

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток підтримки діяльності магазину продуктів»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-82 Станкевич Даніїл Володимирович

Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою

«__» _____ 2022 р.

Науковий керівник

(підпис)

к.т.н., доц., Федотова Н.А.
(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик

«__» _____ 2022 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Станкевича Данііла Володимировича

1 Тема роботи Web-додаток підтримки діяльності магазину продуктів

керівник роботи Федотова Н.А., к.т.н.,

затверджені наказом по університету від «27» квітня 2022 р. №0301-IV

2 Строк подання студентом роботи «07» червня 2022 р.

3 Вхідні дані до роботи технічне завдання на розробку web-додатку

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку, розробка web-додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання: 01.10.2021

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Оформлення планування робіт	02.12.2022 – 02.12.2022	
2	Оформлення технічного завдання	02.12.2022 – 03.12.2022	
3	Проведення аналізу предметної області	02.12.2022 – 03.12.2022	
4	Проведення проектування web-додатку	01.02.2022 – 01.02.2022	
5	Розробка web-додатку	01.02.2022 – 07.02.2022	
6	Тестування web-додатку	03.04.2022 – 06.04.2022	
7	Оформлення пояснювальної записки	01.05.2022 – 06.06.2022	

Студент

_____ (підпис)

Станкевич Д.В.

Керівник роботи

_____ (підпис)

к.т.н., Федотова Н.А.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки діяльності магазину продуктів».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел із 14 найменувань, 3 додатків. Загальний обсяг пояснювальної записки складає 52 сторінки, у тому числі 39 сторінок основного тексту, 2 сторінки списку використаних джерел, 11 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку підтримки діяльності магазину продуктів.

У першому розділі проведено огляд останніх досліджень за тематикою роботи та проаналізовано аналоги розроблюваного web-додатку, визначено їх переваги та недоліки. Також було поставлено мету й задачі проекту, визначено засоби реалізації.

У другому розділі проведено структурно-функціональне моделювання, визначено варіанти використання web-додатку та спроектовано базу даних. У результаті було змодельовано такі діаграми як: контекстна діаграма IDEF0 та її декомпозиція, діаграма варіантів використання, діаграма автоматів.

У третьому розділі описано процес розробки web-додатку, яка є результатом проектування. Також наведено архітектуру web-додатку..

Ключові слова: WEB-ДОДАТОК, БАЗА ДАНИХ, ДІАГРАМА, PHP, CSS, РОЗРОБКА.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Загальна характеристика предметної області	9
1.2 Аналіз існуючих додатків	9
1.3 Постановка задачі	15
2 Моделювання Web-додатку підтримки діяльності магазину продуктів.....	17
2.1 Моделювання використання Web-додатку	17
2.2 Структурно-функціональне моделювання	18
2.3 Проектування бази даних.....	23
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	26
3.1 Архітектура Web додатку	26
3.2 Програмна реалізація	29
3.3 Використання програмного додатку	34
ВИСНОВОК.....	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	38
ДОДАТОК А	40
ДОДАТОК Б	45
ДОДАТОК В	51

ВСТУП

Сьогодні розвиток технологій є досить важливим питанням у нашому житті. Тому велика кількість ресурсів направлено на їх розвиток. Однією з галузей розвитку є розробка веб-сайтів та додатків. У наш час веб-сайт — це симбіоз інформаційних, програмних, та медійних засобів, що логічно зв'язані між собою. Наразі веб-сайт — це дзеркало яке показує успішність фірми, його можна порівняти з її обличчям. Веб-сайт слугує для виконання таких основних завдань:

- реклама товарів, послуг, що надаються, цікавих ідей. Правильно розроблений веб-сайт набагато швидше приведе клієнта до рішення про необхідність придбання товару, або послуг, що надаються, або ідей, що він поширює на ньому;
- продаж продукції, або послуг, або інформації чи навіть ідей. У сьогоденної людини фізично не вистачає часу для прогулянок по магазинах. Тому можливість, яку надає онлан магизини, а сама придбання товарів і послуг, не встаючи з-за комп'ютера, значно збільшує персональні можливості як клієнта, так і продавця;
- безкоштовне давання послуг або інформації про те про що йдеться на сайті. У дійсності надання послуг або інформації — це один із варіантів привернення уваги відвідувачів до ресурсу, що містить в собі цю інформацію чи послуги для розміщення реклами, в тому випадку коли це простір для розміщення реклами;
- допомога клієнтам.

Наразі, кількість нових веб-сайтів, зростає загрозливими темпами. Тому головне завдання розробки сайту – зробити сайт, що буде максимально привабливим для потенційних користувачів з врахуванням індивідуальних можливостей.

На початку існування Інтернету, веб-сайт міг залучати велику кількість відвідувачів без особливих зусиль або матеріальних витрат цього. Але, коли зростає конкуренція, тоді на перші шпальта виходє функціональність. Тому, немає

різниці, чи створений веб-сайт задля відображення інформації о компанії чи задля бажання заробляти гроші на рекламі, його головна функція завжди – зацікавити аудиторію. Тому правильний підхід до вибору стратегії та виконавців є вирішальним для успіху.

Мета – розробити додаток підтримки діяльності магазину продуктів.

Задачі, які необхідно виконати для створення проекту:

- провести аналіз предметної області
- створити інтерфейс, який максимально відображає необхідну інформацію,
- розробити базу даних, в якій буде зберігатися вся інформація щодо замовлень,
- розмістити базу даних на локальному чи сервері хостинга та підключити її до програми,
- провести тестування,
- скласти всі необхідні документи щодо проекту.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика предметної області

Майже кожна людина хоча б раз замовляла щось до дому. Для цього їй скоріш за все потрібно було обрати товар а потім замовити доставку. Але це не дуже розвинено у сфері продуктів. Для цього необхідно виходити з дому та купувати продукти в магазині. Створити додаток як інструмент, який дозволяє людям купувати будь-які речі, не виходячи з дому це плюси інтернет магазинів.

Сьогодні ці питання можна вирішити більш комфортно, не виходячи з дому, завдяки автоматизованій інформаційній системі, яка буде надавати можливість замовити продукти до дому. Автоматизація процесів матиме якісний вплив і на зручність для користувачів, а отже кількість задоволених людей збільшиться.

1.2 Аналіз існуючих додатків

Сьогодні можна знайти величезну кількість web-додатків для замовлення різних товарів в мережі інтернет. Для аналізу був обран вебсайт «ROZETKA» [1]. На його прикладі розглянути такі задачі - основний функціонал web-додатку та дизайн.

Головна сторінка сайту зображена на рисунку 1.1.

Навігація по сайту наступна – зліва меню з необхідними категоріями товару. Меню не перевантажено зайвою інформацією, лише перелік головних категорій. Користувач легко може обрати потрібний товар з випадаючого списку.

Центральна частина головної сторінки містить інформацію про акції та інші необхідні опції, наприклад, пошук, зміна мови. Дані функції зображені на рисунку 1.2.

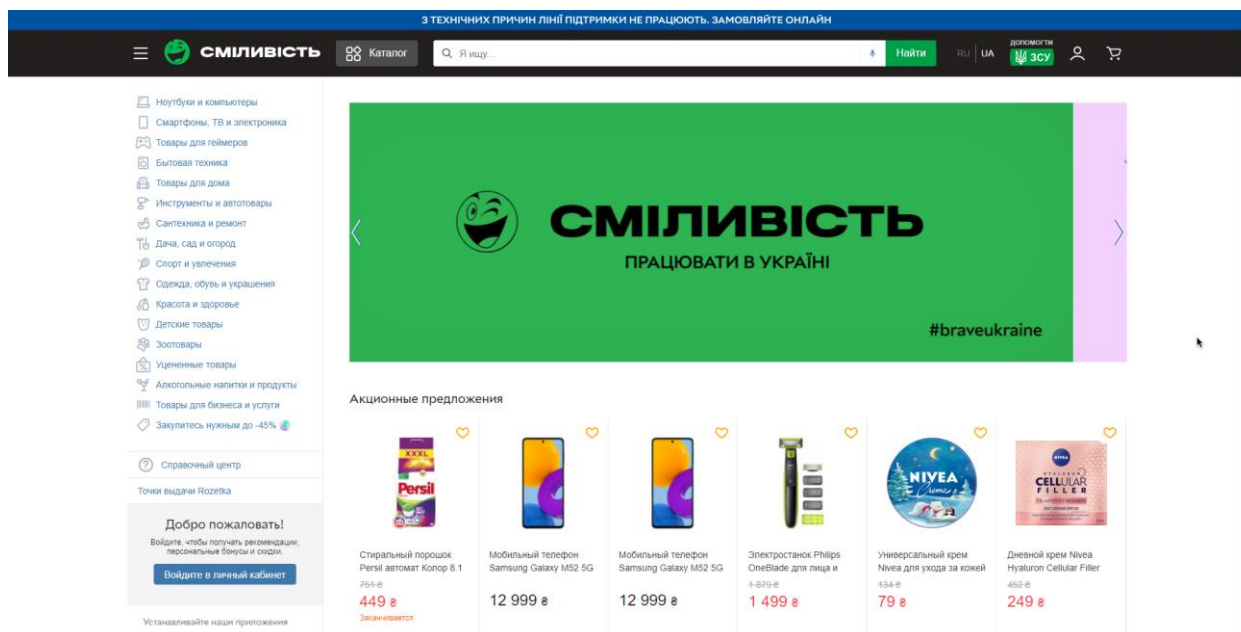


Рисунок 1.1 – Головна сторінка сайту «ROZETKA»

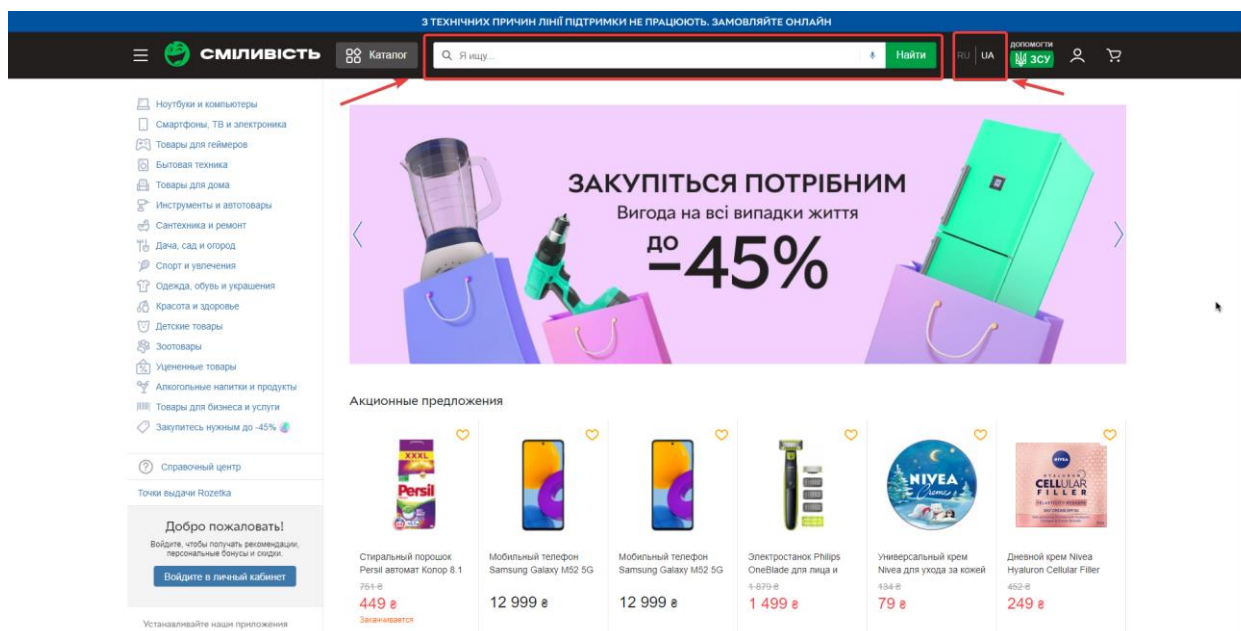


Рисунок 1.2 – Функції пошуку та зміни мови відображення.

Сайт не має футера як такого, він був перенесений під меню, завдяки чого займає менше простору. Це зображено на рисунку 1.3.

