

Міністерство освіти і науки України
Український державний університет науки і технологій

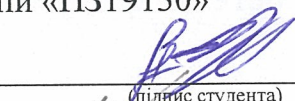
Факультет «Комп'ютерні технології і системи»
Кафедра «Комп'ютерні інформаційні технології»

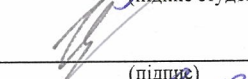
Пояснювальна записка
до кваліфікаційної роботи бакалавра

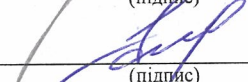
на тему: «Реалізація інтернет – магазину на платформі ASP. NET з використанням архітектурного патерну MVC»
за освітньою програмою: «Інженерія програмного забезпечення»
зі спеціальності: «Інженерія програмного забезпечення»
Виконав: студент групи «ПЗ19130»

Керівник:

Нормоконтролер:


(підпис студента)


(підпис)


(підпис)

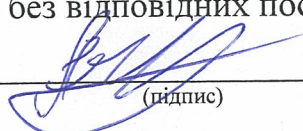
/Олександр ГЕТМАНЕНКО/
(Ім'я ПРІЗВИЩЕ)

/доц. Олександр ІВАНОВ/
(посада, Ім'я ПРІЗВИЩЕ)

/доц. Олена КУРОП'ЯТНИК/
(посада, Ім'я ПРІЗВИЩЕ)

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

Студент


(підпис)

Дніпро – 2022 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»
Department «Computer information technology»

Explanatory Note
to Bachelor's Thesis

on the topic: «Implementation of an online store on the platform ASP .NET using
architectural MVC pattern»
according to educational curriculum «Software engineering»
in the Speciality: «121 Software engineering»


Done by the student of the group PZ19130:	<u>/Oleksandr HETMANENKO/</u>
Scientific Supervisor:	<u>/Oleksandr IVANOV/</u>
Normative controller:	<u>/Olena KUROPIATNYK/</u>

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: «Комп'ютерні технології і системи»
Кафедра: «Комп'ютерні інформаційні технології»
Рівень вищої освіти: бакалавр
Освітня програма: «Інженерія програмного забезпечення»
Спеціальність: «121 Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри КІТ

 /Вадим ГОРЯЧКІН/
(підпис)

Дата 22.12.2021

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра
студенту Гетманенку Олександру Руслановичу

1. Тема роботи: «Реалізація інтернет – магазину на платформі ASP. NET з використанням архітектурного патерну MVC»

Керівник роботи: Іванов Олександр Петрович, доцент
затверджені наказом № 77 ст від 08.12.2021

2. Строк подання студентом роботи: 28.05.2022 р.

3. Вихідні дані до роботи: _____

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

Збір та аналіз вимог, зовнішнє проектування, внутрішнє проектування,
проектування архітектури системи, розробка програми, тестування та
налагодження, висновки та рекомендації

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

Відео демонстрації роботи програми, презентація

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Технічне завдання: Постановка задачі, збір інформації, вибір та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	31.01.22 – 18.02.22	
2	Робочий проект. Програмування та відладка програми.	19.02.22 – 01.05.22	
3	Робочий проект. Тестування програми	02.05.22 – 24.05.22	
4	Робочий проект. Розробка, узгодження і затвердження програмної документації.	25.05.22 – 07.06.22	
5	Подання кваліфікаційної роботи до кафедри	08.06.22	
6	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	23.06.22	

Студент

(підпис)

Олександр ГЕТМАНЕНКО

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

(підпис)

доц. Олександр ІВАНОВ

(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Метою дипломного проекту є створення сайту інтернет-магазину, який буде легко масштабуватись завдяки використовуваному патерну MVC та міститиме зрозумілу систему маршрутизації в рядку URL.

Об'єктом дипломної роботи є методологія розробки веб-застосунків на платформі ASP .Net, з використанням патерну MVC, яка є схожою до платформи Rails, що також використовує даний патерн при розробці веб-застосунків. Компанія Microsoft анонсувала в 2007 році платформу ASP .Net, MVC, й продовжує підтримувати її на теперішній день.

Предметом дипломної роботи є сайт, який побудований на вказаній платформі з використанням вказаної методології, що дає йому широкі можливості розширюваності, тестування та забезпечує зрозумілу систему маршрутизації в рядку URL.

Пояснювальна записка складається зі вступу, 6 розділів, висновків, бібліографічного списку та 4 додатків:

- вступ описує актуальність побудови сайту та засоби його розробки мовою C#, містить 2 сторінки;
- перший розділ містить 3 сторінки, описує постановку задачі та загальні вимоги до сайту;
- другий розділ містить 8 сторінок, описує призначення сайту та опис функціональних вимог;
- третій розділ містить 8 сторінок, описує архітектуру системи;
- четвертий розділ містить 9 сторінок де описані внутрішня будова сайту;
- п'ятий розділ містить 6 сторінок, описує алгоритм роботи модулів та їх рівні;
- шостий розділ містить 4 сторінки, описує процес тестування та налагодження програми;
- додатки містять технічне завдання, опис та специфікацію програми.

Усього робота містить: таблиць – 11, рисунків – 21, бібліографія – 1 джерело.

Ключові слова: C#, ASP.NET, MVC PATTERN, VIEW, CONTROLLER, MODEL, DATABASE, ENTITY FRAMEWORK, SITE, SQL, SQL SERVER, WEB.

ЗМІСТ

Вступ.....	3
1 Збір та аналіз вимог.....	5
Висновки до розділу 1	7
2 Зовнішнє проєктування	8
Висновки до розділу 2	10
3 Внутрішнє проєктування.....	16
Висновки до розділу 3	29
4 Проєктування архітектури системи.....	30
Висновки до розділу 4	38
5 Розробка програми	39
Висновки до розділу 5	43
6 Тестування та налагодження.....	44
Висновки до розділу 6	46
Висновки та рекомендації	47
Список використаної літератури	48
Додатки.....	49

ВСТУП

Програмний додаток «Інтернет-магазин на платформі asp.net з використанням архітектурного патерну mvc», що розробляється призначений для продажу товарів покупцям.

В сучасному світі досить часто виникає потреба в розробці веб-застосунків різної складності та масштабів. Майже кожен розвинений магазин приходить до необхідності в розробці сайту, який його рекламуватиме та, найбільш необхідне, надаватиме користувачам можливість ознайомитись з асортиментом товарів й придбати їх.

При побудові веб-застосунків на мові C# використовують декілька спеціальних фреймворків, до них належать ASP.Net MVC та ASP.Net core, які розширюють платформу розробників .NET інструментами та бібліотеками спеціально для створення веб-застосунків.

Також до їх складу входить веб-платформа Blazor, що дозволяє створювати інтерактивні веб-інтерфейси, використовуючи C# замість JavaScript для швидкого створення вмісту сторінок за допомогою коду вбудованого в html розмітку сторінок. Програми Blazor складаються з багаторазових компонентів веб-інтерфейсу, реалізованих за допомогою C#, HTML і CSS.

При використанні архітектури MVC сайт може легко поліпшуватись та розширюватись, всі рівні системи розділені між собою, що дозволяє розробляти їх окремо один від одного.

