фрейми;
- простежити, щоб в html-кодi сторінки не було зайвого сміття;
- закрити від індексації серверні логи, адміністративну панель і піддомени з тестовою версією;
- прописати на сторінках відповідне кодування;
- позбавитися від дублів сторінок, які не доступні за адресами /index.aspx або / index.html;
- вибрати основне дзеркало (з www або без).

У третьому розділі наведено опис тестових сценаріїв для тестування системи онлайн-замовлень в харчовій крамниці. Виконано дослідження функціоналу розробленої системи за допомогою створених тестових сценаріїв. Наведено рекомендації щодо подальшого вдосконалення розробленої системи онлайн замовлень в харчовій крамниці.

ВИСНОВКИ

Метою даної роботи є розробка і реалізація онлайн системи замовлення в харчовій крамниці.

Відповідно до мети роботи були поставлені і вирішені наступні основні задачі:

- виконано огляд аналогів та інструментів розробки веб-додатків для онлайн замовлень;
- розроблена програмна частина веб-додатку для онлайн замовлень;
- досліджено програмну частину в веб-додатку для онлайн замовлень.

У роботі використані методи об'єктно-орієнтованого аналізу та проектування .

У першому розділі виконано огляд аналогів та інструментів розробки веб-додатків для онлайн замовлень та виконано формулювання завдання, а також зроблено вибір моделі проектування. Проводиться опис обраних в роботі засобів проектування і розробки. В якості СУБД обраний MS SQL Server. Для розробки клієнт-серверної веб-орієнтованої системи використовується технологія ASP.NET, в якості мови програмування C#.

У другому розділі проведено дослідження та аналіз предметної області, а також виконано проектування предметної області. наведено опис функціональних вимог до проектованого web-додатку. Описано макети веб-сторінок порталу, а також програмний код для реалізації бізнес-логіки програми. У розділі описаний інтерфейс системи і наведені приклади використання системи.

У третьому розділі виконано тестування розробленої системи. Описано тестові сценарії для дослідження онлайн системи оформлення замовлень в харчовій крамниці. На основі описаних тестових сценаріїв виконано дослідження розробленої системи онлайн замовлень в харчовій крамниці.

Наведено рекомендації щодо подальшого розвитку та вдосконалення веб-додатку для онлайн замовлень.

В роботі створено веборієнтований додаток, який може застосуватися для онлайн-замовлень у харчовій крамниці. Вебдодаток дозволяє зареєструватися в персональному кабінеті , оформляти замовлення та переглядати статус замовлення. Додаток розрахований на широке коло користувачів. В подальшому пропонується розширення функціоналу додатку за рахунок використання адаптивного дизайну та додавання функцій онлайн-оплати замовлення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Бердышев, С.Н. Информационный маркетинг / С. Н. Бердышев. – М. : Дашков и Ко, 2010. – 216 с.
2. Burns, A.C. Marketing Research / A. C. Burns, R. F. Bush. – Pearson, 2014. – 497 p.
3. Данченко, Л.А. Маркетинг в социальных медиа. Интернет-маркетинговые коммуникации / Данченко, Л.А. – П. : СПб., 2013. – 288 с.
4. Ашманов И. , Иванов А.. Оптимизация и продвижение сайтов в поисковых системах. Издательство «Питер» 2011. — 464 с.
5. Ашманов И. , Иванов А.. Оптимизация сайтов в поисковых системах. Издательство «Питер» 2010. — 304 с.
6. Загуменов, А. П. Как раскрутить и разрекламировать Web_сайт в сети Интернет. – М. : ДМК Пресс, 2005. – 384 с.
7. Балдин, К. В. Информационные системы в экономике: учебник / К. В. Балдин, В. Б. Уткин. – М. : Дашков и К, 2013. – 395 с.
8. Васильков, А.В. Информационные системы и их безопасность: учебное пособие / А. В. Васильков, А. А. Васильков, И. А. Васильков. – М. : Форум, 2013. – 528 с.
9. Горбенко, А.О. Информационные системы в экономике / А. О. Горбенко. – М.: БИНОМ. ЛЗ, 2012. – 292 с.
10. Исаев, Г. Н. Информационные системы в экономике : учебник для студентов вузов / Г. Н. Исаев. – М. : Омега-Л, 2013. — 462 с.
11. Норенков, И. П. Автоматизированные информационные системы : учебное пособие / И.П. Норенков. – М. : МГТУ им. Баумана, 2011. – 342 с.
12. Уткин, В.Б. Информационные системы в экономике: учебник для студентов высших учебных заведений / В.Б. Уткин, К.В. Балдин. – М. : ИЦ Академия, 2012. – 288 с.