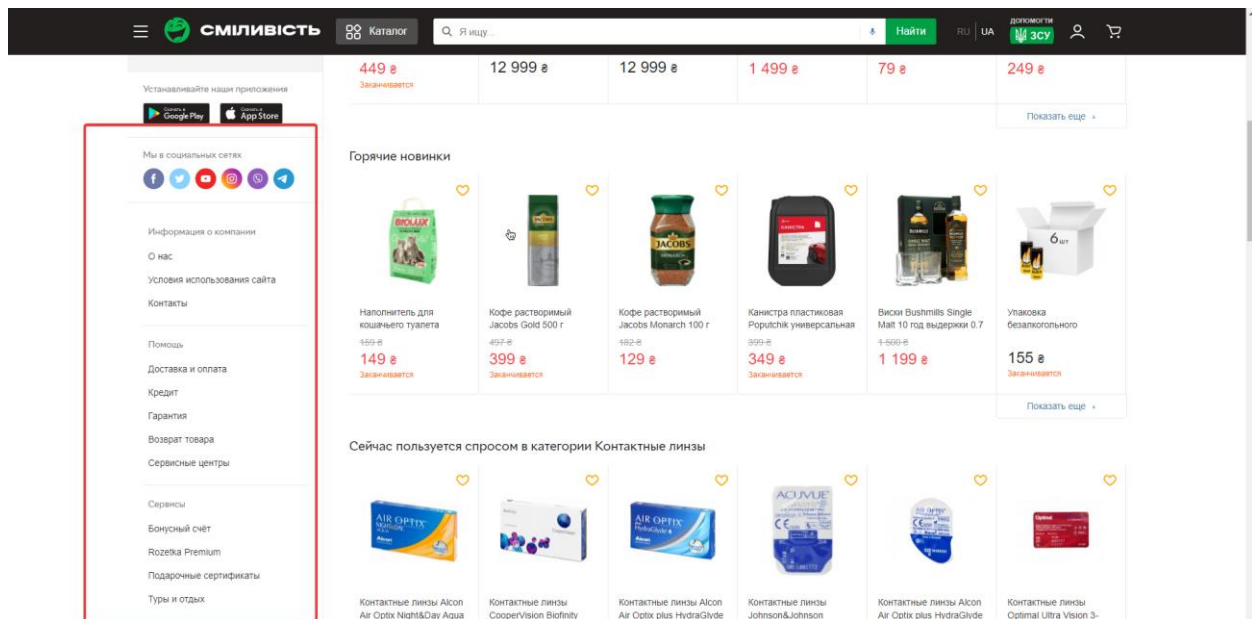


Рисунок 1.3 – Контактна інформація

При перегляді навігаційного меню категорій можна помітити кнопку, що відкриває головне функціональне меню сайту. Даний блок зображений на рисунку 1.4

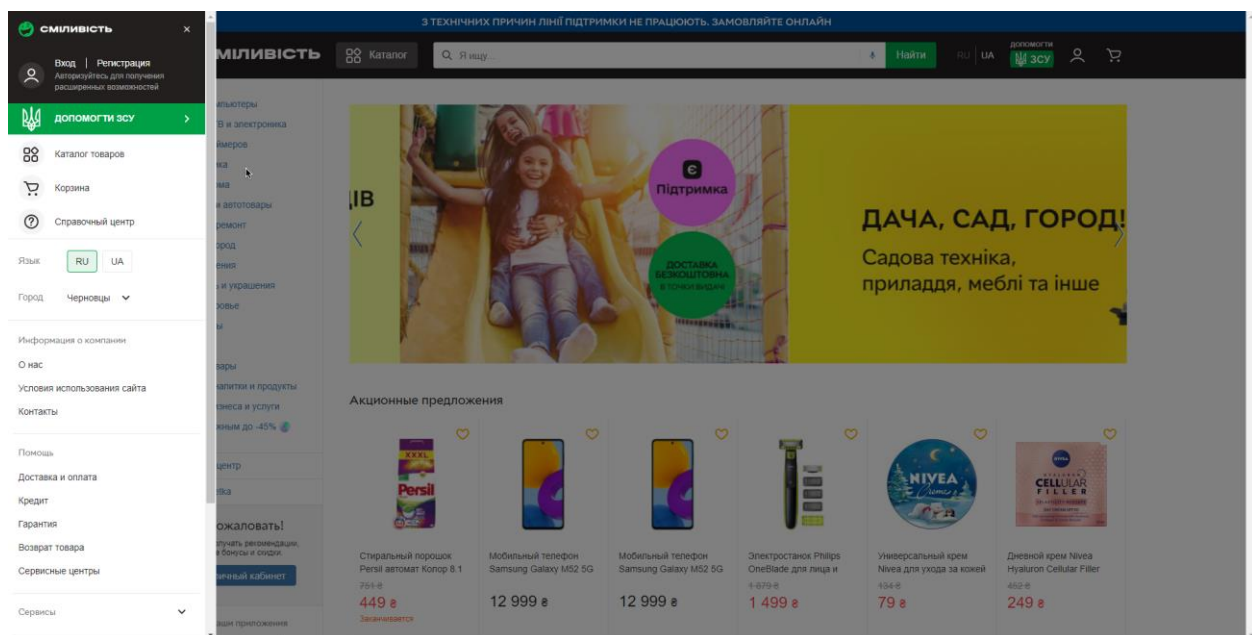


Рисунок 1.4 – Головне функціональне меню сайту

Обираючи товар та клікнувши по ньому користувач потрапляє на сторінку з цим товаром. Приклад зображений на рисунку 1.5.

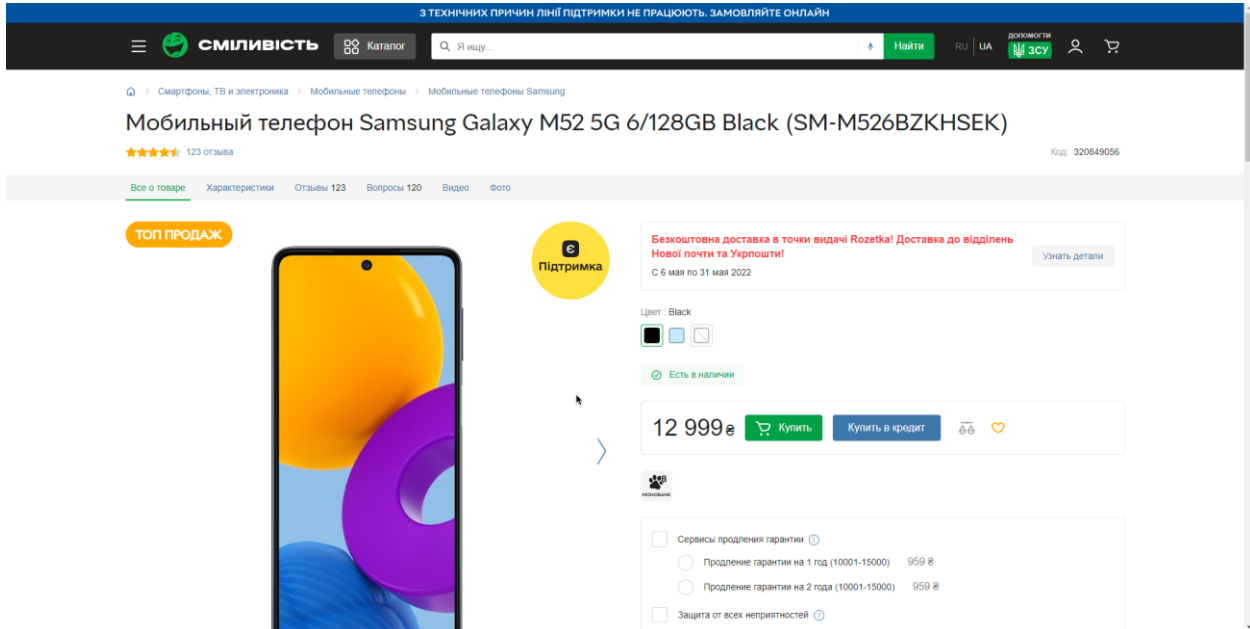


Рисунок 1.5 – Приклад товару

З цієї сторінки користувач має змогу переглянути основну інформацію про товар, його характеристики, обрати різні варіації товару: наприклад колір та додати товар до корзини.

Наступний приклад це сайт «OLX» [2].

Це інше представлення сайту, який на відміну від попереднього має іншу структуру.

Головна сторінка сайту зображена на рисунку 1.6.

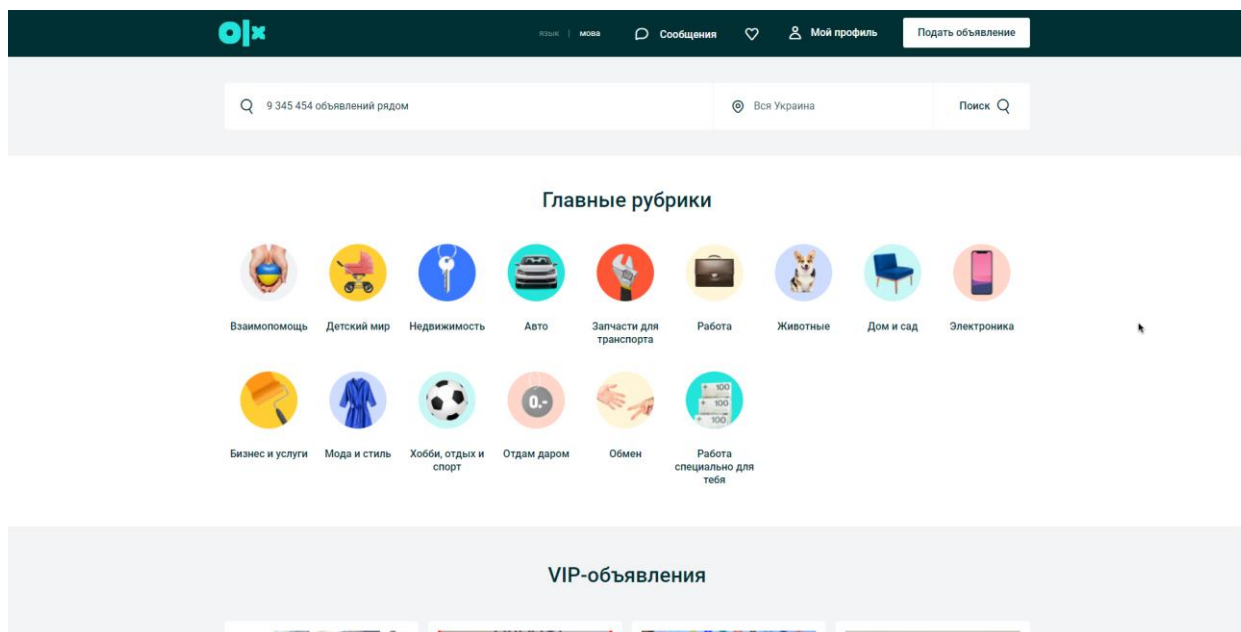


Рисунок 1.6 – Головна сторінка сайту «OLX»

Перша відмінність, це відсутність навігаційного меню. Але дана веб-система на відміну від іншої, на головній сторінці розміщує інформацію про себе.

Також в цьому прикладі є футер, який зображений на рисунку 1.7. Головна сторінка на рисунку 1.8.

Подібно попередньої веб-системі, «OLX» на головній сторінці має поле для пошуку, основні категорії товарів, а також нижче є перелік товарів.

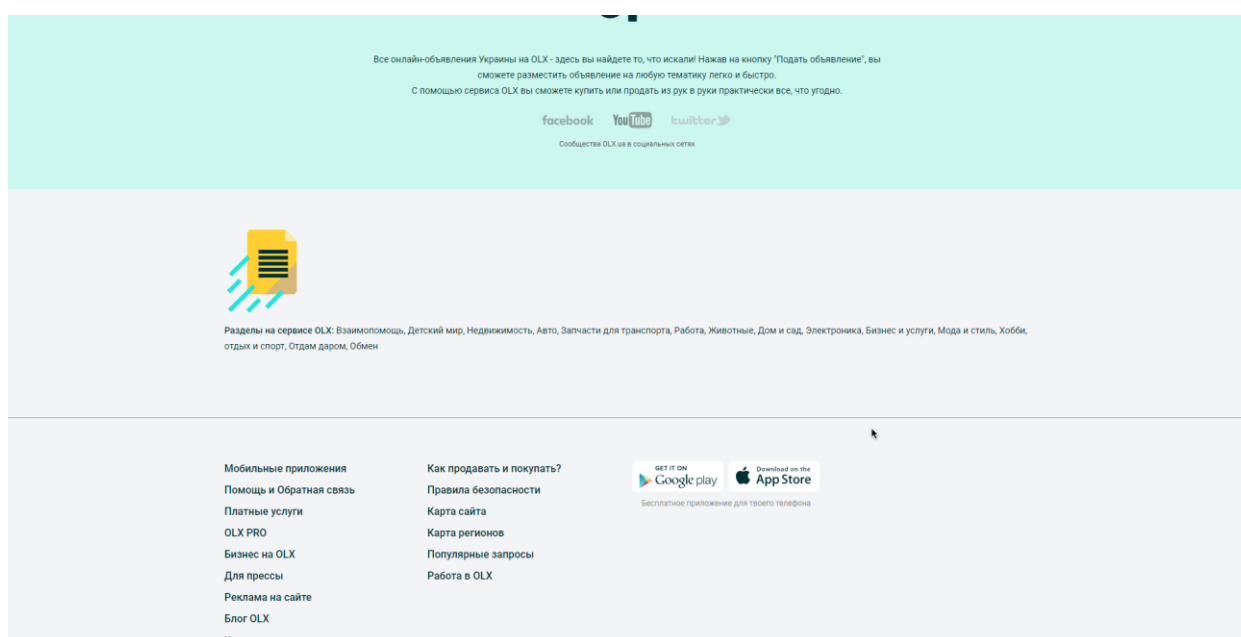


Рисунок 1.7 – Контактна інформація

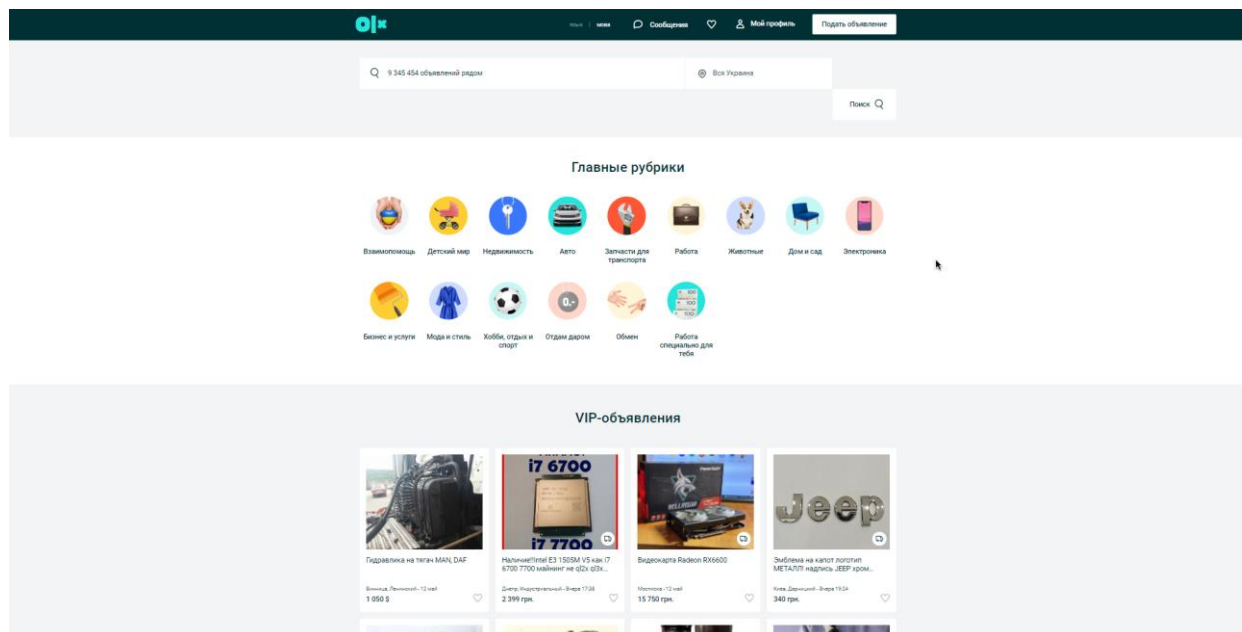


Рисунок 1.8 – Головна сторінка сайту

На сторінці товару користувач бачить основні поля: назва, опис, ціна, локацію звідки буде доставлятися товар. Приклад, так званої «картки товару» зображено на рисунку 1.9.