Особливістю даного фреймворку є те, що він надає зрозумілу адресу ресурсу, що дозволяє легко орієнтуватись у сайті й швидко визначати на якій сторінці знаходиться користувач.

Архітектура ASP.Net MVC часто використовується при розробці сайтів, що підтверджує актуальність її вибору. Паралельно до неї набуває актуальності фреймворк ASP.Net core, проте здебільшого він має схожу до обраної архітектури структуру сайту. На даний момент цей фреймворк лише набуває популярності, тому його використання має менші переваги щодо готових програмних рішень та бібліотек що додаються до них. Зважаючи на вище зазначені недоліки фреймворку ASP.Net

Core було обрано фреймворк ASP.Net MVC.

При роботі сайту використовується фреймворк Bootstrap 4.6.1, який надає можливості для швидкої розробки адаптивних сторінок сайтів, також використовується бібліотека JavaScript – JQuery, що дозволяє легко отримувати доступ до елементів сторінки та їх даних, а також керувати ними, за допомогою цієї бібліотеки можливо легко оперувати іншими елементами сайту, наприклад, можливо легко обробляти вміст URL (адреси ресурсу), що значно полегшує виконання запитів та перехід до сторінок сайту за допомогою скриптів.

JQuery надає потужні можливості по обробці інтерфейсової частини на стороні клієнту й за необхідності виконання запитів до серверу.

Область застосувань: програмний продукт призначений для власників магазинів та користувачів інтернет-магазинів.

Експлуатаційне призначення – за допомогою інтернет-магазину продавець подає свій товар в зручному для покупців вигляді, а покупці в свою чергу отримують доступ до повного асортименту товарів магазину, мають можливості ознайомлення з їх детальними характеристиками, швидкого пошуку згідно наданих критеріїв та за потреби можливість придбання обраних товарів.

1 ЗБІР ТА АНАЛІЗ ВИМОГ

Постановка задачі:

- розробити програмний продукт, що представляє собою веб-портал, що є інтернет-магазином;
- інтернет-магазин використовує зрозумілу для користувачів URL адресацію, внутрішня архітектура проекту повинна відповідати принципу побудови додатків Model-View-Controller.

В результаті збору та аналізу вимог від замовника було отримано функціональні особливості сайту.

Система забезпечує можливість авторизації користувача та в разі відсутності аккаунту його створення.

Також система забезпечує надсилання повідомлення на пошту через поштовий сервіс «mailtrap.io».

Можливості користувача:

- редагування профілю (ім'я, прізвище, електронна пошта, логін, пароль);
- перегляд товарів;
- фільтрація товарів за такими критеріями, як: категорія, ціна;
- пошук товарів за назвою;
- додання товарів до списку покупок;
- видалення товарів зі списку покупок;
- зменшення кількості товарів у списку покупок;
- збільшення кількості товарів у списку покупок;
- оплата замовлення через систему платежів liqpay;
- перегляд історії замовлень;
- перегляд головної сторінки та сторонніх сторінок.

Можливості адміністратора:

- редагування профілю (ім'я, прізвище, електронна пошта, логін, пароль);
- перегляд товарів;
- фільтрація товарів за такими критеріями, як: категорія, ціна;

- пошук товарів за назвою;
- перегляд історії замовлень;
- перегляд головної сторінки та сторонніх сторінок;
- редагування головної сторінки та сторонніх сторінок;
- додання сторонніх сторінок (наповнення контенту);
- видалення сторонніх сторінок;
- сортування категорій;
- додання категорій (назва);
- редагування категорій;
- видалення категорій;
- додання товару;
- редагування товару (назва, опис, ціна, категорія, основне фото та фото галерея, помітка про наявність (видалено чи доступно));
- видалення товару;
- перегляд контенту призначеного для користувача, без можливості роботи зі списком покупок.

Сайт повинен мати адаптивний дизайн, при змінні розмірів екрану змінюється його дизайн. В головному меню розташовуються посилання на сторінки, сайту, побічні сторінки у вигляді випадаючого списку, а також посилання на профіль користувача й кнопка авторизації / виходу. Для адміністратора в правій частині меню знаходиться кнопка перемикавання режимів з адміністратора на користувача і навпаки.

В режимі адміністратора у випадаючому списку «Menu» повинні відображатись наступні сторінки:

- Pages – робота зі сторінками;
- Sidebar – робота з колонкою сайту, яка відображається справа на вказаних сторінках (може використовуватись для рекламування);
- Categories – робота зі списком категорій та керування категоріями;
- Products – робота з товарами;
- Orders – перегляд списку замовлень користувачів.

Висновки до розділу 1

На етапі збору вимог від замовника було отримано вимоги щодо дизайну сайту, його адаптивності на різних пристроях та функціональних можливостей.

Проаналізувавши вимоги замовника було обрано основні з них для першої ітерації проекту.

На наступні ітерації було висунуто такі вимоги:

- покращення адаптивності сайту;
- додання видів фільтрації;
- додання нових властивостей до товару в залежності від його категорій;
- додання сторінок зі спеціальними інтерфейсом для створення та редагування властивостей товару;
- властивості товару додаються до нього не за допомогою одного поля опису, а за допомогою списку властивостей товару, які можливо додатково налаштовувати;
- розробити систему налаштувань властивостей товару.

2 ЗОВНІШНЄ ПРОЄКТУВАННЯ

Функціональне призначення полягає в організації роботи магазину через інтернет ресурс.

Експлуатаційне призначення – за допомогою інтернет-магазину продавець подає свій товар в зручному для покупців вигляді, а покупці в свою чергу отримують доступ до повного асортименту товарів магазину, мають можливості ознайомлення з їх детальними характеристиками, швидкого пошуку згідно наданих критеріїв та за потреби можливість придбання обраних товарів.

Функціональні вимоги – програмний продукт надаватиме наступні можливості в залежності від рівня доступу до роботи сайту.

Покупець:

- додання товарів до кошику;
- видалення товарів з кошику;
- перегляд замовлення;
- редагування замовлення (збільшення кількості товарів, зменшення кількості товарів, видалення товарів у замовленні);
- перегляд історії замовлень;
- перегляд товарів;
- пошук товарів за назвою;
- фільтрація за мінімальною та максимальною ціною товару;
- фільтрація за категорією;
- детальний перегляд інформації про характеристики товару;
- оплата товару;
- робота з профілем: створення акаунту та редагування даних профілю;

Адміністратор:

- додання товарів та інформації про них;
- редагування інформації про товари;
- видалення товарів;
- додання категорій товарів;
- редагування категорій товарів;

- видалення категорій товарів.
- пошук товарів за назвою;
- фільтрація за мінімальною та максимальною ціною товару;
- фільтрація за категорією;
- додання сторінок для пунктів меню;
- видалення сторінок для пунктів меню;
- робота з виглядом головної сторінки для користувача;
- робота з бічною частиною сторінки.