13. Ясенев, В.Н. Информационные системы и технологии в экономике : учебное пособие для студентов вузов / В.Н. Ясенев. – М. : ЮНИТИ-ДАНА, 2012. – 560 с.
14. Амириди, Ю.В. Информационные системы в экономике. Управление эффективностью банковского бизнеса : учебное пособие / Ю. В. Амириди, Е. Р. Кочанова, О. А. Морозова; под ред. Д. В. Чистов. — М. : КноРус, 2011. – 174 с.
15. Александров, Д. В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы : учебное пособие / Д. В. Александров. – М. : ФиС, 2011. – 224 с.
16. Олейник, П. П. Корпоративные информационные системы : учебник для вузов. Стандарт третьего поколения / П. П. Олейник. – СПб. : Питер, 2012. – 176 с.
17. Патрушина, С. М. Информационные системы в экономике : учебное пособие / С. М. Патрушина, Н. А. Аручиди. – М. : Мини Тайп, 2012. – 144 с.
18. Маклафлин, Б. Объектно-ориентированный анализ и проектирование / Б. Маклафлин. – СПб. : Питер, 2013. – 608 с.
19. Пирогов, В. Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В. Ю. Пирогов. – СПб. : БХВ-Петербург, 2009. – 528 с.
20. Baskerville, R. Designing Organizational Systems: An Interdisciplinary Discourse / R. Baskerville, M. De Marco, P. Spagnoletti (Eds.). – Springer, 2013 – 342 p.
21. Burgess, Mark. In Search of Certainty: The Science of Our Information Infrastructure. – O'Reilly Media, 2015. – 473 p.
22. Емельянова, Н.З. Проектирование информационных систем: Учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. - М.: Форум, 2013. - 432 с.

23. Заботина, Н. Н. Проектирование информационных систем : учебное пособие / Н. Н. Заботина. – М. : НИЦ ИНФРА-М, 2013. – 331 с.
24. Исаев, Г. Н. Проектирование информационных систем: учебное пособие / Г. Н. Исаев. – М. : Омега-Л, 2013. – 424 с.
25. Coates, K. An Introduction to Information Design / K. Coates, A. Ellison. – Laurence King Publishing, 2014. – 208 p.

ДОДАТОК А

A.1 Текст програми MasterPage

```

<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!DOCTYPE html>

<html>
<head runat="server">
    <title>Система онлайн замовлень харчової крамниці</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta name="description" content="" />
    <meta name="keywords" content="" />
    <!--[if lte IE 8]><script src="css/ie/html5shiv.js"></script><![endif]-->
    <script src="js/jquery.min.js"></script>
    <script src="js/jquery.dropotron.min.js"></script>
    <script src="js/skel.min.js"></script>
    <script src="js/skel-layers.min.js"></script>
    <script src="js/init.js"></script>
    <noscript>
        <link rel="stylesheet" href="css/skel.css" />
        <link rel="stylesheet" href="css/style.css" />
        <link rel="stylesheet" href="css/style-noscript.css" />
    </noscript>
    <!--[if lte IE 8]><link rel="stylesheet" href="css/ie/v8.css" /><![endif]-->
>
    <!--[if lte IE 9]><link rel="stylesheet" href="css/ie/v9.css" /><![endif]-->
>
</head>
<body class="index loading">
    <form id="form1" runat="server">
    <!-- Header -->
        <header id="header" class="alt">
            <h1 id="logo"><a href="Default.aspx">Харчова
крамниця<span>онлайн замовлень</span></a></h1>
            <nav id="nav">
                <ul>
                    <li class="current"><a
href="Default.aspx">Ласкаво просимо</a></li>
                    <li class="submenu">
                        <a href="">Меню</a>
                    </li>
                </ul>
            </nav>
        </header>
    </form>

```

```

sidebar.html">Кабінет клієнта</a></li>
sidebar.html">Менеджер</a></li>
sidebar.html">Адміністратор</a></li>
href="pageMnu.aspx">Каталог</a></li>
</ul>
</li>
<li><a href="NewUser.aspx" class="button
special">Реєстрація</a></li>
</ul>
</nav>
</header>

<!-- Banner -->
<section id="banner">

<!--
".inner" is set up as an inline-block so it automatically
expands
in both directions to fit whatever's inside it. This
means it won't
automatically wrap lines, so be sure to use line breaks
where
appropriate (<br />).
-->
<div class="inner">

<header>
<h2>Онлайн замовлення</h2>
</header>
<p>Це <strong>онлайн-система</strong>,
автоматизації<br />
замовлення та доставки продуктових наборів<br
/>
</p>
<footer>
<ul class="buttons vertical">
<li><a href="#main" class="button fit
scrolly">Подробиці</a></li>
</ul>

```

```

        </footer>

    </div>

</section>

<div>
    <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

        </asp:ContentPlaceHolder>
    </div>
    <!-- Footer -->
        <footer id="footer">

            <span class="copyright">&copy; Система онлайн-
заказов. All rights reserved. </span>

        </footer>

    </form>
</body>
</html>