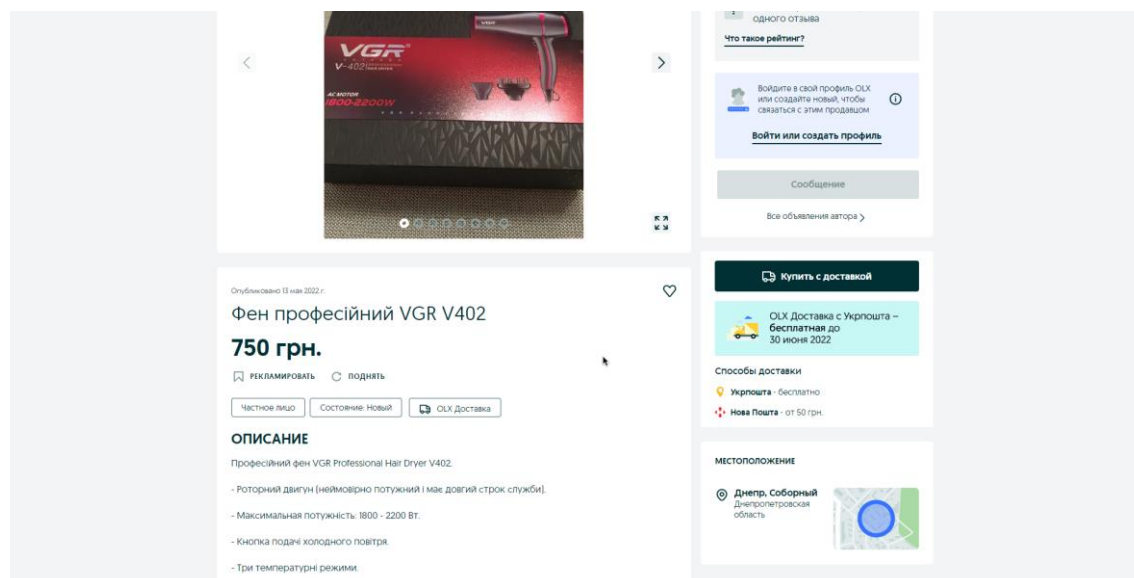


Рисунок 1.9 – Картка товару

Після аналізу головних елементів сайтів-аналогів створимо порівняльну таблицю, в якій відобразити основні переваги та недоліки розглянутих сайтів. Результати аналізу зображені в таблиці 1.1

Таблиця 1.1 – Порівняльний аналіз характеристик веб-сайтів

Характеристика	«ROZETKA»	«OLX»
Можливість роботи на декількох мовах	+	-
Сучасний дизайн	+	-
Навігація сайтом	+	+
Зручність перегляду товару	+	+
Можливість допомоги та консультації	+	+

Проведений аналіз таблиці порівняння дає нам можливість виділити основні переваги та недоліки розглянутих сайтів. За допомогою даної таблиці ми використаємо корисний функціонал сайту та побудуємо свій додаток.

1.3 Постановка задачі

Мета кваліфікаційної роботи є створення веб-додатку для замовлення продуктів. Додаток дозволить замовляти додому і як наслідок це збільшення зацікавленої аудиторії та зацікавлення користувачів у даному напрямку.

Інтерфейс веб-додатку необхідно розробити у стилі, що слугує для ведення бізнесу. Також він має бути сучасний, щоб зацікавити нових користувачів. Він повинен бути інтуїтивно зрозумілий та простий у використанні.

Проаналізувавши основні переваги та недоліки сайтів-аналогів та виконав аналіз предметної області, були визначені головні вимоги до створення веб-додатку:

- Створити корзину та можливість додавання до неї замовлення;
- Виконати елемент розподілу інформації по блокам задля надання наочного представлення наступної навігації;
- Створити базу даних, де буде зберігання інформації та/або оновлення інформації на сайті.

Задля досягнення поставленої мети та вимог потрібно вирішити наступні задачі, які дозволять виконати проект:

- провести аналіз предметної області
- створити інтерфейс, який максимально відображає необхідну інформацію,
- розробити базу даних, в якій буде зберігатися вся інформація щодо замовлень,
- розмістити базу даних на локальному чи сервері хостинга та підключити її до програми,
- провести тестування,
- скласти всі необхідні документи щодо проекту.

Мови, що будуть використовуватися для розробки: HTML, CSS, JAVA, JAVA SCRIPT. Ці мови програмування є універсальними для створення веб-додатків. Вони дозволяють створити інтерфейс та функціонал

База даних буде розроблена з використанням SQL, та інтерфейсу доступу MySQL задля змоги створення потрібної бази даних. Використання цього дасть можливість дуже просто обновляти та експлуатувати додаток.

Повне технічне завдання на виконання робіт міститься в додатку А, планування виконання робіт наведено в додатку Б.

2 МОДЕЛЮВАННЯ WEB-ДОДАТКУ ПІДТРИМКИ ДІЯЛЬНОСТІ МАГАЗИНУ ПРОДУКТІВ

2.1 Моделювання використання Web-додатку

UML – це аббревіатура, що розшифровується як Unified Modeling Language, є способом візуального представлення архітектури, проектування та реалізації складних програмних систем. Коли йде процес написання коду, у програмі є тисячі рядків, і важко відстежити взаємозв'язки та ієрархії в програмній системі. Діаграми UML поділяють цю програмну систему на компоненти та підкомпоненти [15].

UML — це стандартизована мова моделювання, яку можна використовувати в різних мовах програмування та в різних процесах розробки, тому більшість розробників програмного забезпечення зрозуміють її та зможуть застосувати у своїй роботі. UML:

- надає вказівки щодо порядку діяльності команди;
- визначає, які процеси слід розробляти;
- керує завданнями окремих розробників та команди в цілому;
- пропонує критерії для моніторингу та вимірювання продуктів та діяльності проекту.

Існує декілька варіантів діаграм UML, і кожна призначена з для певної мети, незважаючи на те, вона розробляється до впровадження чи потім (як одна з частин документації).

Дві найбільш розповсюджені типи, які включають всі інші категорії, - це поведінкова діаграма UML та структурна діаграма UML. Згідно з назви цих діаграм, деякі з цих UML аналізують та зображують структуру процесу чи системи, в той час коли інші діаграми характеризують поведінку системи, її істот або її компонентів [5].

Діаграма випадків використання відноситься до поведінкових діаграм [15]. Модель використання описує функціональні вимоги системи з точки зору випадків використання. Це модель передбачуваної функціональності системи (випадки використання) та її середовища (актори). Приклади використання дозволяють співвіднести те, що потрібно від системи, з тим, як система задовольняє ці потреби.

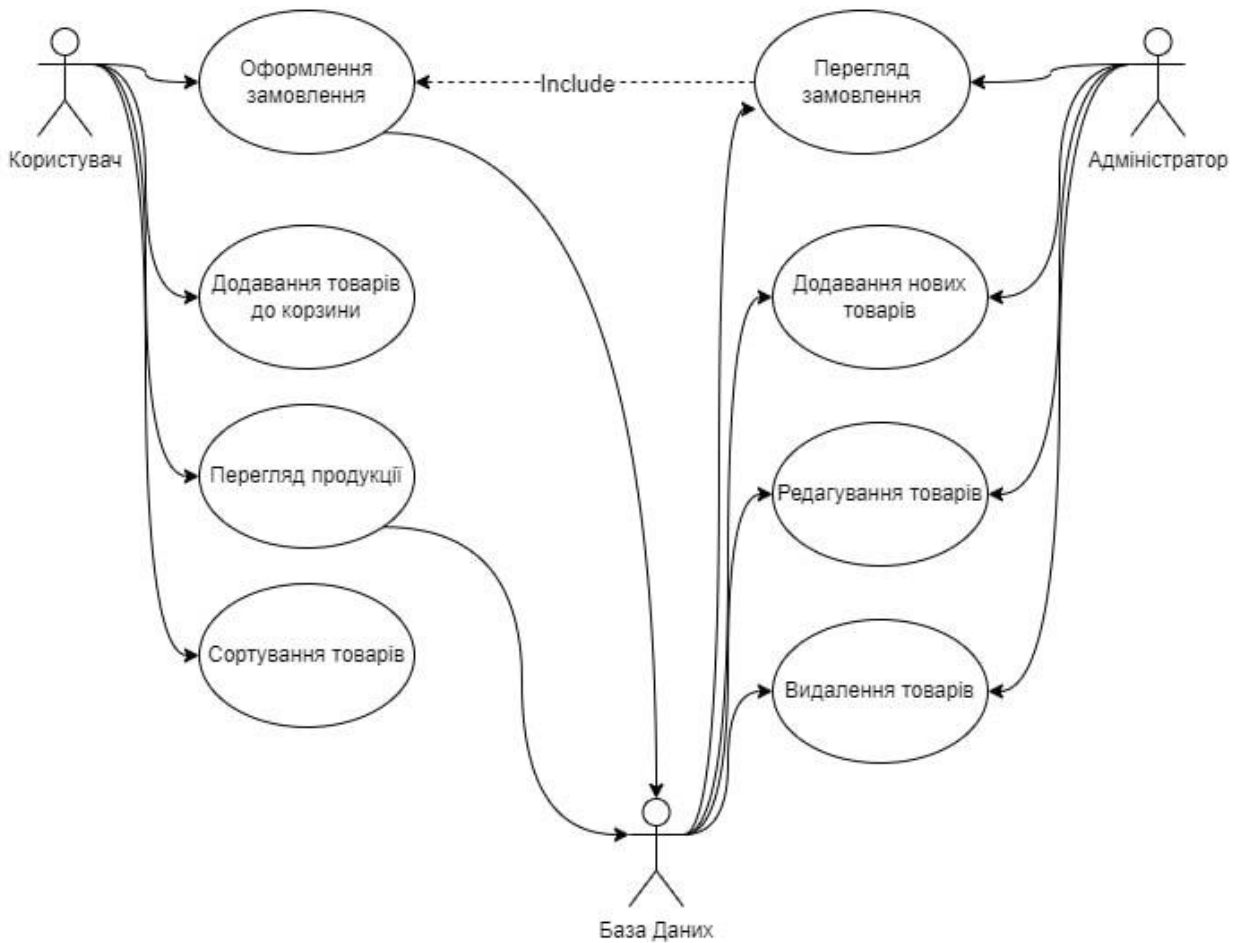


Рисунок 2.1 – Діаграма варіантів використання

2.2 Структурно-функціональне моделювання

Використання опису системи за допомогою методології IDEF0 має класичну назву - функціональна модель. Функціональна модель слугує для опису існуючих

бізнес-процесів, в якому використовуються як природня, так і графічна мови. Для передачі інформації про конкретну систему джерелом графічної мови є сама методологія IDEF0. Дана методологія при описі функціонального аспекту інформаційної системи конкурує з методами, орієнтованими на потоки даних (DFD) [6]. Різниця між ними в том, що IDEF0 дозволяє:

- описувати будь-які системи які потрібні користувачу, а не тільки інформаційні (DFD призначена для опису програмного забезпечення);
- створювати опис системи та її зовнішнього оточення до визначення остаточних вимог до неї.

Іншими словами, за допомогою цієї методології можна поступово вибудовувати і аналізувати систему навіть тоді, коли важко ще уявити її втілення.

Тому ми застосовуємо методології IDEF0 як пристрій для аналізу функцій існуючих систем і вироблення рішень щодо їх поліпшення. Функціональне моделювання Web-додатку наведено на рис. 2.2.

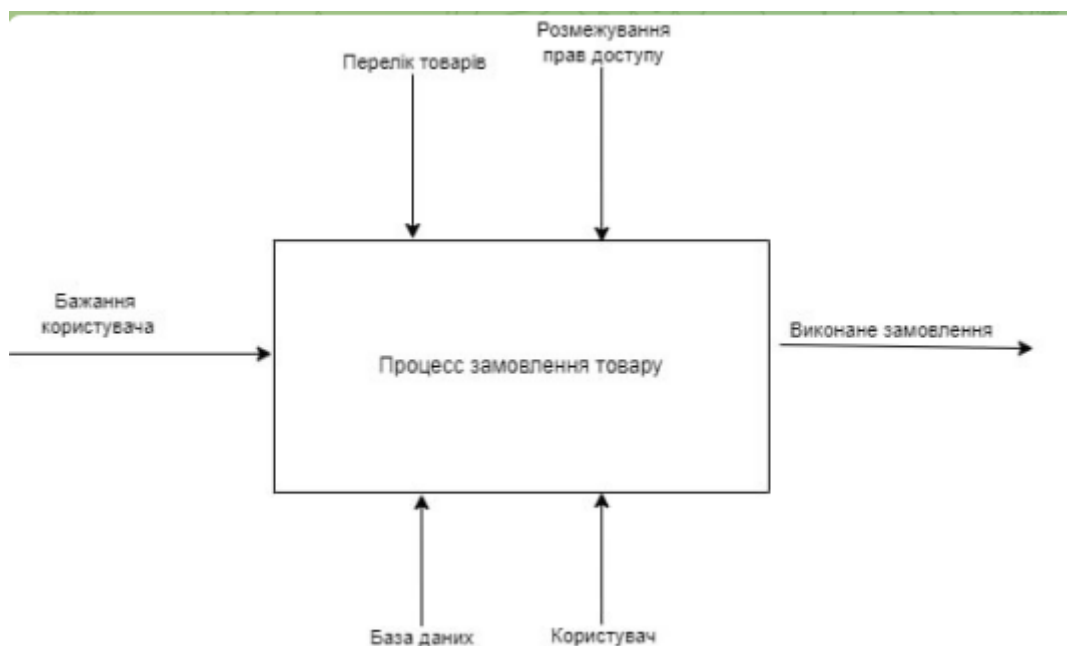


Рисунок 2.2 – Функціональна діаграма (IDEF0)

Діяльність будь-якого підприємства можна зобразити як циклічна зміна стану інтелектуальних та фізичних об'єктів, що мають на пряму відносяться до

підприємства, таких як співробітники, ресурси виробництва, виготовлені товари, ідеї, матеріальні ресурси і т.д. Для найбільш успішного управління цим процесом, кожна зміна будь-якого об'єкта має мати своє відображення у вигляді документації.