Обмеження системи та додаткові можливості:

- головна сторінка не повинна видалятися;
- сторінки редагуються як в коді html так і за допомогою редактору сторінок;
- при видаленні товарів вони не знищуються, а лише помічаються як видалені й надалі ці товари можуть використовуватись у перегляді історії замовлень;
- відсутня можливість обрання ролей адміністраторів, тобто існує стільки адміністраторів скільки буде необхідно замовнику, їх дані не змінюються;
- адміністратори не можуть редагувати інформацію про інших користувачів;
- при видаленні категорій, товари не мають категорії за замовченням, відповідно й відображатись вони не будуть;
- валідація даних та відповідні повідомлення про некоректність введеної інформації;
- розділення сторінок на ролі: тобто певні сторінки відповідають тільки ролі адміністратору, чи користувачу. Користувач не повинен отримати доступ до сторінки адміністратора лише за допомогою коректного для адміністративної зони url, так само адміністратор не може отримати доступ до додання товарів у корзину відповідно й видаляти він їх не може, проте якщо адміністратор захоче переглянути корзину він має на це право, проте необхідно вивести відповідне повідомлення про відсутність товару;

- при виході з акаунту корзина очищується;
- перегляд галереї товарів за допомогою спеціального вікна;
- додання картинок до галереї товару за допомогою перетягування файлів;
- фільтрація файлів картинок, допустимі розширення: .jpg, .jpeg, .gif, .png;
- критерії для фільтрації повинні зберігатися після виконання пошуку.

Для повного опису функціональних вимог, було розроблено діаграми прецедентів (рис. 2.1 – 2.5), які описують таких акторів, як: неавторизований та авторизований користувач, адміністратор в режимі роботи з неадміністративною зоною та в режимі роботи з адміністративною зоною.

Висновки до розділу 2

В цьому розділі було повністю визначено функціональні вимоги до сайту та розроблено діаграми прецедентів, що найбільш повно описують функціональні вимоги за допомогою представлення послідовності дій користувача на сайті в залежності від кінцевої мети.