```



```
class="buttons">
    <ul
    <li>
    <asp:Button ID="Button1" runat="server"
    Text="Увійти" />
    </li>
    </ul>
    </div>
    </div>
    </form>
    </div>
    </section>
    </article>
</asp:Content>
```



```

        <asp:Label ID="Label5" runat="server"
Text="Пароль"></asp:Label> </br>
        <asp:TextBox ID="TextBox5" runat="server"
TextMode="Password"></asp:TextBox>

        </div>
        <div class="6u">
            <asp:Label ID="Label6" runat="server" Text="Пароль
ще раз"></asp:Label> </br>
            <asp:TextBox ID="TextBox6" runat="server"
TextMode="Password"></asp:TextBox>

        </div>
    </div>
    <div class="row half no-
collapse-1">

        <div class="6u">
            <asp:Label ID="Label3" runat="server" Text="адрес
"></asp:Label> </br>
            <asp:TextBox ID="TextBox3" runat="server"
></asp:TextBox>

        </div>
        <div class="6u">
            <asp:Label ID="Label4" runat="server"
Text="телефон"></asp:Label> </br>
            <asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>

        </div>
    </div>
    <div class="row">
        <div class="12u">
            <ul
class="buttons">

                <li>
                    <asp:Button ID="Button1" runat="server"
Text="Реєстрація" />

                </li>
            </ul>
        </div>
    </div>
</form>

```

</div>

</section>

</article>

</asp:Content>


```

        <asp:Label ID="Label3" runat="server" Text="адреса
доставки"></asp:Label> </br>
        <asp:TextBox ID="TextBox3" runat="server"
></asp:TextBox>

        </div>
        <div class="6u">
            <asp:Label ID="Label4" runat="server" Text="телефон
"></asp:Label> </br>
            <asp:TextBox ID="TextBox4"
runat="server"></asp:TextBox>

        </div>
        <div class="6u">
            <asp:Label ID="Label6" runat="server"
Text="Примітки"></asp:Label> </br>
            <asp:TextBox ID="TextBox6"
runat="server"></asp:TextBox>

        </div>
    </div>
    <div class="row">
        <div class="12u">
            <ul
class="buttons">
                <li>
                    <asp:Button ID="Button1" runat="server"
Text="Замовити" />
                </li>
            </ul>
        </div>
    </div>
</form>
</div>

</section>

</article>
</asp:Content>