Стандарт IDEF1 був розроблений як засіб для розгляду і вивчення взаємозв'язків між потоками інформації в межах комерційної діяльності підприємства.

При побудові інформаційної моделі особа, що проектує завжди працює з двома головними областями, кожній з яких призначена велика кількість характерних об'єктів [11]. Де першою є реальний світ, або ж сукупність фізичних і інтелектуальних об'єктів, таких, як люди, місця, речі, ідеї і т.д., а також всі властивості цих об'єктів і залежності між ними [10]. Другий же є інформаційна область. Вона включає в себе існуючі інформаційні відображення об'єктів першої області і їх властивостей. Інформаційне відображення, по суті, не є об'єктом реального світу, проте зміна його, як правило, є наслідком деякого зміни відповідного йому об'єкта реального світу. Декомпозиції оформлення замовлення представлене на рисунку 2.3.

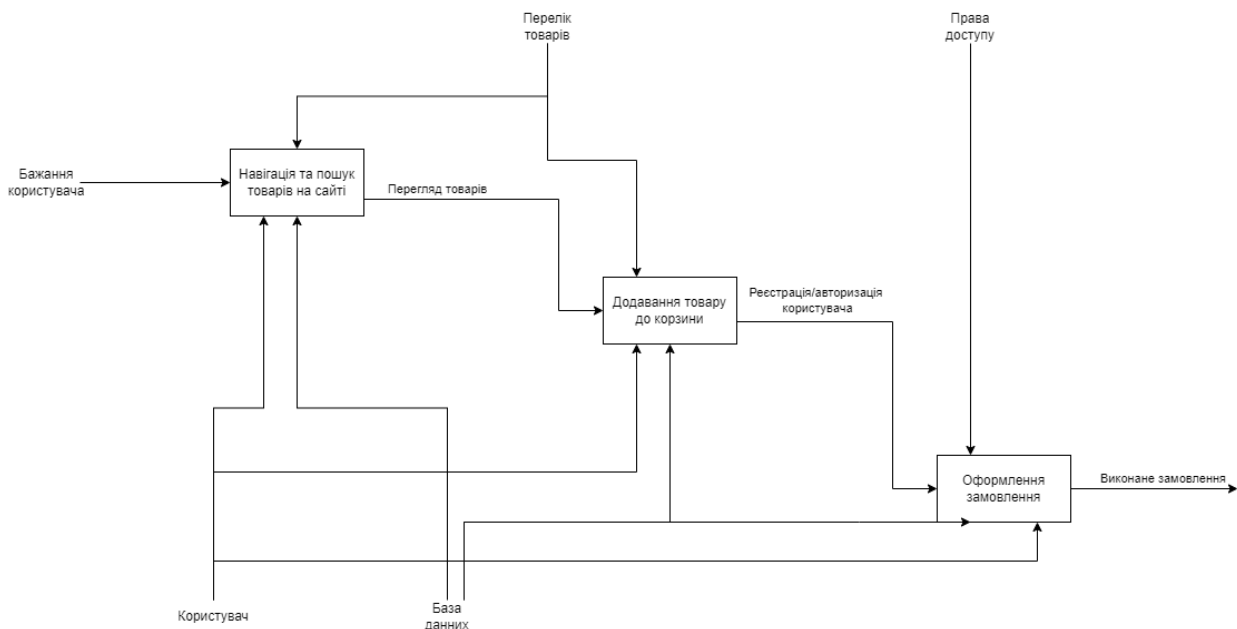


Рисунок 2.3 – Діаграма декомпозиції оформлення замовлення(IDEF1)

Діаграми станів зображують стани та переходи. Стан відноситься до різних комбінацій інформації, яку може містити об'єкт, і ця діаграма UML може візуалізувати всі можливі стани та спосіб переходу об'єкта з одного стану в наступний. Вона бере назву, станів автомата, оскільки діаграма, по суті, є машиною, яка описує кілька станів об'єкта та те, як вони змінюється на основі внутрішніх та зовнішніх подій[7, 8].

Діаграма автоматів Web-додатку наведена на рисунку 2.4.

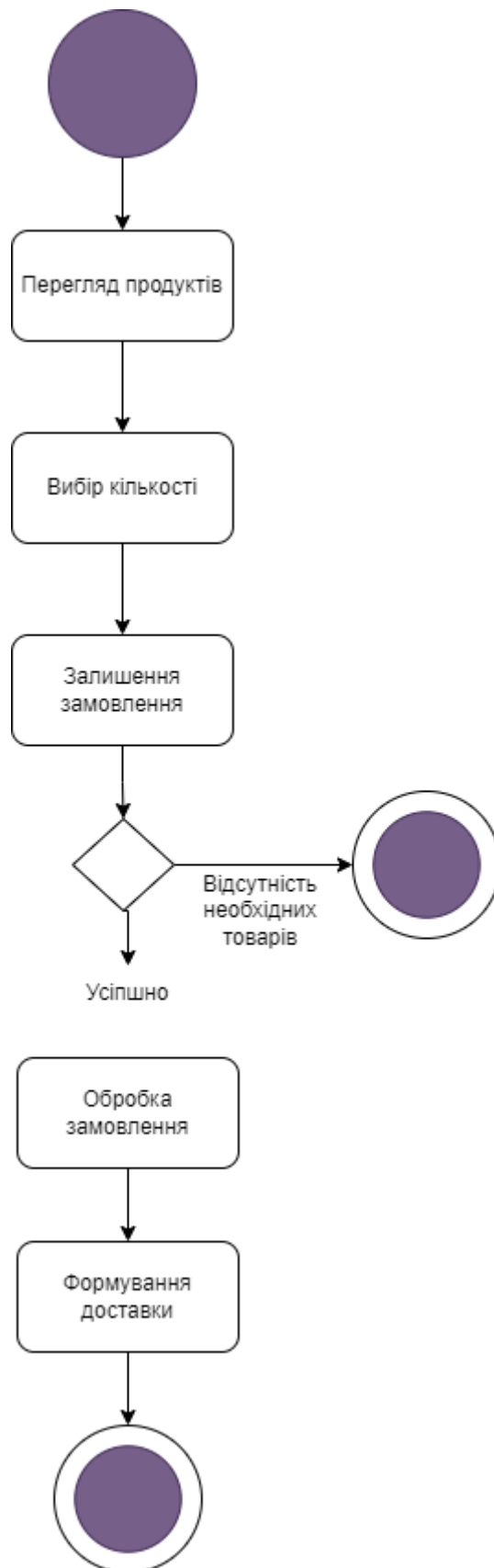


Рисунок 2.4 – Діаграма автоматів

2.3 Проектування бази даних

Під час виконання дипломного проектування була розроблена база даних, яка містить 5 таблиць: користувачі – для збереження інформації про користувачів, категорії товарів, замовлень – основна інформація про замовлення користувачів, деталі замовлення – детальна інформація про замовлення користувача.

Описання полів таблиць бази даних наведено в табл. 2.1 – 2.5.

Таблиця 2.1 – Описання таблиці “Users”

Ім'я поля сутності	Тип даних	Інформація, що зберігається
ID	INT	ID користувача
Name	Varchar(20)	Ім'я користувача
Password	Varchar(40)	Пароль користувача
Address	Varchar(40)	Адреса

Таблиця 2.2 – Описання таблиці “Categories”

Ім'я поля сутності	Тип даних	Інформація, що зберігається
ID	INT	ID категорії
Name	Varchar(20)	Ім'я категорії
DeSCRIPTION	Varchar(40)	Опис категорії

Таблиця 2.3 – Описання таблиці “Orders”

Ім'я поля сутності	Тип даних	Інформація, що зберігається
ID	INT	ID замовлення
Date	Date	Дата замовлення
Total price	Float	Сумма замовлення
User ID	INT	ID користувач, що замовлял

Таблиця 2.4 – Описання таблиці “Products”

Ім'я поля сутності	Тип даних	Інформація, що зберігається
ID	INT	ID товару
Name	Varchar(20)	Ім'я товару
Price	Float	Вартість товару
DeSCRIPTION	Varchar(40)	Опис товару
Category ID	INT	Номер категорії до якої належить товар

Таблиця 2.5 – Описання таблиці “Order details”

Ім'я поля сутності	Тип даних	Інформація, що зберігається
ID	INT	ID запису детальної інформації про замовлення
Product name	Varchar(20)	Ім'я товару
Product price	Float	Вартість товару
Product ID	Varchar(40)	Номер товару
Order ID	INT	Номер замовлення

Логічна модель бази даних має вигляд (рис. 2.5):

База даних має такі реляційні зв'язки:

- Order_details.Product_ID → Products.ID (1 – 1)
- Order_details.Order_ID → Orders.ID (∞ - 1). Декілька записів з таблиці

Order_details можуть посилатися на ID одне і те саме замовлення.

- Orders.User_ID → Users.ID (∞ - 1). Декілька записів з таблиці Orders можуть посилатися на ID одного і того самого користувача.

- Products.Category_ID → Categories.ID (∞ - 1). Декілька товарів можуть посилатися на одну і ту саму категорію.

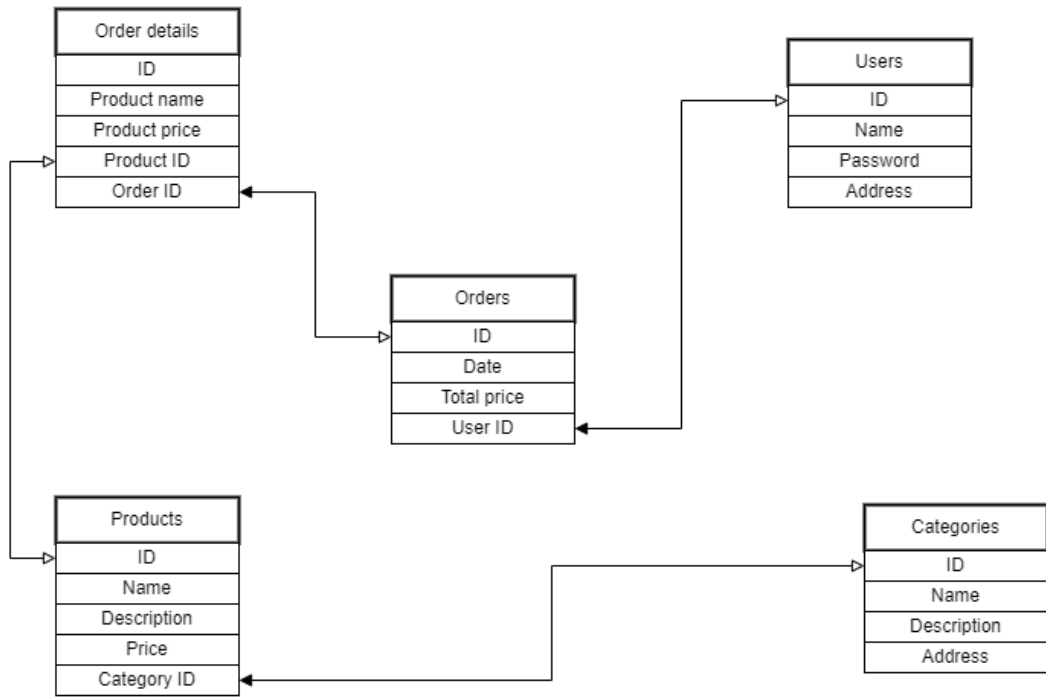


Рисунок 2.5 – Логічна модель бази даних

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Архітектура Web додатку

Проектування архітектури Web-додатку це перший крок проектування. Це робиться за допомогою створення діаграми HLD (High Level Design). Вона показує та описує компоненти, які застосовуються під час розробки програмного продукту. Це концептуальне представлення або робочий план додатку, щоб розроблюється.

Web-додаток був розроблений згідно з технічним завданням. Додаток має базові функції інтернет магазину та відповідає сучасним критеріям. Цей додаток є конкурентно здатним з іншими магазинами, що зосереджуються на доставці їжі та/або продуктів. Діаграма високого рівня представлена на рис. 3.1

Головна ціль веб-сайту – це надання користувачеві можливості зроби запит на доставку бажаного продукту до дому. Як і було зазначено в технічному завданні сайт був розроблений використовуючи наступні мов програмування: MySQL, JAVA, JAVASCRIPT, HTML та CSS.

Інтерфейс має сучасний дизайн та є інтуїтивно зрозумілим для будь-якого користувача який вміє користуватися персональним комп'ютером. Основні елементи, що відповідають за навігаю представленні у вигляді кнопок, що відповідають за певну дію. У процесі реалізації були сторенні такі розділи:

- Головна сторінка – сторінка-каталог товарів, яка надає користувачу змогу ознайомитися с доступними товарами для замовлення.
- Сторінка замовлень – сторінка на якій користувач має змогу продивитися свої існуючі замовлення.