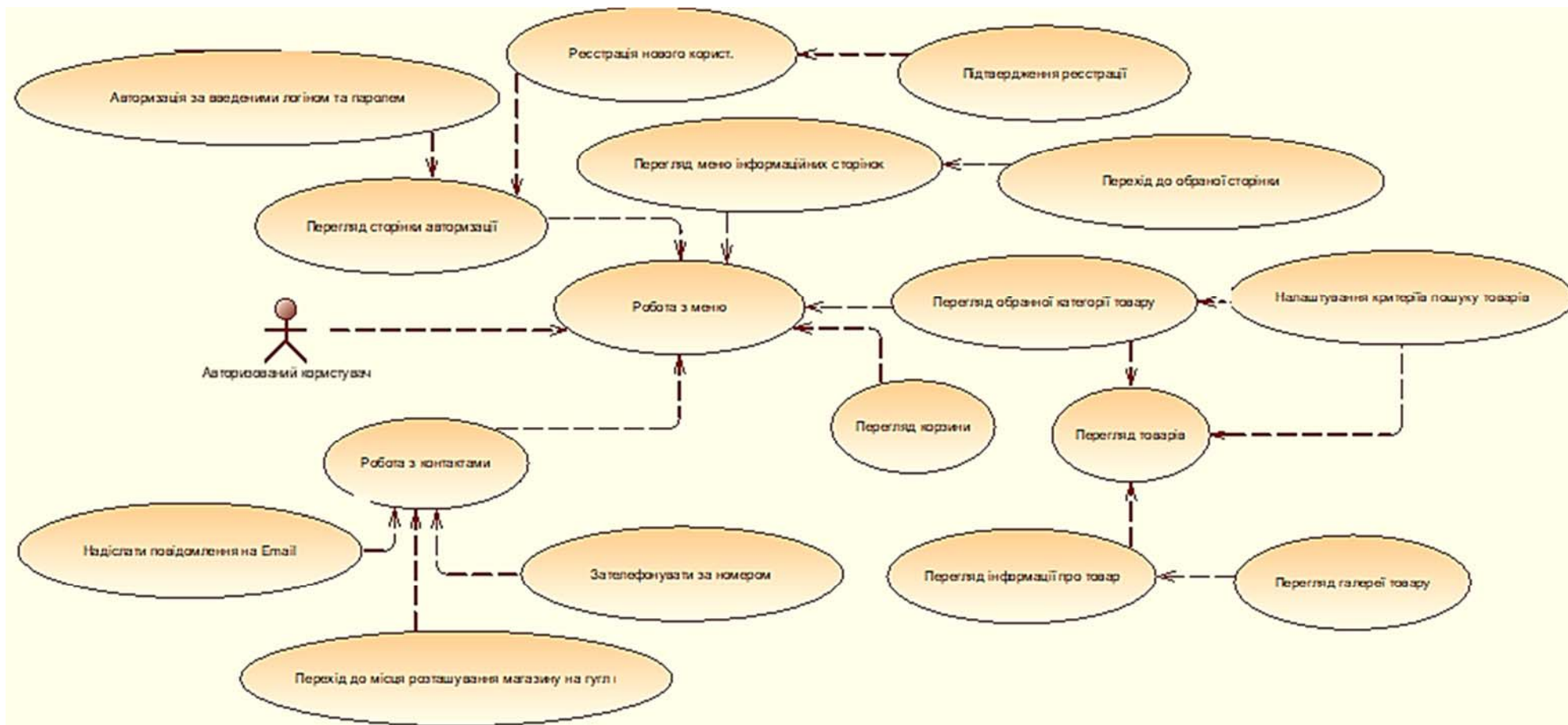


Рисунок 2.1 – Діаграма прецедентів для неавторизованого користувача

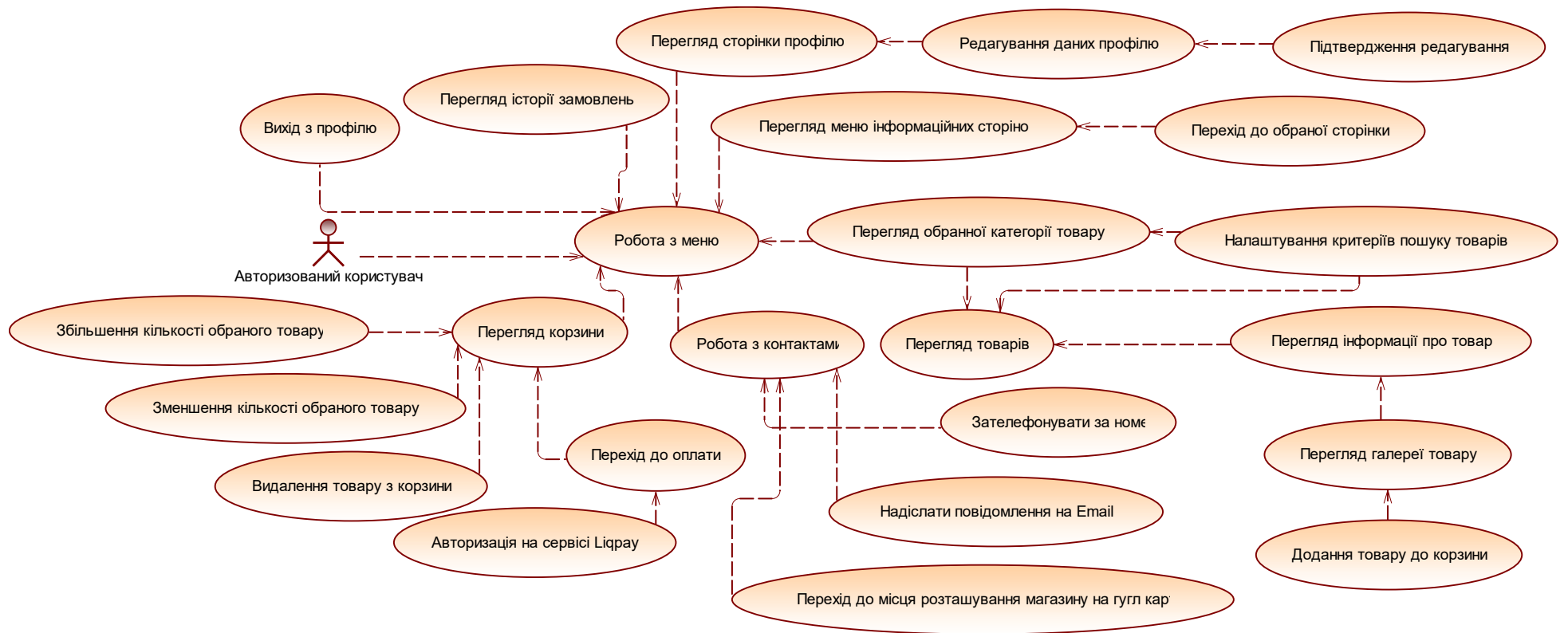


Рисунок 2.2 – Діаграма прецедентів для авторизованого користувача

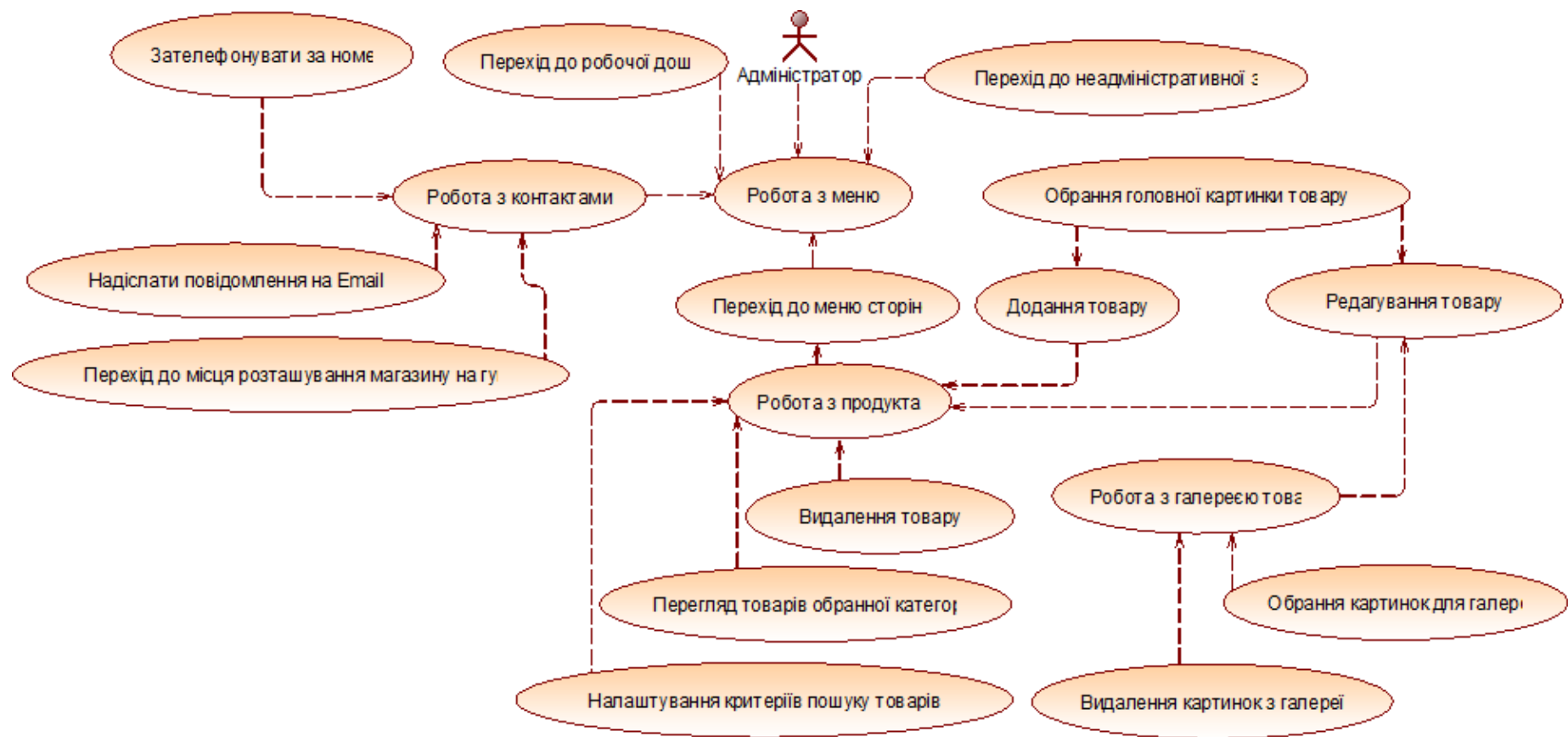


Рисунок 2.4 – Діаграма прецедентів для адміністратора в режимі роботи з адміністративною зоною



Рисунок 2.5 – Діаграма прецедентів для адміністратора в режимі роботи з адміністративною зоною

3 ВНУТРІШНЄ ПРОЄКТУВАННЯ

При побудові веб-застосунку використовується архітектурний патерн MVC.

Використання даного патерну означає, що робота програми міститиме три основні частини:

- моделі - бувають двох видів, перші використовуються для представлення елементів ведення даних з якими працюють користувачі, другі виконують роль посередника та зберігають в собі дані з бази даних;
- представлення - використовуються для відображення даних моделі у вигляді інтерфейсу користувача;
- контролери – виконують обробку подій у представленні, працюють з даними моделі що надходить або заповнюють модель для подальшого відображення у представленні.

Кожна частина такої архітектури має притаманні лише їй обов'язки:

- логіка моделей призначена для роботи на логічному рівні, моделі забезпечують збереження даних користувача (так звані View Model) або дані предметної області (Data transfer object), їх перегляд у представленні та додаткові можливості перевірки їх вмісту чи відображення у представленні (так, наприклад, у C# є спеціальні види анотацій, які задають певні правила відображення полів даних та їх перевірку);
- логіка представлення - дані моделі повинні представляти собою засоби відображення цих даних, тобто інтерфейс користувача;
- логіка контролера, контролер є сполучником між моделями та представленнями, його обов'язок полягає в обробці подій користувача та отриманні необхідних даних для подальшого переходу до представлення, що відповідає отриманій моделі.

Завдяки якісному поділу між частинами додаток буде легше супроводжувати та розширювати протягом його часу існування незалежно від того, наскільки великим він стане [1].

Для наочного відображення структури програми було розроблено діаграми класів (рис. 3.1 – 3.6).