```

Додаток Б
СЛАЙДИ ПРЕЗЕНТАЦІЇ

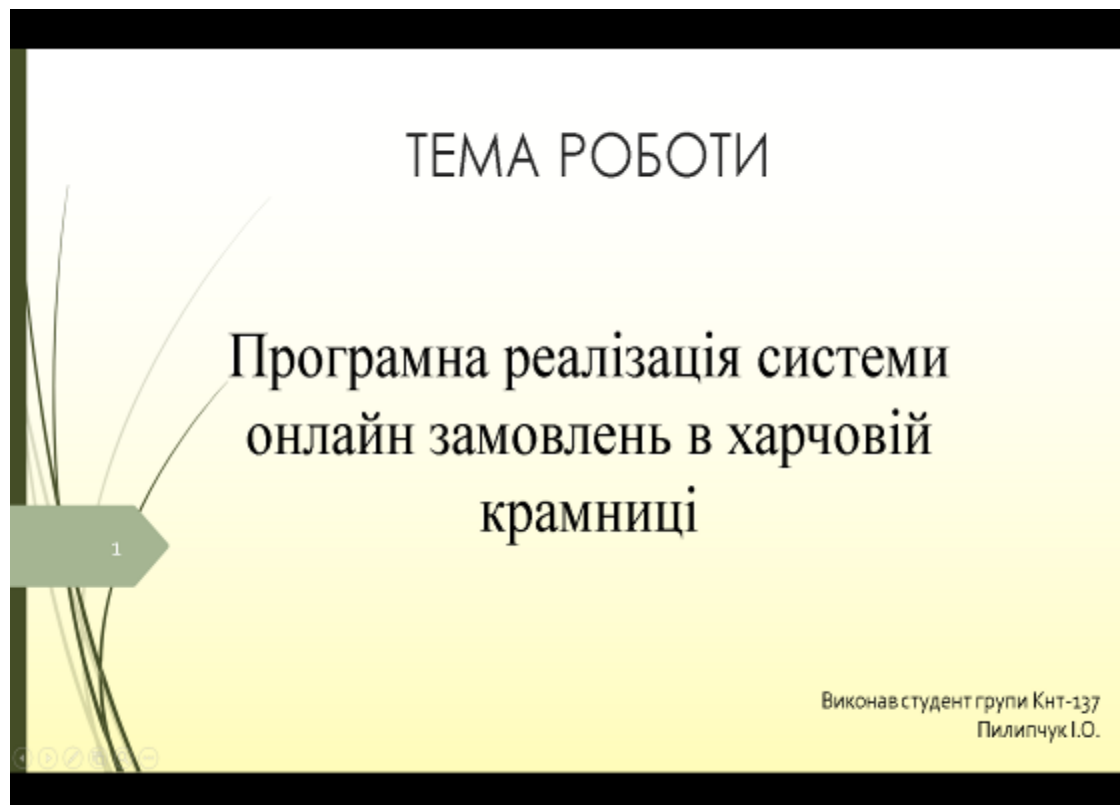


Рисунок Б.1 - Слайд 1

2

Мета роботи



- Об'єкт дослідження – є інформаційне забезпечення вебсайту систем онлайн замовлень.
- Предмет дослідження – є об'єктно-орієнтований аналіз та проектування онлайн систем замовлень в харчовій крамниці.
- Мета роботи – є розробка і реалізація системи онлайн замовлень в харчовій крамниці.

Рисунок Б.1 - Слайд 2

3

Функції програми

- надання інформації про товари;
- реєстрацію нових клієнтів;
- вибір клієнтом товару;
- оформлення замовлення на товар;
- оформлення доставки товару;
- управління профілю клієнтів;
- аналіз продажів по групах товарів;
- аналіз продажів по клієнтам;
- підтримка клієнтської бази даних;
- сегментація клієнтської бази.

Рисунок Б.1 - Слайд 3



Рисунок Б.1 - Слайд 4

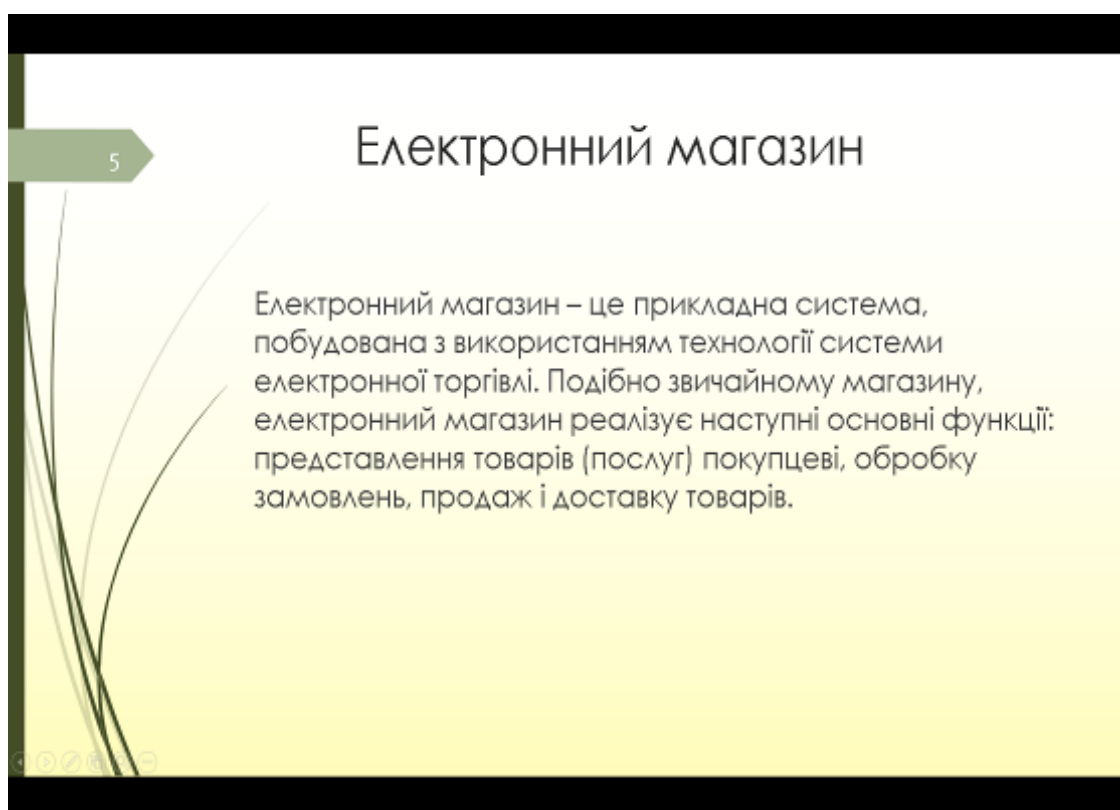


Рисунок Б.1 - Слайд 5

6

PHP Vs ASP.NET: Що краще?