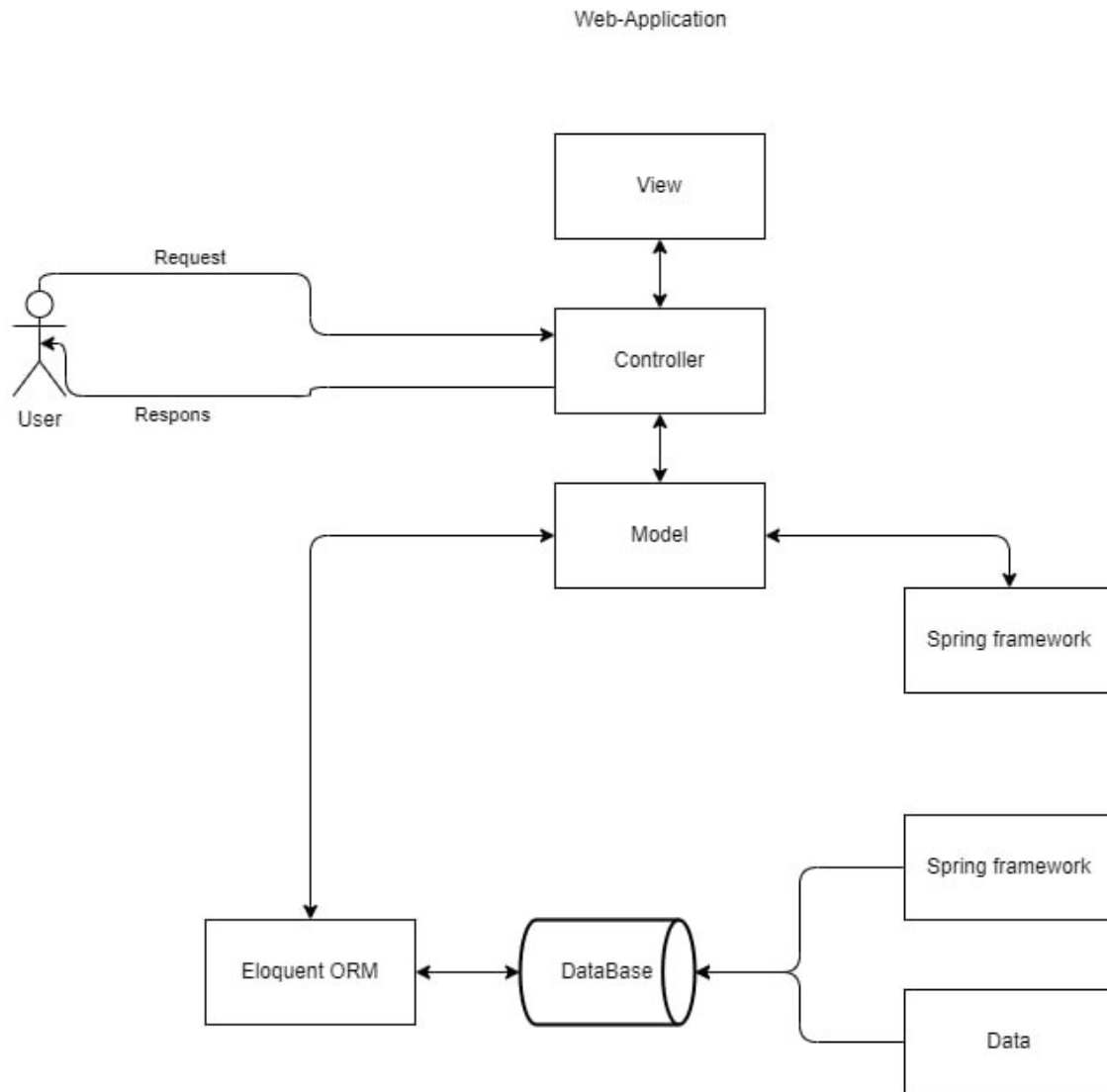


Рисунок 3.1 - Діаграма високого рівня

- Сторінка підтвердження – сторінка, на якій користувач бачить обранні товари та поля для введення даних для доставки.
- Сторінка авторизації – сторінка, що відповідає за авторизації для перегляду зроблених замовлень.
- Сторінка успішного замовлення – сторінка, що повідомляє користувача, що його замовлення було успішним і передано на опрацювання.

На рисунку 3.2 зображено макет веб-сайту

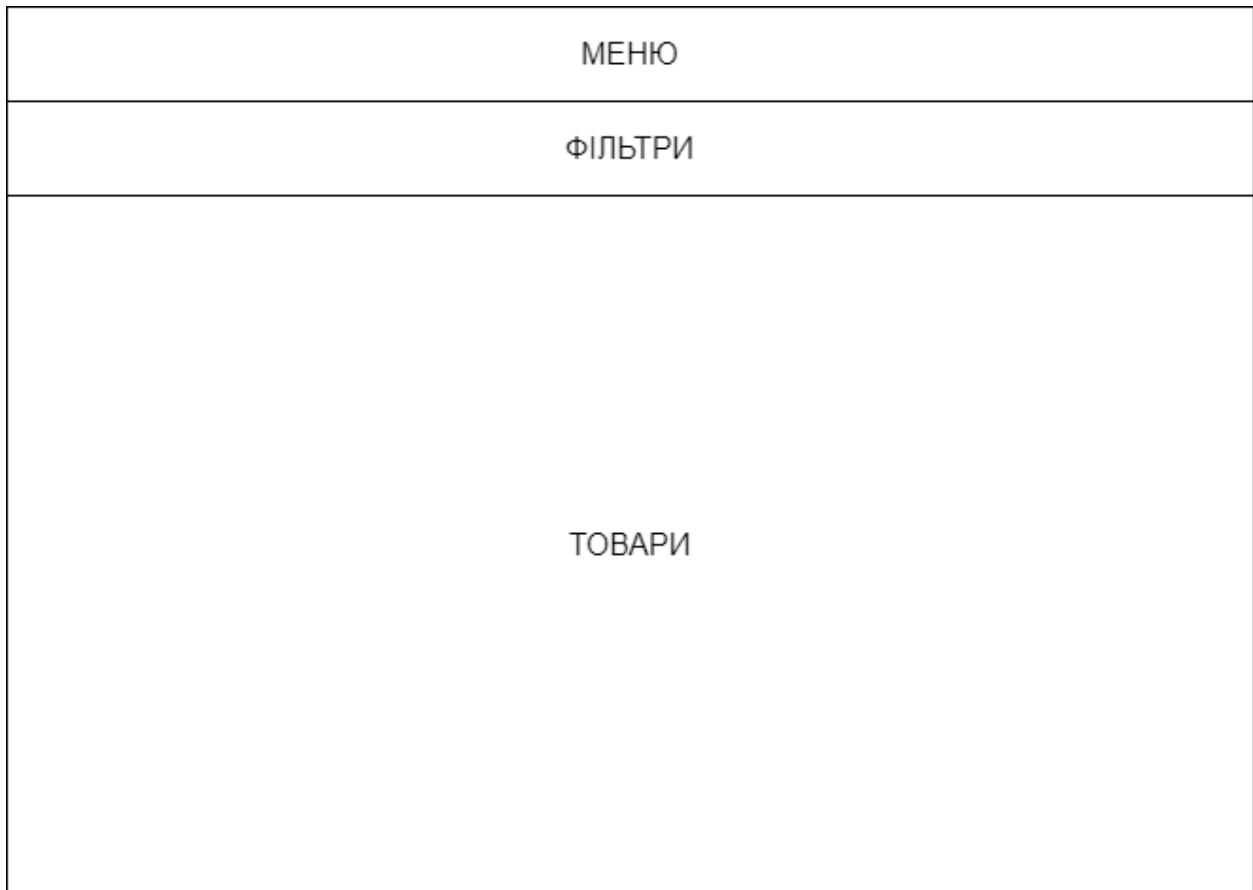


Рисунок 3.2 – Макет Web-додатку

Ссилаючись на макет та діаграму високого рівня, що зображені на рисунках 3.1-3.2, а також технічне завдання була розроблена головна сторінка, яка в одночас є вітриною. На цій сторінці користувач може знайти бажаний товар, відфільтрувати товар за категоріями, додати його до корзини. Також користувач має можливість з сторінки-вітрини побачити корзину товарів та потрапити на сторінку з його замовленнями.

Головна сторінка проілюстрованна на рисунку 3.3.

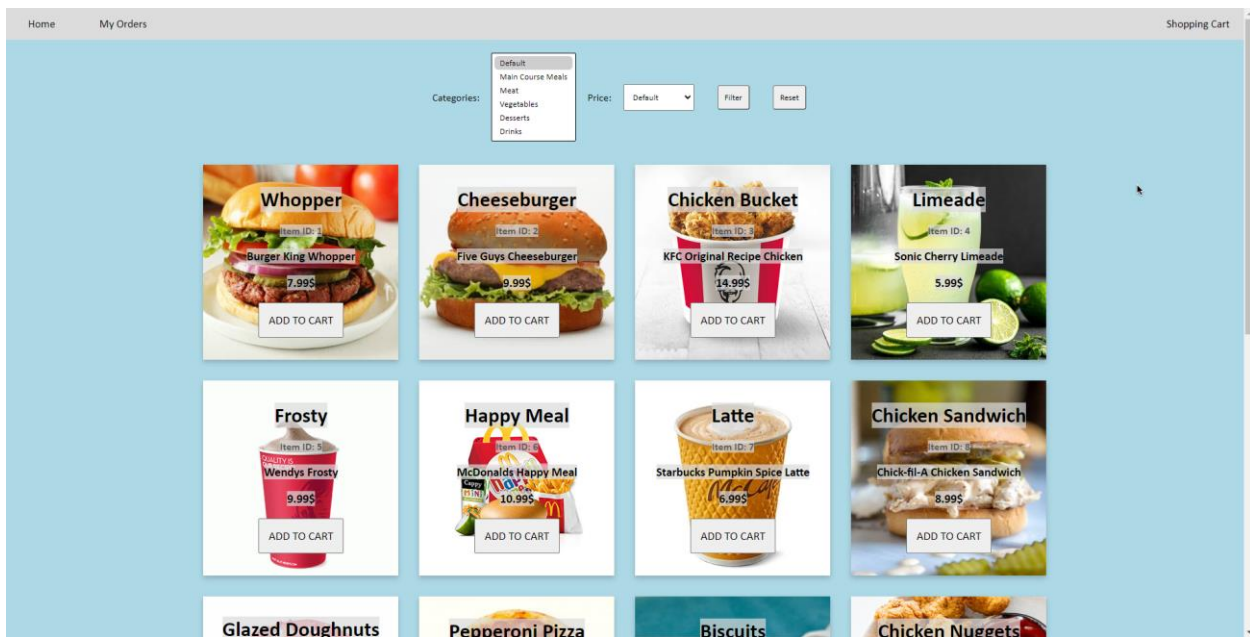


Рисунок 3.3 – Головна сторінка Web-додатку

3.2 Програмна реалізація

Навігаційне меню, за допомогою якого користувач переміщується по сайту було створенн за допомогою CSS (рис. 3.4). Меню закріпленне вгорі екрану та розташована на всіх сторінках. Це меню просте та зрозуміле для будь-якого користувача. Дизайн цього меню є лаконічними та сучасним, виконане у світлих кольорах. Кожна кнопка, що є в меню веде користувача на бажану ним сторінку.

Засобами CSS було створене головне навігаційне меню (рис. 3.4). Розташування менюбуло вирішено робити у лівому блоці сторінки та зафіксувати на всіх сторінках. Це дозволить інтуїтивно та просто використовувати його. Дизайн вибрано лаконічним, з переважним використанням світлих кольорів. Кожен пункт меню повинен мати посилання на відповідну сторінку.



Рисунок 3.4 – Навігаційне меню Web-додатку

На головній сторінці, що є в одночас вітриної користувач має можливість переглянути доступні для замовлення товари. Також має можливість відфільтрувати усі товари, за категорією яку він бажає бачити та зробити сортування за ціною. Для того щоб скинути всі фільтр наявна кнопка, що відповідає за цей функціонал. Головним функціоналом цієї сторінки є додавання товару до корзини. Це можна зробити клікнувши по кнопці «додати до корзини». В HTML сторінці сайту був створений порядок цих елементів на сторінці. Рендерінг товарів робиться динамічно. Бекенд перевіряє які товари необхідно відобразити при певних фільтрах і віддає ці інформацію на фронтенд, після цього товари відображаються на екрані. Бекенд логіка була створена за допомогою мови програмування “JAVA”. За зовнішній вигляд відповідає CSS файл, що привязаний до цієї сторінки.

Описані вище елементи можна переглянути на рисунку 3.5 .

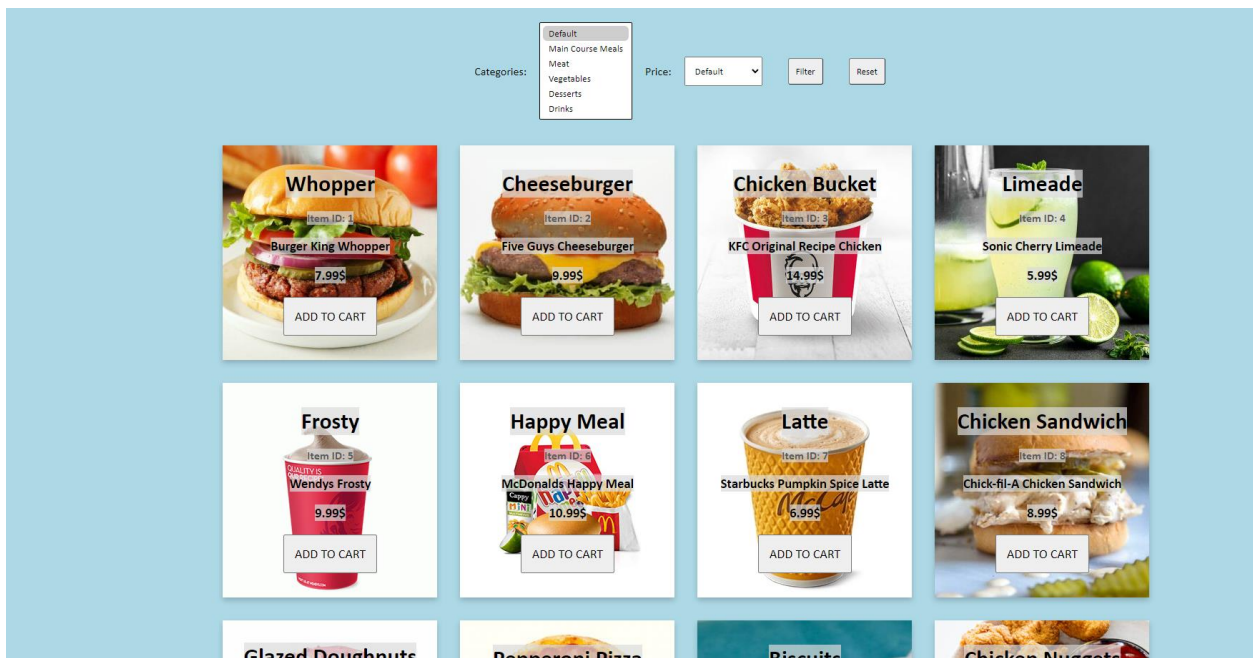


Рисунок 3.5 – Основні елементи головної сторінки

Сформувавши корзини має можливість переглянути її та видалити товари, якщо необхідно. Формування корзини було реалізовано за допомогою мови “JAVA”. Зовнішній вигляд за допомогою HTML, JAVASCRIPT, CSS. Корзина зображена на рисунку 3.6.