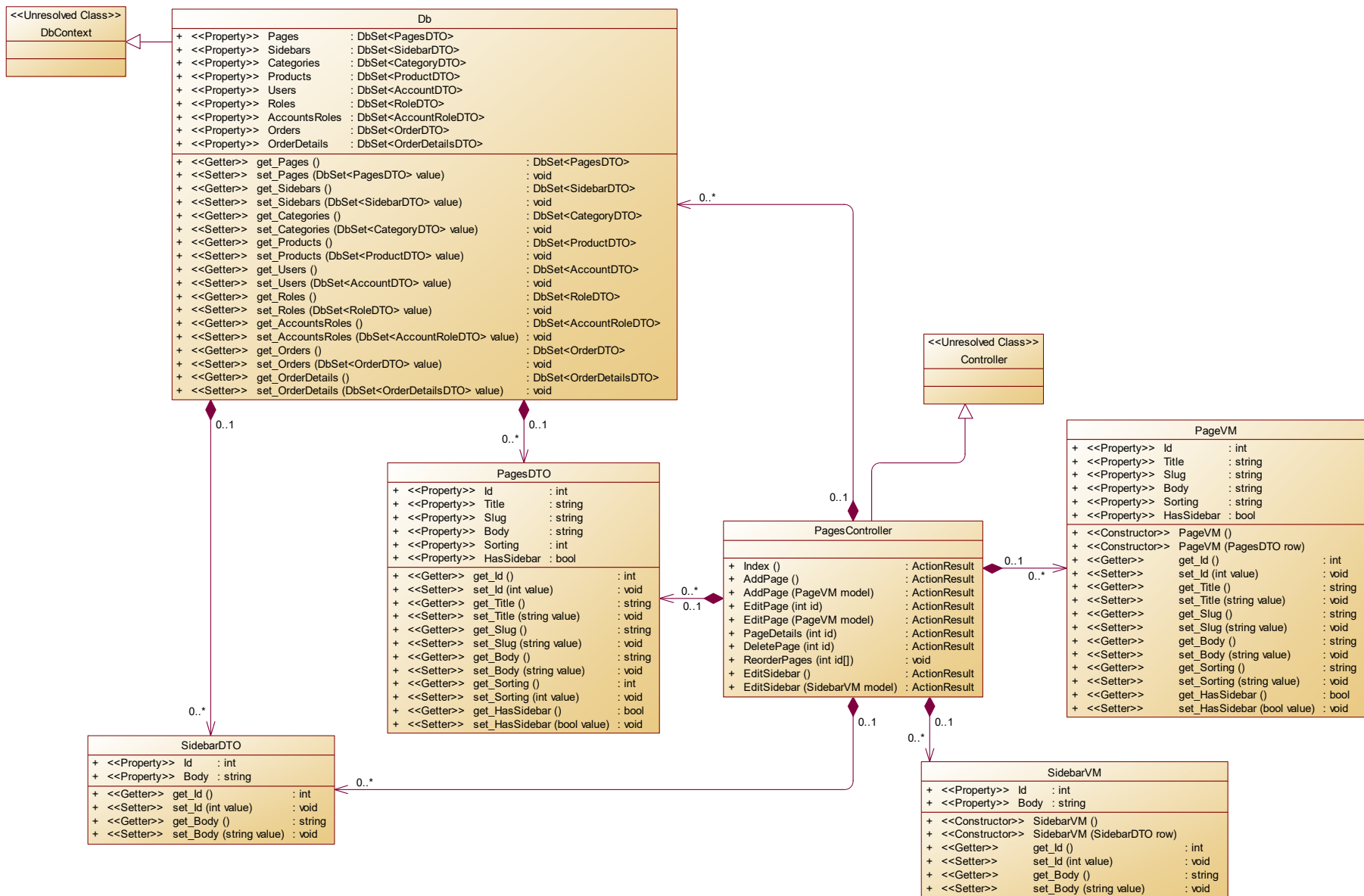


Рисунок 3.1 – Діаграма класів, відносно контролеру сторінок адміністративної зони

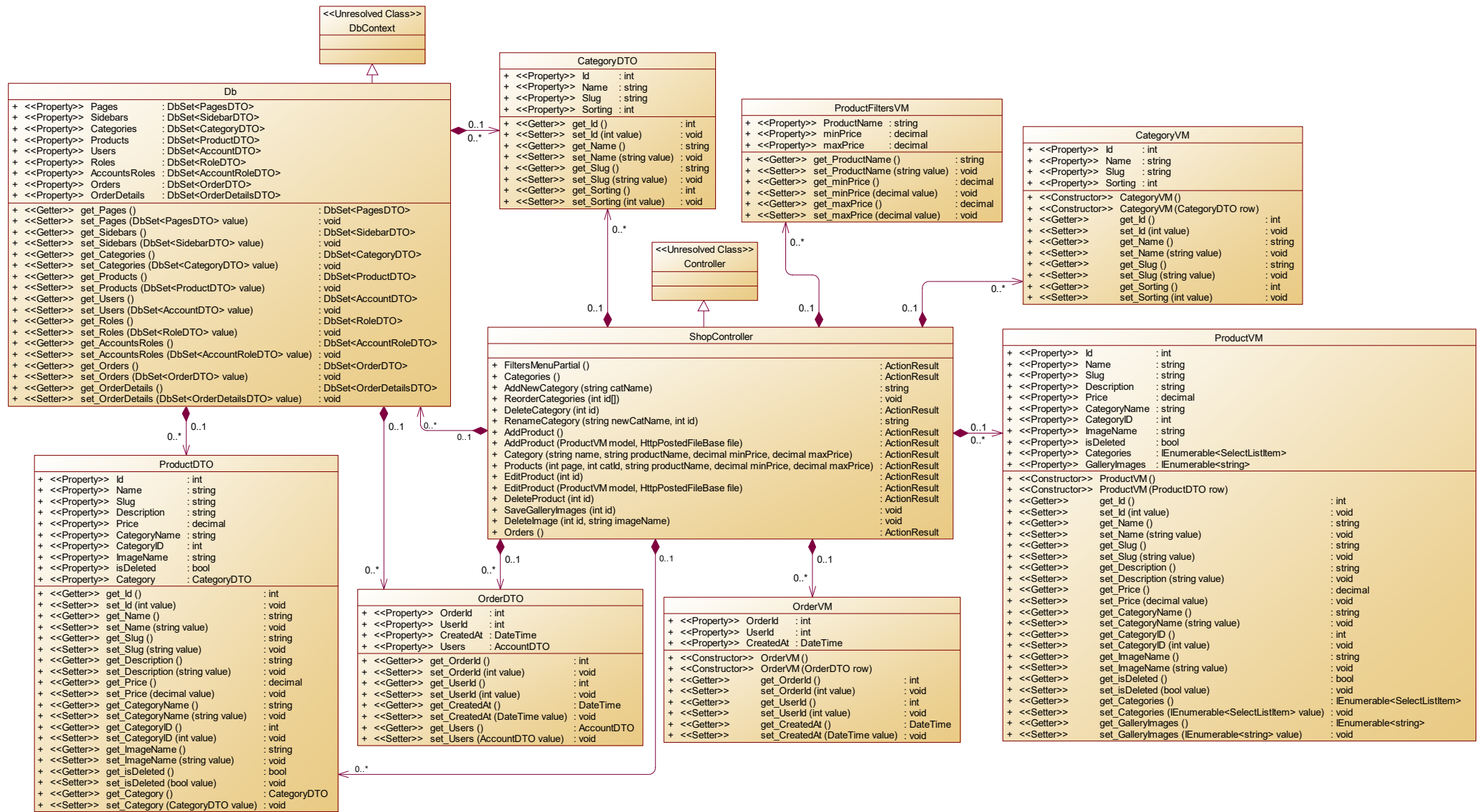


Рисунок 3.2 – Діаграма класів, відносно контролеру продуктів та категорій адміністративної зони