Порівняння	PHP	ASP.Net
1.Доступність	Безкоштовно	Потрібно купувати
2.Простота	Легкий	Складно вивчається
3. Швидкість роботи	-	+
4. Підтримка мов програмування	-	+
5. Інформує про помилки	-	+
6.Підтримка Windows	+	+
7. Портал спільноти	Більше	Менше

Рисунок Б.1 - Слайд 6

7

Діаграма компонентів
модель вихідного
програмного коду



Діаграма компонентів -
модель виконуваного
програмного коду



Рисунок Б.1 - Слайд 7

8

Діаграма компонентів – модель артефактів системи з триланковою архітектурою

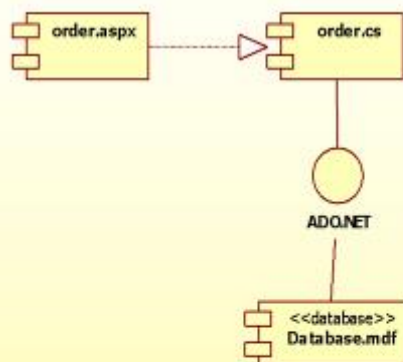


Рисунок Б.1 - Слайд 8

9

Сторінка авторизації

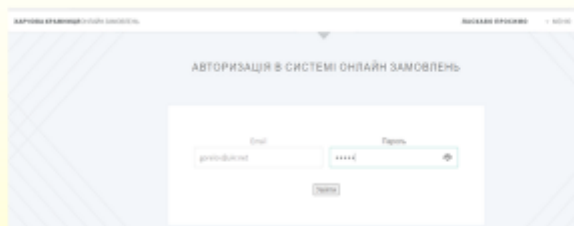


Рисунок Б.1 - Слайд 9

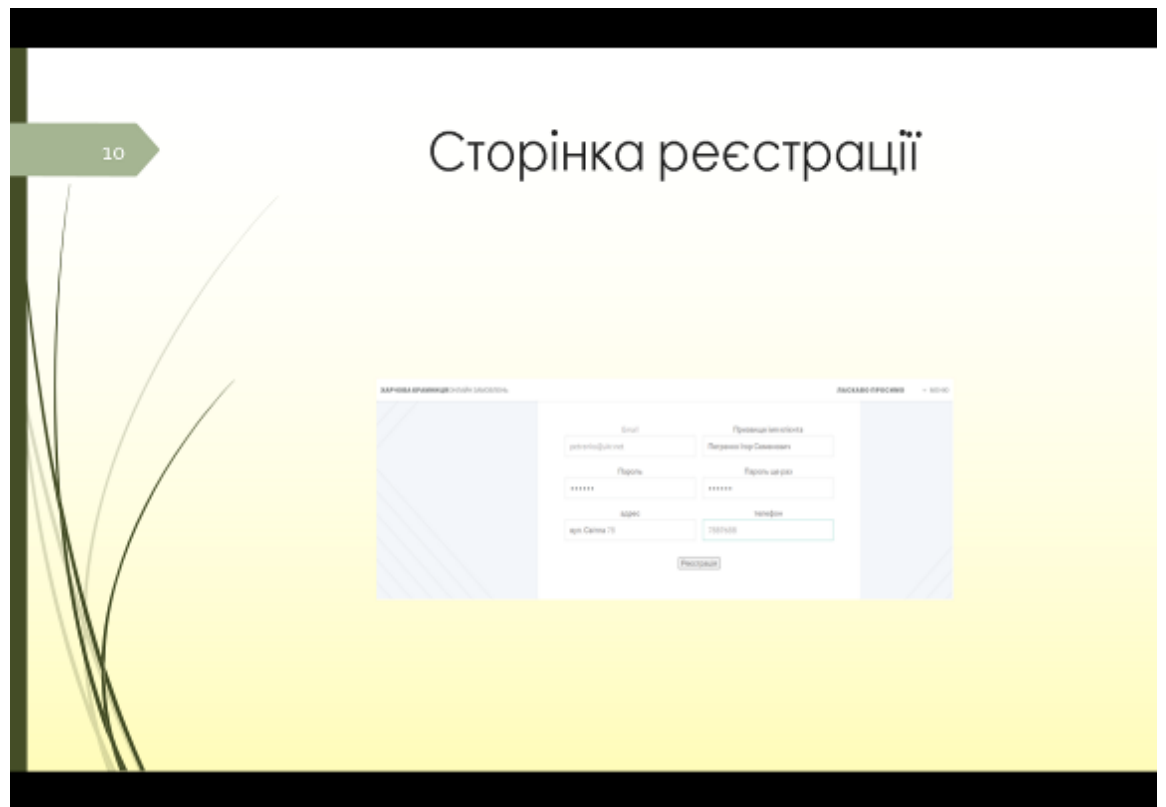


Рисунок Б.1 - Слайд 10

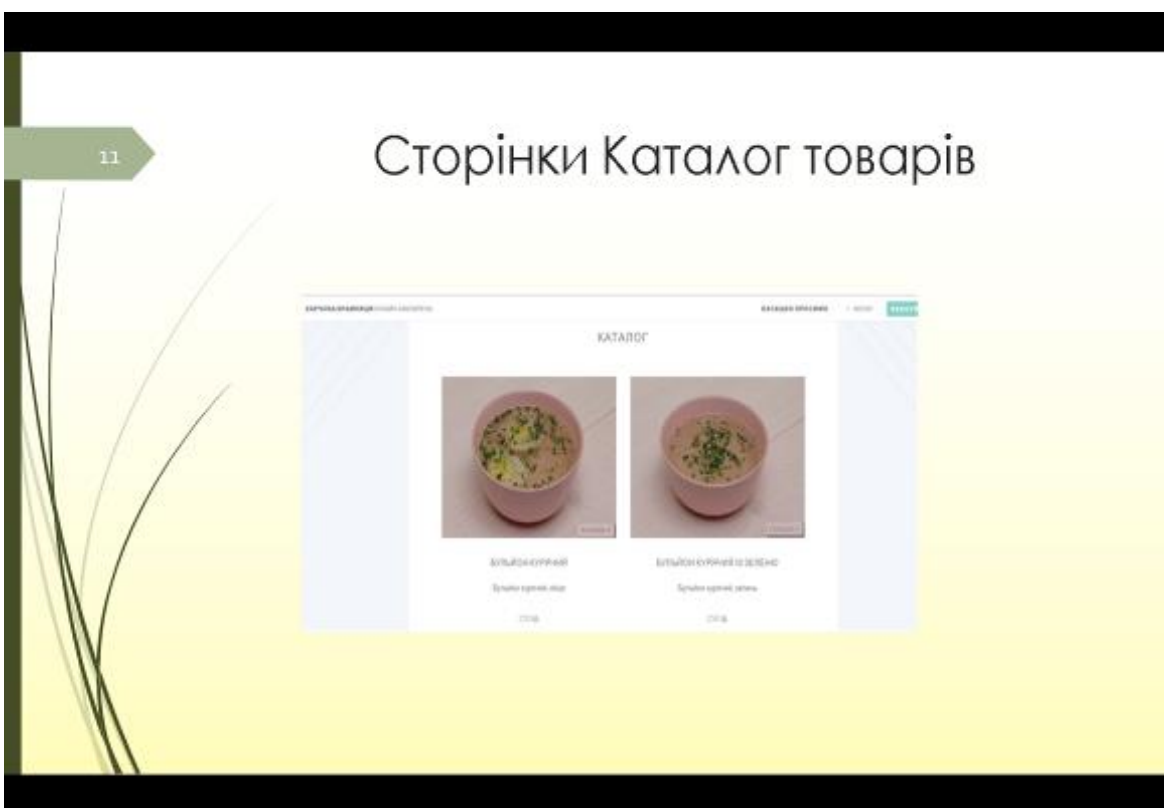


Рисунок Б.1 - Слайд 11