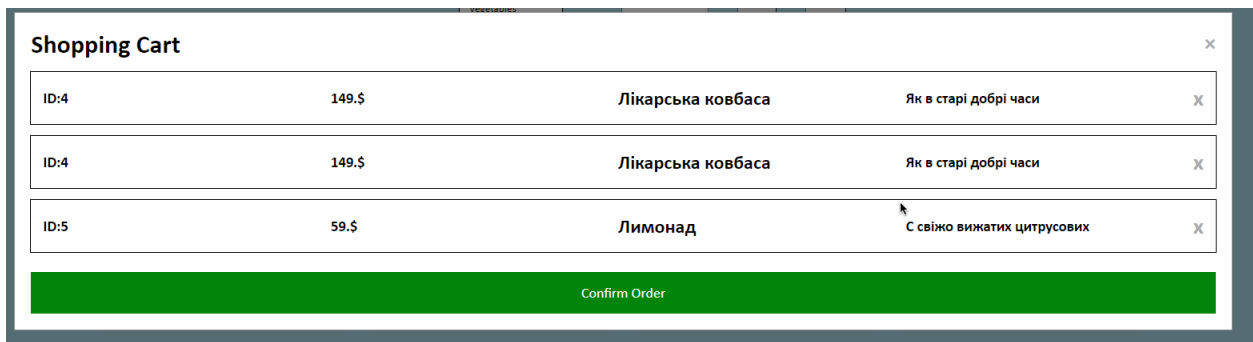


Рисунок 3.6 – Корзина

Далі була розблена сторінка підтвердження замовлення. На цій сторінці користувач бачить перед собою обрані товари, що були перенесені з корзини, а також поля для реєстрації. Якщо користувач з таким ім'ям вже зареєстрований в системі, то новий користувач не буде створений, а виконається вхід у вже існуючий акаунт.

Поля для вводу були створенні за допомогою HTML, стиль був доданий за допомогою CSS. Дані, які передаються до сервера передаються за допомогою мови програмування "JAVA". Вигляд сторінки зображений на рисунку 3.7.

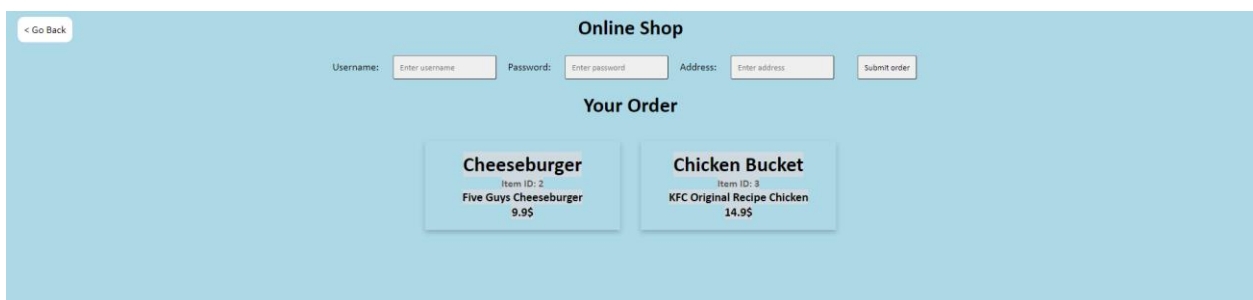


Рисунок 3.7 – Сторінка підтвердження замовлення

Після того як користувач натисне кнопку «Замовити» дані віправляються до бази даних, а користувач в свою чергу бачить сторінку, з текстом про успішне замовлення.

Для того щоб потрапити на сторінку з замовленнями користувач має пройти авторизацію. Сторінка авторизації зображена на рисунку 3.8. На цій сторінці користувач має ввести данні вже зареєстрованого користувача для того щоб потрапи на для списку замовлень. Вигляд форми авторизації зображений на рисунку 3.8.

Please enter your credentials below to access your Orders

Username

Password

Login

Рисунок 3.8 – Форма авторизації

Вигляд сторінки замовлень зображений на рисунку 3.9. Ця сторінка отримує дані з бази даних для користувача який був авторизований. Сторінка була розроблена за допомогою мови програмування HTML, CSS, а логіка за допомогою JAVA.

Online Shop

Your Orders:

Order ID: 1 2022-06-06 Total: 30.7\$ Cheeseburger, 9.9\$ Limede, 5.9\$ Repeat Order	Order ID: 2 2022-06-06 Total: 0.0\$ Repeat Order	Order ID: 3 2022-06-06 Total: 24.8\$ Cheeseburger, 9.9\$ Chicken Bucket, 14.9\$ Repeat Order
Order ID: 4 2022-06-06 Total: 0.0\$ Repeat Order		

Рисунок 3.9 – Сторінка замовлень

Під час дипломного проектування необхідно було створити 5 таблиць, за допомогою яких було здійснене внесення та вивантаження інформації, яка відображається на сайті. Таблиці бази даних зображені на рисунках 3.11-3.12.

	Field	Type	Null	Key	Default	Extra
1	category_id	int	NO	PRI	<null>	auto_increment
2	name	varchar(20)	NO		<null>	
3	description	varchar(40)	YES		<null>	

Рисунок 3.11 – Таблиця “Categories” яка відповідає за інформацію про категорії товарів

	Field	Type	Null	Key	Default	Extra
1	details_id	int	NO	PRI	<null>	auto_increment
2	product_name	varchar(20)	NO		<null>	
3	product_id	int	NO	MUL	<null>	
4	product_price	float	NO		<null>	
5	order_id	int	NO		<null>	

Рисунок 3.12 – Таблиця “Order_details” яка відповідає за деталі замовлення

	Field	Type	Null	Key	Default	Extra
1	order_id	int	NO	PRI	<null>	auto_increment
2	date	date	NO		<null>	
3	total_price	float	NO		<null>	
4	user_id	int	NO	MUL	<null>	

Рисунок 3.13 – Таблиця “Orders” яка відповідає за загальну інформацію про замовлення

	Field	Type	Null	Key	Default	Extra
1	product_id	int	NO	PRI	<null>	auto_increment
2	name	varchar(20)	NO		<null>	
3	price	float	NO		<null>	
4	description	varchar(40)	YES		<null>	
5	category_id	int	NO	MUL	<null>	

Рисунок 3.14 – Таблиця “Products” яка містить інформацію про продукти

	Field	Type	Null	Key	Default	Extra
1	user_id	int	NO	PRI	<null>	auto_increment
2	name	varchar(20)	NO		<null>	
3	password	varchar(20)	NO		<null>	
4	address	varchar(30)	NO		<null>	

Рисунок 3.14 – Таблиця “Users” яка містить інформацію про зареєстрованих користувачів у веб-додатку

Підключення бази даних до веб-додатку відбувається за допомогою цієї частини коду яка наведена у Додатку В.

Після підключення до бази даних, посилається запит контролера, який обробляє це та в свою чергу відправляє запит до бази даних. У відповіді від бази контролер обробляє ці дані та надсилає на сторінку для їх подальшого рендерінгу. (код наведено у додатку В)

3.3 Використання програмного додатку

Основний сценарій використання програмного продукту це замовлення товару. З моменту коли користувач зайшов на сайт до того моменту коли він побачить сторінку підтвердження.

Користувач заходить на сторінку-вітрину та бачить перед собою перелік товарів. Ілюстрація зображена на рисунку 3.15.

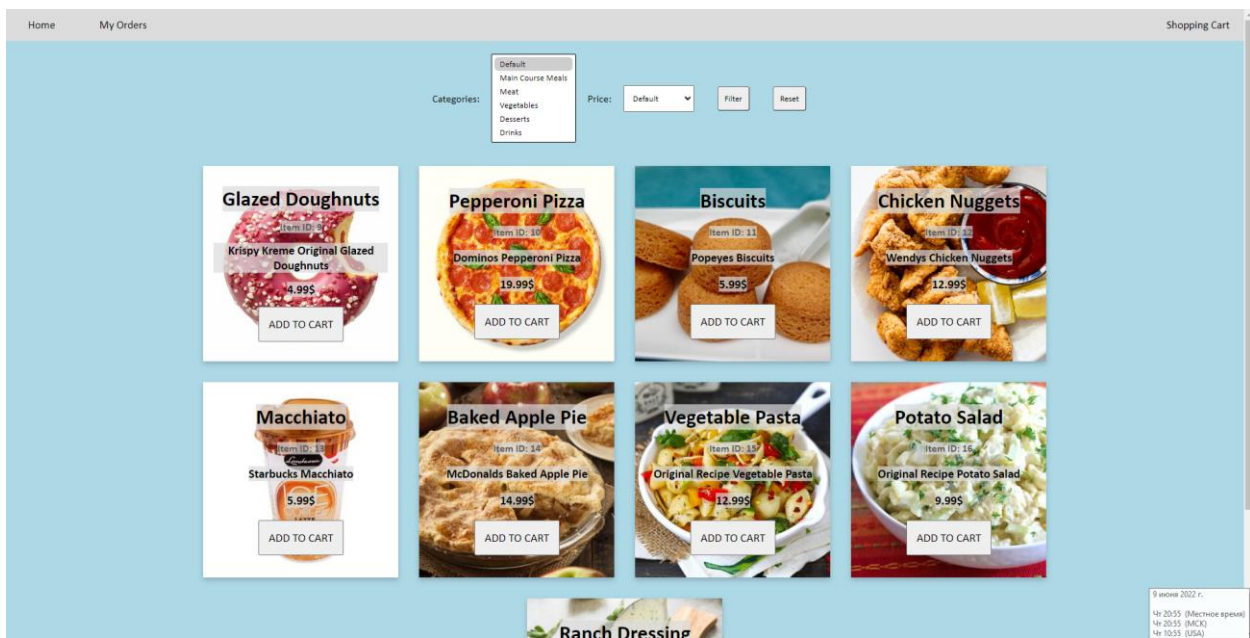


Рисунок 3.15 – Сторінка вітрина

Далі користувач може відфільтрувати за необхідними фільтрами та потім додає бажані товари в корзину. Корзина представлена у вигляді модального вікна та зображена на рисунку 3.16

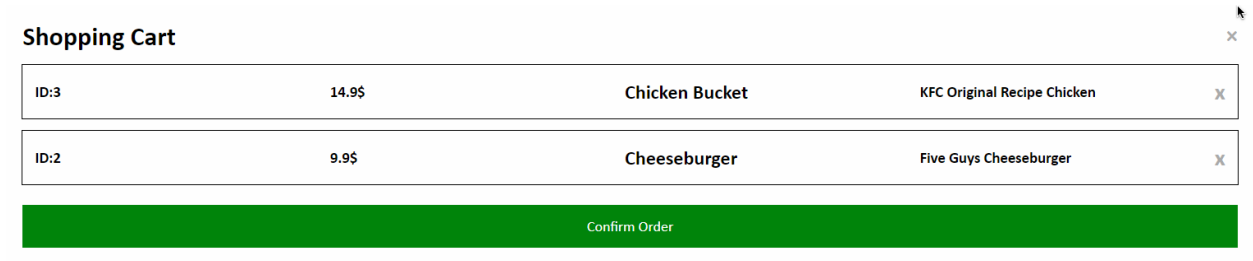


Рисунок 3.16 – Корзина

Користувач може видалити товари та перейти на сторінку підтвердження замовлення, яка зображена на рисунку 3.17.

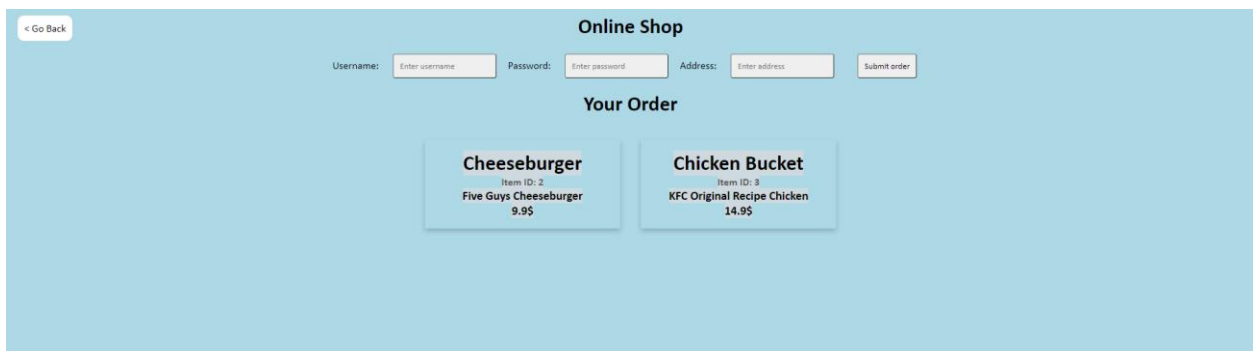


Рисунок 3.17 – Корзина

На цій сторінці користувач ще раз переглядає обрані ним товари, та бачить перед собою поля для вводу своїх даних. Якщо він вводить ім'я та пароль вже існуючого користувача, то він не буде зареєстрований заново, а виконуватиметься вхід у вже існуючий аккаунт. Після натискання кнопки «підтвердження замовлення» користувач потрапляє на сторінку яка інформує його про успішне замовлення. Вона зображена на рисунку 3.18.