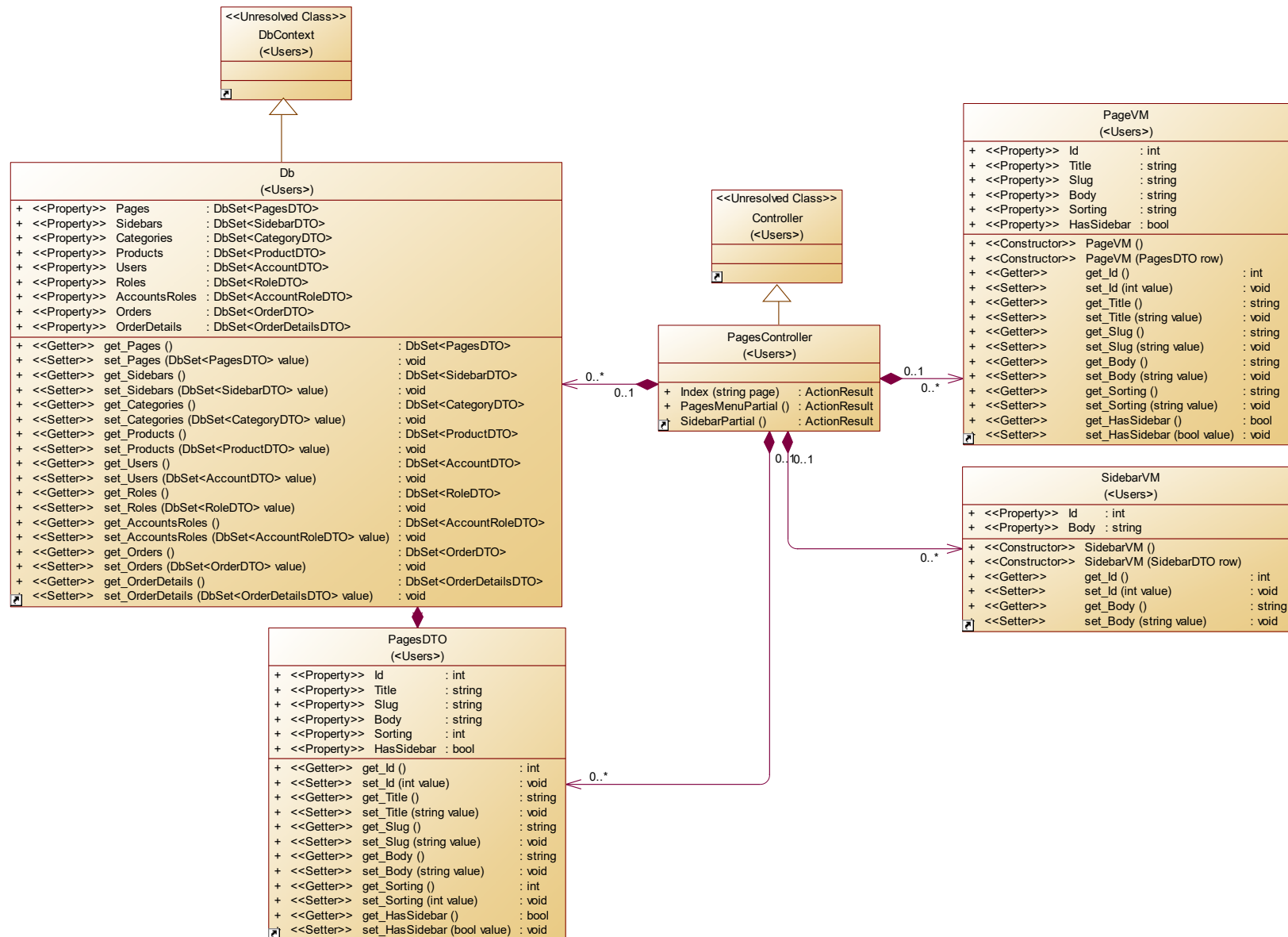


Рисунок 3.4 – Діаграма класів, відносно контролера сторінок користувацької зони

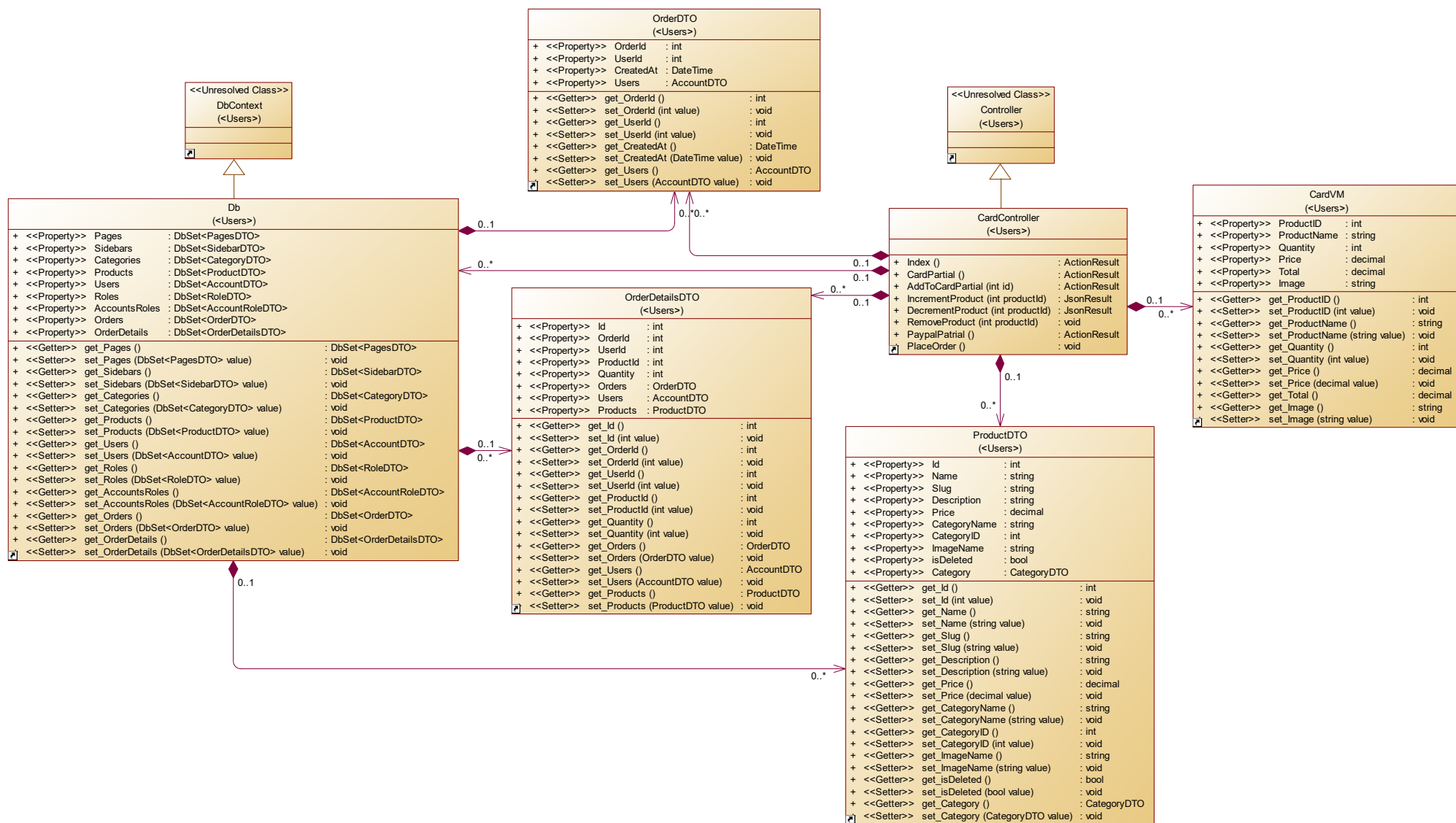


Рисунок 3.5 – Діаграма класів, відносно контролеру кошику товарів користувацької зони

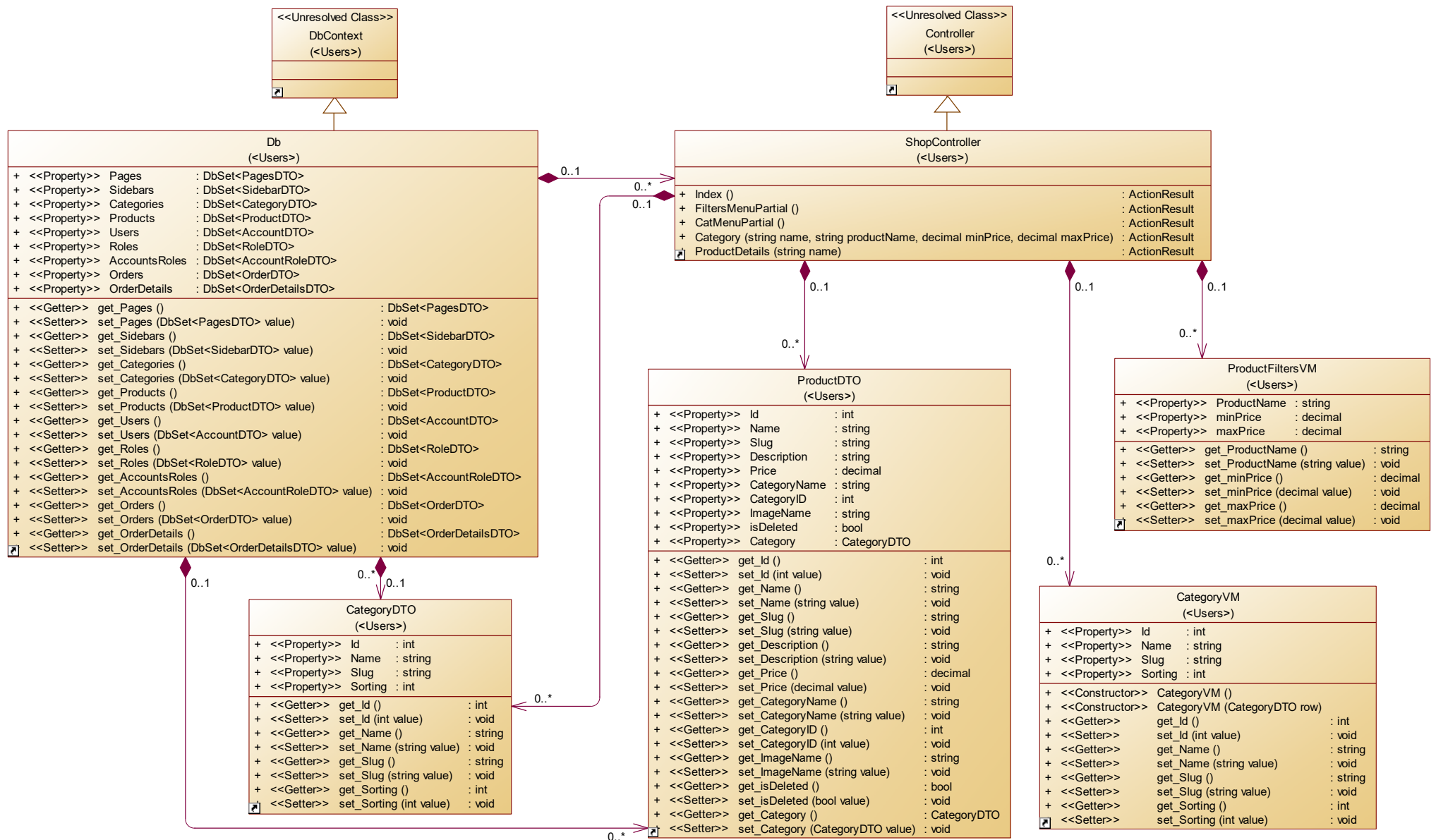


Рисунок 3.6 – Діаграма класів, відносно контролеру продуктів та категорій користувацької зони

Однією з важливих частин роботи додатку є інтерфейс користувача, при проектуванні інтерфейсу необхідно враховувати кольорову палітру елементів, їх розміри та розташування відносно інших елементів та відносно їх важливості для користувача.

Також важливими є засоби акцентування уваги на певних елементах: наявність галереї картинок, що супроводжується переглядом картинок за