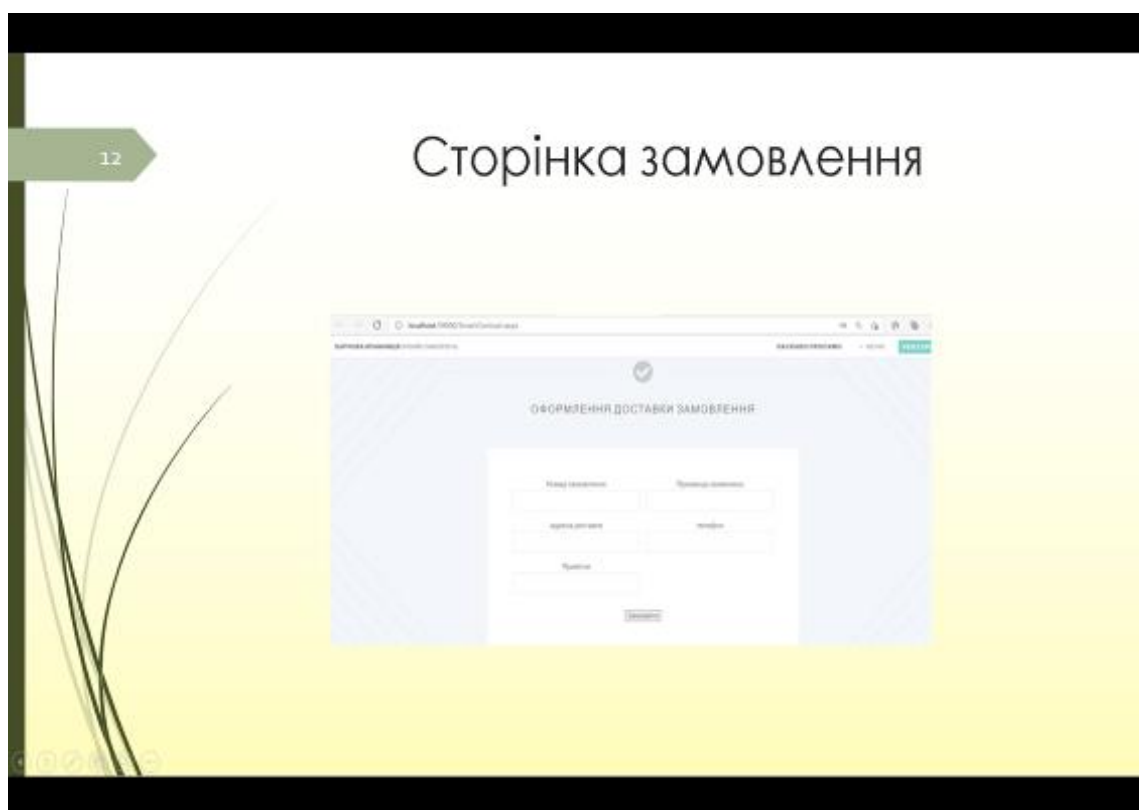


Рисунок Б.1 - Слайд 12

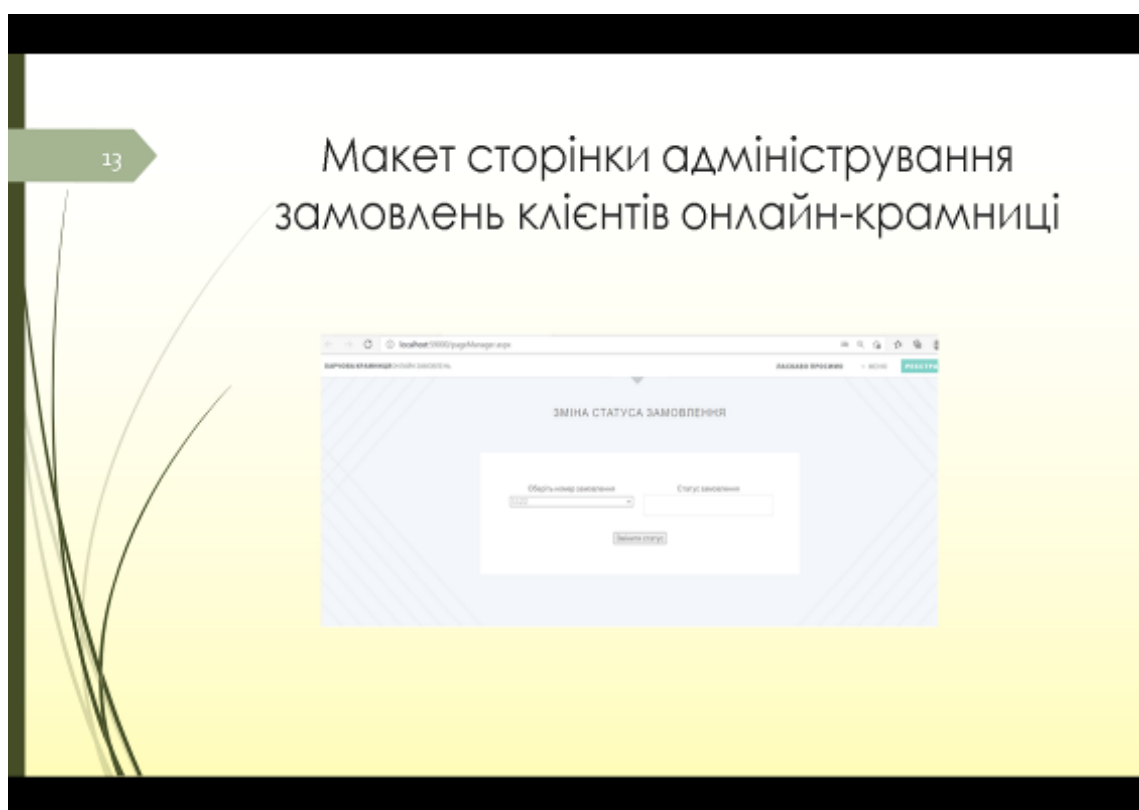


Рисунок Б.1 - Слайд 13

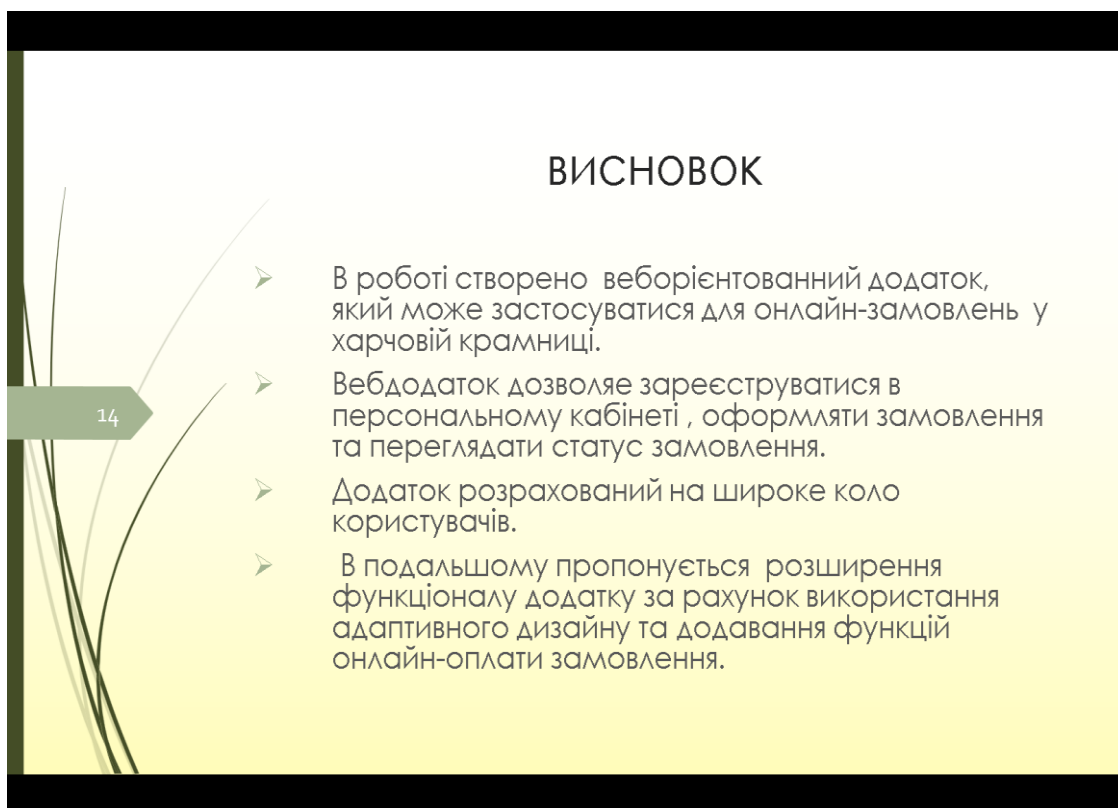


Рисунок Б.1 - Слайд 14

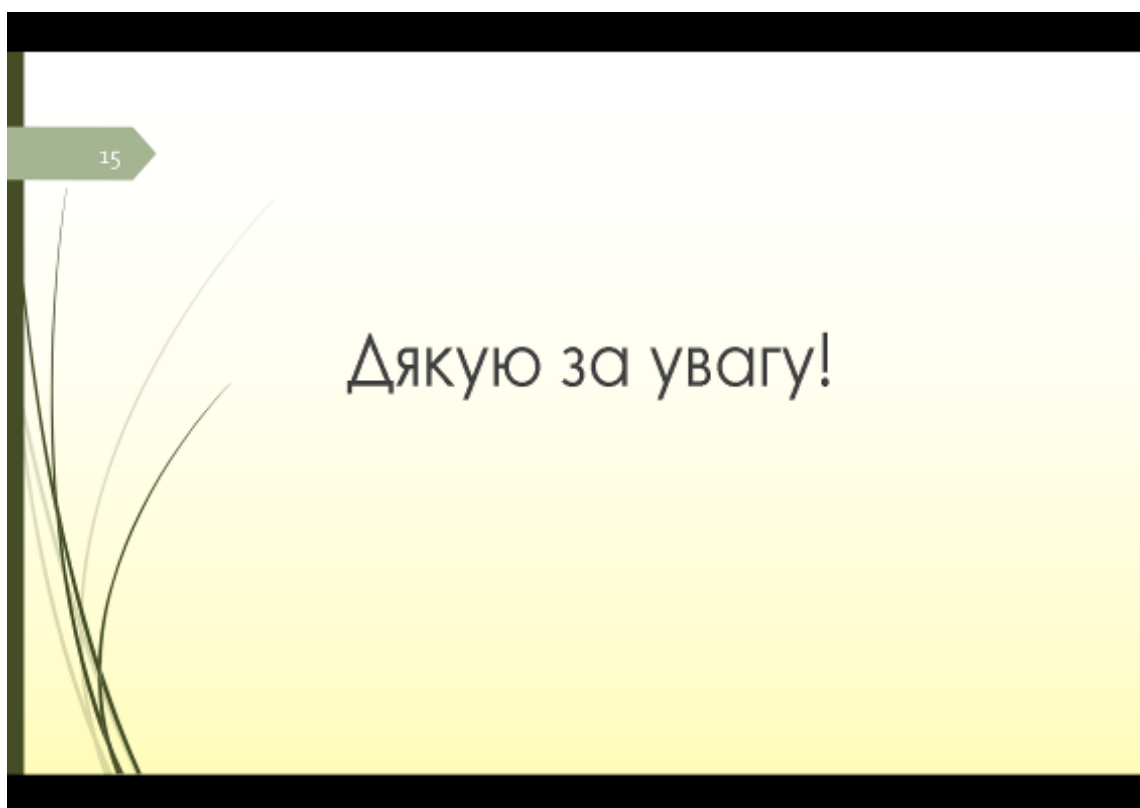


Рисунок Б.1 - Слайд 15