Рисунок 3.18 – Сторінка успішного замовлення

Далі користувач настикає кнопку «Home» та потрапляє на сторінку-вітрину. З неї користувач може потрапити на сторінку з своїми замовленнями, клікнувши по кнопці «My Orders». Вона зображена на рисунку 3.19.

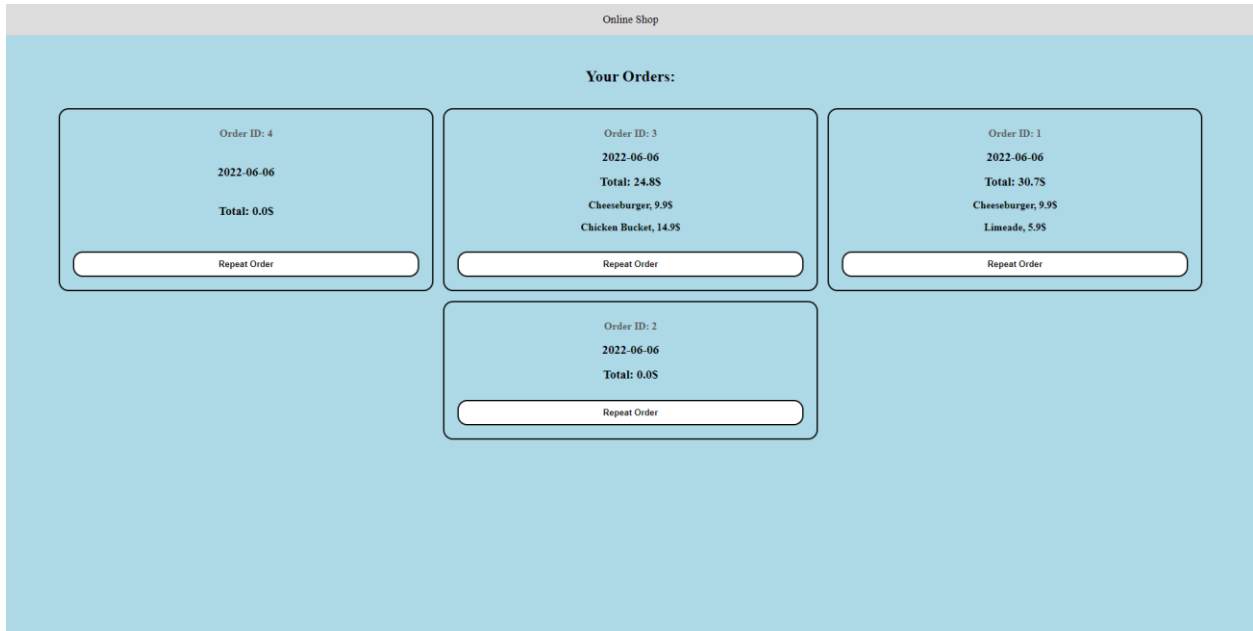


Рисунок 3.19 – Сторінка замовлень користувача

Далі користувач може повторити замовлення. Для цього необхідно клікнути по кнопці «Repeat Order». Після натискання корзина користувача буде сформована згідно з тим замовленням, яке він побажав повторити. Це зображено на рисунку 3.20.

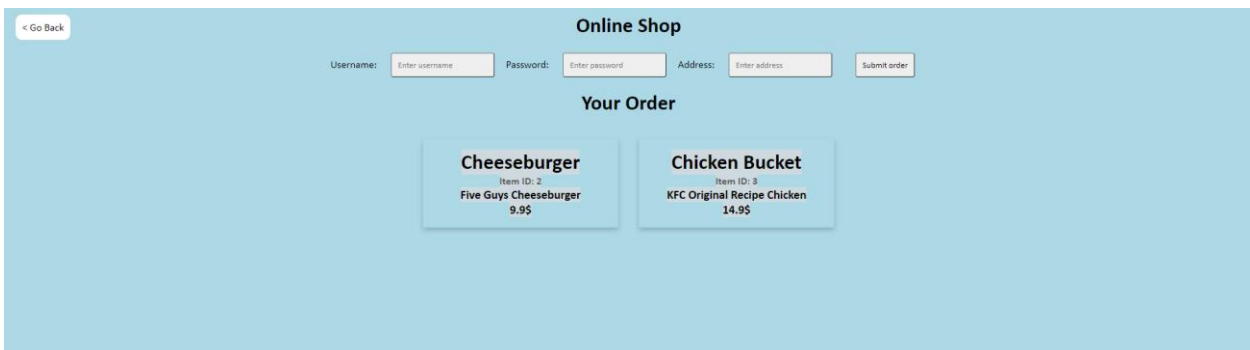


Рисунок 3.20 – Сторінка підтвердження користувача

ВИСНОВОК

В результаті виконання бакалаврської роботи розроблено веб-додаток підтримки діяльності магазину продуктів. У роботі над дипломним проектом було проведено аналіз предметної області. Проведений аналіз аналогів дозволив визначити основні вимоги та характеристики створення продукту.

Сформульовано мету та завдання для розробки проекту «Веб-додаток підтримки діяльності магазину продуктів». Також визначені шляхи та засоби розробки веб-додатку.

У другій частині дипломної роботи здійснене структурно-функціональне моделювання, побудовані необхідні діаграми.

Програмна реалізація здійснюється в наступній частині роботи. Для цього були розроблені креслення високого рівня, розроблені макети та розроблена база даних.

У Додатку Б представлені плани роботи та проаналізовані ризики, які можуть виникнути під час розробки програмного забезпечення.

Результатом цього проекту є програмний продукт у вигляді веб-додатка, створеного для полегшення споживачам користування послугою, тобто замовлення продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Інтернет магазин Розетка/Торгівельна площадка. [Електронний ресурс] – Режим доступу до ресурсу: <https://rozetka.com.ua/> (дата звернення: 17.05.2022).
- 2 Торгівельна площадка [Електронний ресурс] – Режим доступу до ресурсу: <https://www.olx.ua/>. (дата звернення: 18.05.2022).
- 3 Що таке інформаційні технології [Електронний ресурс] – Режим доступу до ресурсу: <http://apers.kpi.ua/shcho-take-informatsiini-technologii> (дата звернення: 05.12.2022).
- 4 Готовая CMS или самописный сайт? [Електронний ресурс] – Режим доступу до ресурсу: <https://hyperhost.ua/info/gotovaya-cms-ili-samopisnyiy-sayt/> (дата звернення: 07.12.2022).
- 5 Глава 4. Диаграмма вариантов использования (use case diagram) [Електронний ресурс] – Режим доступу до ресурсу: <http://khpriip.mipk.kharkiv.edu/library/case/leon/gl4/gl4.html#1> (дата звернення: 10.12.2019).
- 6 Методология IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: https://studme.org/87184/ekonomika/metodologiya_Idef0 (дата звернення: 10.12.2022).
- 7 Диаграммы автоматов. [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tem_12/tem_12_3 (дата звернення: 13.12.2022).
- 8 Рекомендация: Диаграмма последовательности [Електронний ресурс] – Режим доступу до ресурсу: http://dit.isuct.ru/Publish_RUP/core.base_rup/guidances/guidelines/sequence_diagram_AFA76EBB.html (дата звернення: 18.12.2022).
- 9 Глава 5 Диаграмма классов (class diagram) [Електронний ресурс] – Режим доступу до ресурсу: <http://khpriip.mipk.kharkiv.edu/library/case/leon/gl5/gl5.html> (дата звернення: 18.12.2022).

10 ОМПС. Діаграми діяльності [Електронний ресурс] – Режим доступу до ресурсу: https://studopedia.su/16_5481_omps-diagrami-diyalnosti.html (дата звернення: 19.12.2020).

11 Інформаційне моделювання Idefi і моделювання даних в бізнес-процесах (Ideflx) [Електронний ресурс] – Режим доступу до ресурсу: https://stud.com.ua/113637/menedzhment/informatsiyne_modelyuvannya_Idefi_modelyuvannya_danih_biznes_protsesah_Ideflx

12 Проектування баз даних [Електронний ресурс] – Режим доступу до ресурсу: https://pIdru4niki.com/11570718/bankivska_sprava/proektuvannya_baz_danih

13 Базы данных сайта [Електронний ресурс] – Режим доступу до ресурсу: <https://semantica.in/blog/baza-dannykh-sajta.html>

14 Infographic: HTML vs CSS? [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.codingdojo.com/blog/html-vs-css>

15 UML для бізнес-моделювання: для чого потрібні діаграми процесів [Електронний ресурс]. – Режим доступу до ресурсу: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>

ДОДАТОК А**ТЕХНІЧНЕ ЗАВДАННЯ
на розробку
Web-додатку підтримки діяльності магазину продуктів»****ПОГОДЖЕНО:**

Доцент кафедри комп'ютерних наук

_____ Федотова Н.А.

Студент групи ІТ-82-0

_____ Станкевич Д.В.

Суми 2022

1 Призначення й мета створення веб-системи

1.1 Призначення веб-системи

Веб-система повинна представляти собою веб-сторінки для пошуку необхідних товарів та їх замовлення.

1.2 Мета створення інформаційної системи

За допомогою цього веб-додатку можливо буде обирати необхідні товари та додати їх до кошика формуючи замовлення.

Надає можливість сформувати кошик з попереднього замовлення.

1.3 Цільова аудиторія

- Користувачі віком від 18 до необмеженого віку, які мають бажання придбати доступні товари.

- Інвестори.

2 Вимоги до веб-системи

2.1 Вимоги до веб-системи в цілому

2.1.1 Вимоги до структури й функціонування веб-системи

Веб-система повинна бути реалізована у вигляді веб-сторінок пов'язаних між собою.

2.1.2 Вимоги до персоналу

Для експлуатації від користувачів не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів.

2.1.3 Вимоги до збереження інформації

У системі керування додатком передбачено механізм вибірки інформації та передбачений пошук, а також збереження особистих даних.

2.1.4 Вимоги до розмежування доступу

Інформація є безкоштовною.

Розмежування доступу до інформації є у такому вигляді - незареєстрований користувач може лише бачити головну сторінку. Для перегляду каталогу товарів необхідно зареєструватися або увійти до існуючого аккаунту. Реєстрація - безкоштовна

2.2 Вимоги до функцій, виконуваних додатком

2.2.1 Основні вимоги

2.2.1.1 Структура додатку

Веб-додаток складається з наступних розділів:

- Головна сторінка – сторінка на яку потрапляє користувач переходячи на домен. Вона є свого роду вітальною листівкою. З неї користувач може потрапити до будь-якої сторінки сайту. Наприклад для того щоб переглянути каталог товарів користувачеві необхідно увійти до системи або зареєструватися.
- Сторінка реєстрації – сторінка, на якій користувач створює свій власний обліковий запис.
- Сторінка авторизації – сторінка, на якій користувач входить до вже існуючого облікового запису.
- Сторінка каталог - сторінка на якій будуть розміщені товари
- Сторінка-корзина - сторінка на якій будуть показані всі додані користувачем товари.
- Сторінка персональної інформації користувача - на якій буде розміщуватися необхідна для замовлення інформація
- Сторінка замовлень - сторінка на якій адміністратор має змогу подивитися усі замовлення
- Сторінка редагування каталогу - сторінка на якій адміністратор матиме змогу змінити товари, що бачать користувачі

2.2.1.2 Навігація

Користувацький інтерфейс додатку повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації. Навігаційні елементи подані у вигляді кнопок, з назвами на них.

2.2.1.3 Наповнення додатку

Вибірка даних, що хотів отримати користувач, буде формуватися з бази даних, інформація до якої буде вноситися та оновлюватися співробітниками компанії

2.2.1.4 Система навігації (карта додатку)

Дизайн додатку представлено на рисунку А.1.

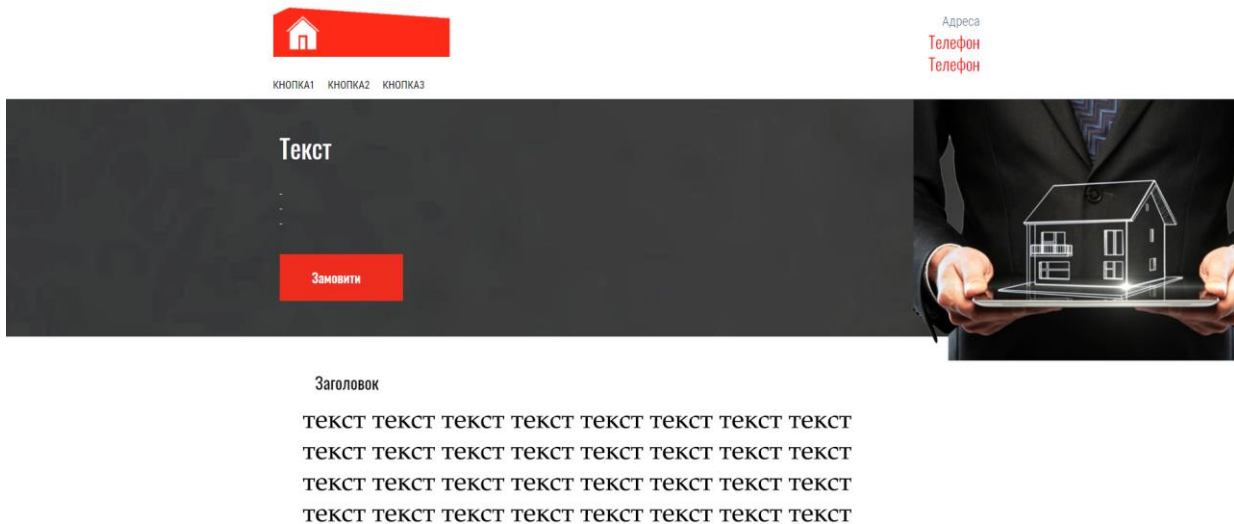


Рисунок А.1 – Прототип головної сторінки додатку

2.2.2 Вимоги до функціональних можливостей

Система надає можливість для перегляду інформації про квартири, що відповідають умовам, які користувач самостійно ввів до відповідних фільтрів.