допомогою анімації, або видалення елементів за допомогою зникнення. Також ці засоби підвищують зручність роботи з додатком та його привабливість.

Для більшого розуміння класів кольорів на рис. 3.7 продемонстровано класи кольорів визначені у bootstrap 4.

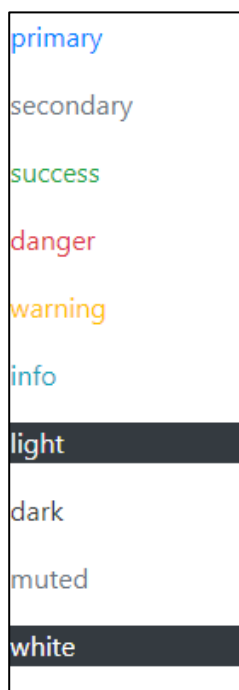


Рисунок 3.7 – Кольори палітри у відповідності до назви класу, які відповідають їх ролям

На основі функціональних вимог та основних елементів притаманних сайтам було сформовано наступні кольорові схеми відносно елементів сайту:

1) меню та підвал сайту:

- фон – #042a74,
- неактивні посилання – #8295ba;
- текстові дані, активні та важливі посилання – кольору #f5f5f5;

- логотип сайту – кольору #f5f5f5;
- кнопка перемикавання режиму для адміністраторів – #fe7700.