2.2.2.2 Загальні вимоги

Стиль додатку можна описати як сучасний, діловий. У якості фоновому кольору рекомендується використовувати білий. Він повинен бути інтуїтивно зрозумілим. Розташування елементів на сторінці додатку схематично показано на рис. А.1.

2.3 Вимоги до видів забезпечення

2.3.1 Вимоги до інформаційного забезпечення

Реалізація веб-сторінки відбувається з використанням мов програмування:

- HTML (ver 5.0).
- CSS (ver 3.0).
- JAVA SCRIPT
- SQL.
- JAVA.

2.3.2 Вимоги до лінгвістичного забезпечення

Додаток повинен бути виконаний українською мовою.

2.3.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Браузер «Google Chrome» (версія: починаючи з 77 і вище).

2.3.4 Вимоги до апаратного забезпечення

Апаратне забезпечення повинне задовольняти наступним вимогам:

- Доступ до мережі інтернет
- Оперативна пам'ять: не менше ніж 500 мб.

Апаратне забезпечення повинне відповідати тим технічним вимогам, що зазначено в 2.2.3.

ДОДАТОК Б

Планування робіт

Сьогодні розвиток технологій є досить важливим питанням у нашому житті. Тому велика кількість ресурсів направлено на їх розвиток. Однією з галузей розвитку є розробка веб-сайтів та додатків. Веб-сайт — це сукупність програмних, інформаційних, а також медійних засобів, логічно пов'язаних між собою. По суті ж веб-сайт — це віддзеркалення успішності фірми, її обличчя.

В даний час кількість нових сайтів збільшується з неймовірною швидкістю. Саме тому, основним завданням, що ставиться при розробці сайту – зробити його максимально привабливим для потенційних користувачів і надати йому індивідуальність.

Деталізація мети проекту методом SMART. Для того, щоб проект був успішним та міг протистояти сучасним аналогам, треба на початковому етапі правильно визначити його мету за допомогою SMART-методу. У таблиці Б.1 наведені результати деталізації проекту методом SMART.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Створити веб-додаток для онлайн замовлення продуктів.
Measurable (вимірювана)	Розробити веб-додатку в короткий термін з використанням мінімальної кількості ресурсів.
Achievable (досяжна, узгоджена)	Для реалізації задачі необхідні знання HTML, CSS, PHP/Java, JavaScript, MySQL та навичок написання документації.
Relevant (реалістична)	Розроблений веб-додаток дозволить користувачу замовити необхідні йому товари прямо до дверей.
Time-framed (обмежена в часі)	Мета проекту має часове обмеження. Дати виконання проекту погоджені між замовником та виконавцем.

Планування змісту роботи. WBS (Work Breakdown Structure – Ієрархічна структура роботи) – це графічний вигляд елементів проекту, які згруповані

ієрархією у єдине ціле з продуктом проекту. Дана методика планування змісту робіт застосовується для досконалого виконання робіт частинами і представляє собою ключову частину проекту, за допомогою якої організовується командна робота. Елементами схеми можуть виступати послуги, дані та продукти. WBS забезпечує можливість для оцінки термінів та контролю за графіком робіт.

На першому рівні даної методології розташовується сам продукт. На другому рівні знаходяться основні дії спрямовані на досягнення мети проекту. Декомпозиція виконується до етапу, поки вони не стануть простими. Тобто не стануть діями, які мають однозначний чіткий результат, та призначені конкретній особі, для якої можна обчислити витрати таких ресурсів, як праця та тривалість. На рисунку Б.1 зображена схема WBS.

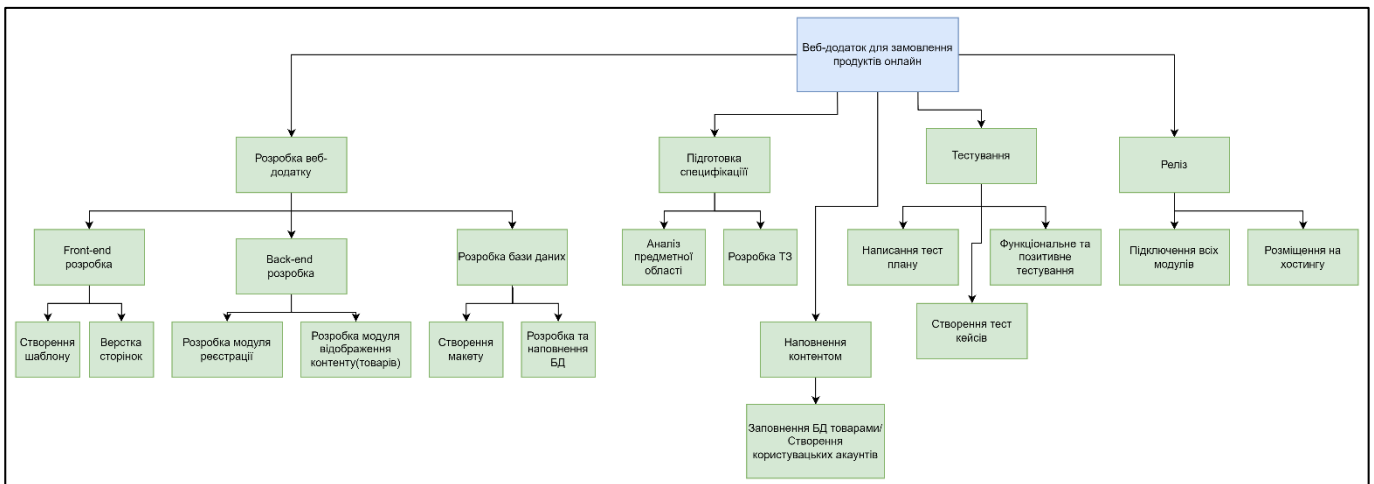


Рисунок Б.1 – WBS – структура

Планування структури виконавців. Після створення WBS структури проекту починається наступний етап – розробка структури виконавців або також звана OBS, яка представляє собою графічну структуру відображення учасників або відповідальних осіб, які беруть участь у створенні та реалізації проекту.

На рисунку Б.2 зображено OBS структуру планування проекту. Виконавці проекту зазначені у таблиці Б.2.

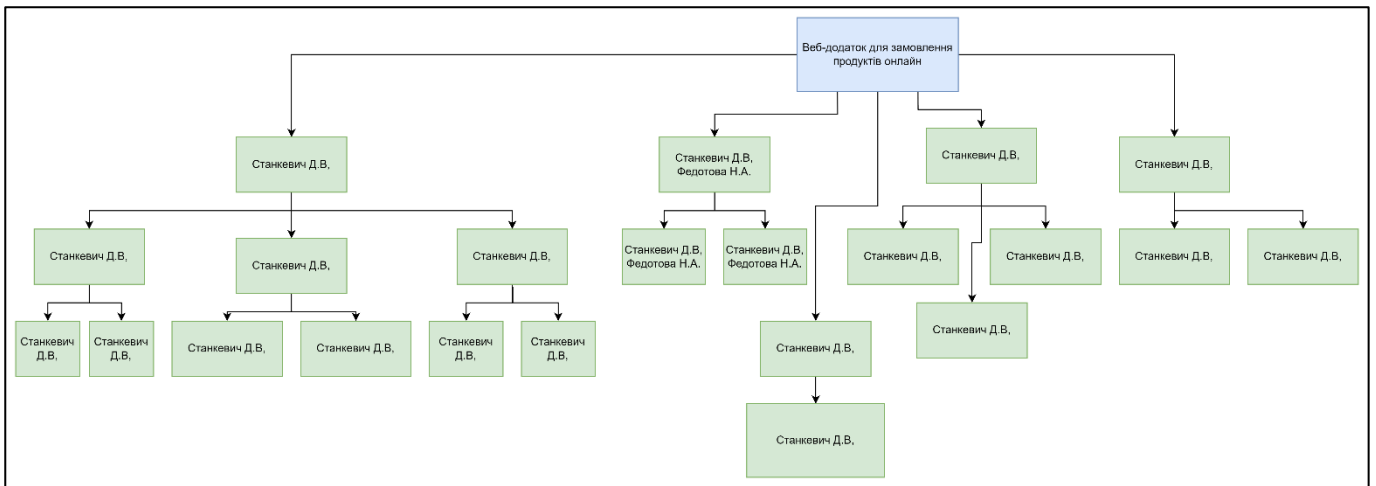


Рисунок Б.2 – OBS структура проекту

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Станкевич Д.В.	Виконання front-end та back-end розробки проекту
Проектувальник	Станкевич Д.В.	Виконує проектування та розробку бази даних
Тестувальник	Станкевич Д.В.	Виконує розробку тест плану, тест кейсів та функціональне тестування
Керівник проекту	Федотова Н.А.	Формує завдання на розробку проекту, затверджує ТЗ
Менеджер проекту	Станкевич Д.В.	Відповідальний за своєчасне виконання проекту та розподіл ресурсів проекту.

Діаграма Ганта. Одним з важливих етапів планування проекту є побудова діаграми Ганта, що представляє собою розклад виконання робіт з розподілом часу. Дякуючи діаграмі можна отримати реальні дати закінчення проекту та більш чітко розподілити витрати ресурсів. Діаграма Ганта проекту зображена на рисунках Б.3.



Рисунок Б.3 – Календарний графік проекту

Управління ризиками проекту. Під час оцінки ризиків проекту, потрібно визначити ризики, які мають бути усунені якнайшвидше для економії часових та грошових ресурсів. Враховуючи ступень важливості ризику, кожен ризик має свій пріоритет та буде виправлений в залежності від пріоритету. У таблиці Б.3 зображено шкалу для класифікації ризиків за впливом на проект.

Таблиця Б.3 – Оцінювання ризиків за ймовірністю виникнення.

Оцінка	Ймовірність	Вплив	Тип
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Для зниження негативного впливу ризиків на проект потрібно планувати реагування на них. План включає в себе визначення ефективності розробки та оцінювання наслідків ризиків на проект. Далі була створена матриця ймовірності виникнення ризиків та вплив їх на проект, за допомогою планування.

Відповідно до отриманого значення індексу, створено класифікація ризиків за рівнем, зображену в таблиці Б.4. Описання ризиків та стратегії реагування на них наведені в таблиці Б.5.

Таблиця Б.4 – Шкала оцінювання.

№	Назва	Межі	Ризики
1	Прийнятні	$1 \leq R \leq 2$	1,9
2	Виправдані	$3 \leq R \leq 4$	2,4,6,8,10
3	Недопустимі	$6 \leq R \leq 9$	3,5,7

Таблиця Б.5 – Описання ризиків

№	Опис ризику	Ймовірність виникнення	Назва	Ранг	План А	Тип реагування	План Б
RS_1	Конфлікт між розробником та замовником	Низька	Середній	3	На початку створення проекту налагодити відносини між замовником та розробником та визначитися с умовами	Попередження	При виявленні ризику спробувати дізнатися причину виникнення та домовитися про умови вирішення
RS_2	Нерозуміння документації	Низька	Високий	4	Зв'язатися з замовником та вирішити непорозуміння	Прийняття	Спробувати запропонувати свою стратегію розробки
RS_3	Не розрахування часових ресурсів	Висока	Високий	9	Зв'язок з розробником для збільшення часу на розробку. Приведення чітких аргументів	Пом'якшення	Аналіз часу на розробку проекту, та змінення порядку розробки модулів, можливе зменшення часу на виконання
RS_4	Постійна зміна ТЗ	Середній	Середній	4	Чітко описати всі вимоги проекту та встановити умови реалізації	Перенесення	Узгодити умови внесення змін на проєкті та встановити всі положення
RS_5	Помилки в розробці	Висока	Високий	9	Продумувати розробку на етапі	Пом'якшення	Здійснювати проміжне

№	Опис ризику	Ймовірність виникнення	Назва	Ранг	План А	Тип реагування	План Б
					проектування та створення документації		тестування на кожному етапі.
RS_6	Не ефективна технологія розробки	Низька	Середній	3	Провести аналіз методів розробки та обрати найпростішу	Пом'якшення	Проаналізувати основні методи розробки та обрати підходящий.
RS_7	Помилка при створенні документації	Висока	Високий	9	Починати тестування на етапі створення документації	Пом'якшення	
RS_8	Збої в роботі програми	Середня	Середній	4	Регулярне створення резервних копій.	Попередження	Зміна сервера та забезпечення
RS_9	Реалізація зайвого функціоналу	Низька	Низький	1	Слідування раніше створеній та узгодженій документації	Пом'якшення	
RS_10	Недостатня кількість тестового покриття	Середня	Середній	4	Залучення спеціалістів	Попередження	Створення тест плану та тест кейсів на початку розробки

ДОДАТОК В

ЛІСТИНГ ПРОГРАМНОГО КОДУ ОСНОВНИХ МОДУЛІВ WEB-ДОДАТКУ

Підключення бази даних до веб-додатку

```

public class DBConnectionPool {
private static BasicDataSource ds = new BasicDataSource();
static {
    ds.setUrl("jdbc:mysql://localhost:3306");
    ds.setUsername("root");
    ds.setPassword("mysqlroot");
    ds.setMinIdle(5);
    ds.setMaxIdle(10);
    ds.setMaxOpenPreparedStatements(100);
    ds.setDefaultCatalog("ecomm_schema");
    ds.setDriverClassName("com.mysql.cj.jdbc.Driver");
}
public static Connection getConnection() throws SQLException {
    return ds.getConnection();
}
private DBConnectionPool() { }
}

```

запит контролера

```

public List<Product> getAllProducts() {
List<Product> productList = new ArrayList<>();
try {
    connection = DBConnectionPool.getConnection();
    statement = connection.createStatement();
    resultSet = statement.executeQuery("SELECT * FROM PRODUCTS");
    while (resultSet.next()) {
        productList.add(parseProduct(resultSet));
    }
    connection.close();
} catch (SQLException e) {
    e.printStackTrace();
}
}

```

```
    }  
    return productList;  
}
```