2) Блок main:

- інформаційні кнопки – #17a2b8;
- кнопки видалення – #dc3545;
- кнопки додання, створення, підтвердження, пошуку – #28a745;
- інші кнопки - відносяться до класу #efefef;
- фон – #ffffff;
- текст – #343a40.

3) Повідомлення на сторінці:

- інформативні – відносяться до класу .text-info або .text-success (рис. 3.7);
- помилка або дії що потребують додаткової обережності – відносяться до класу .text-danger (рис. 3.7).

4) Групи посилань у вигляді кнопок або списків:

- текст – #000000;
- фон – #f1f1f1.

Обрання виду діалогу з користувачем відіграє важливу роль для обрання найкращого виду взаємодії системи з користувачем, оскільки неправильно підібраний вид діалогу може спричинити незручності при роботі з додатком, що зменшує цінність продукту.

В результаті аналізу вимог було обрано вид діалогу на основі екранних форм, оскільки робота з сайтом допускає на кожному кроці безліч відповідей користувача.

Можливості роботи з програмою найбільш повно були описані у розділі 2 на діаграмах прецедентів (рис. 2.1-2.5), тому додаткового представлення в цьому розділі вони не потребують.

Засоби привернення уваги користувача: для привернення уваги користувача використовується стилістичне оформлення елементів сайту, в залежності від обов'язку елемента вони мають свої кольорові палітри.

При роботі з програмою виникають ситуації коли необхідно вивести інформацію про поточний стан роботи, для цього використовуються текстові повідомлення, що представлені в табл. 3.1.

Таблиця 3.1 – Повідомлення системи

Текст 1	Адресат 2	Ситуація 3	Рекомендовані дії 4
Obligatory field	Користувач	Заповнення обов'язкових полів даних таких як пароль логін та інше	Правильно заповнити відповідне поле
Mail does not match	Користувач	Невірно введено електронну пошту	Правильно заповнити відповідне поле
That title already exist.	Адміністратор	Заповнення даних про сторінку в адміністративній зоні	Ввести назву якої не існує в системі
That slug already exist.	Адміністратор	Заповнення даних про сторінку в адміністративній зоні	Ввести назву якої не існує в системі
The page doesn't exist.	Адміністратор	Редагування неіснуючої сторінки	Повернутись до обрання сторінок
You have deleted a page	Адміністратор	Видалення сторінки в адміністративній зоні	Переглянути
You have edited the sidebar	Адміністратор	Успішне редагування бічної панелі	Переглянути
Title taken	Адміністратор	Спроба зберегти категорію з ім'ям, що вже існує	Ввести назву якої не існує в системі
You have deleted a category	Адміністратор	Успішне видалення категорії	Переглянути
Category has been renamed	Адміністратор	Успішне перейменування категорії	Переглянути
That product name is taken	Адміністратор	Спроба зберегти продукт з ім'ям, що вже існує	Ввести назву якої не існує в системі
Wrong image extension	Адміністратор	При завантаженні файлу було обрано файл, що не є зображенням	Повторити спробу додати файли з одним із розширень .jpg, .jpeg, .gif, .png
That product doesn't exist	Адміністратор	Редагування неіснуючого продукту	Повернутись до обрання продукту

Продовження таблиці 3.1

1	2	3	4
You have deleted a product from gallery	Адміністратор	Успішне видалення продукту з галереї продуктів	Переглянути
Password don't match	Користувач / Адміністратор	Поля пароль та підтвердження паролем містять різні дані	Заповнити поля паролів однаковими значеннями
Login is taken	Користувач	Користувач намагається створити або зберегти акаунт з вже наявним логіном	Повторювати спробу з новим логіном поки не буде введений такий, що не існує в системі
Successfully registered	Користувач	Користувач правильно заповнив всі поля форми реєстрації та підтвердив реєстрацію	Переглянути
Not correct login or password	Користувач / Адміністратор	Неправильно заповнено поля форми авторизації та була спроба увійти у систему	Ввести правильні дані для авторизації або зателефонувати за посиланням в розділі контактів внизу сайту для отримання консультації чи відновлення паролю
Password is taken	Користувач	Користувач намагається змінити пароль на пароль що зараз є в системі	Ввести пароль, що відрізняється від поточного
You have edited your information	Користувач	Користувач правильно заповнив поля форми профілю та підтвердив збереження	Переглянути
Your shop card is empty	Користувач	Користувач перейшов у порожню корзину товарів	Переглянути або додати товар до корзини
Category name must be at least 3 characters long	Адміністратор	Адміністратор намагається створити категорію, назва якої менше 3-х символів	Ввести назву кількості символів якої більше 3-х

Продовження таблиці 3.1

1	2	3	4
You have added a product	Адміністратор	Успішне додання товару	Переглянути
The category has been added	Адміністратор	Успішне додання категорії	Переглянути
Confirm product deletion	Адміністратор	Підтвердження видалення продукту	Підтвердити видалення або скасувати
You will be redirected to paypal now	Користувач	Оплата товару, перехід до сторінки оплати	Переглянути повідомлення, очікувати завантаження відповідної форми авторизації, авторизуватись та сплатити
That product has been added to your card.		Успішне додання товару до кошика	Продовжити покупки або перейти до кошика для редагування кількості товару

Перелік браузерів на яких повинні правильно працювати усі функції сайту:

- Google Chrome v.101.0.4951.67+;
- Opera v.86.0.4363.59+;
- Microsoft Edge v.101.0.1210.53+.

Оскільки всі частини додатку було розроблено за моделлю MVC, динаміка системи є схожою для всіх модулів, тому детального проєктування з точки зору динаміки кожний модуль не потребує, тому для наочного представлення роботи системи буде описано роботу системи на основі одного з прецедентів, а саме, відображення товарів використовуючи фільтрацію даних.

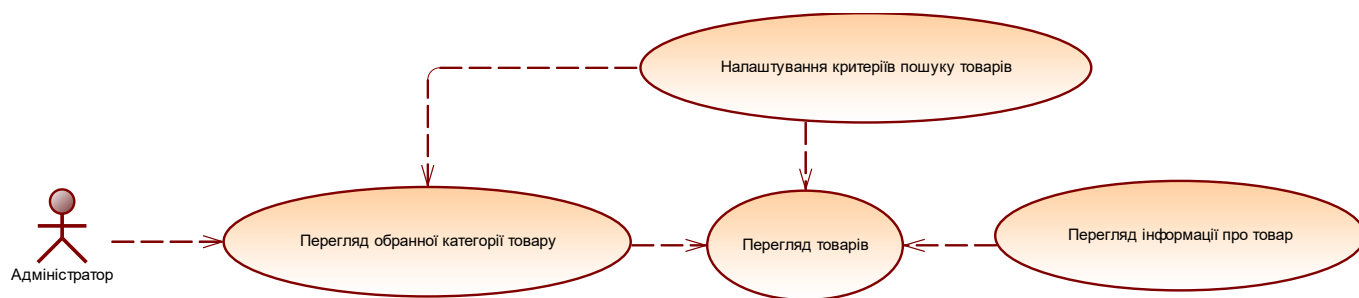


Рисунок 3.8 – Діаграма прецедентів на основі якої буде показано загально динаміку системи

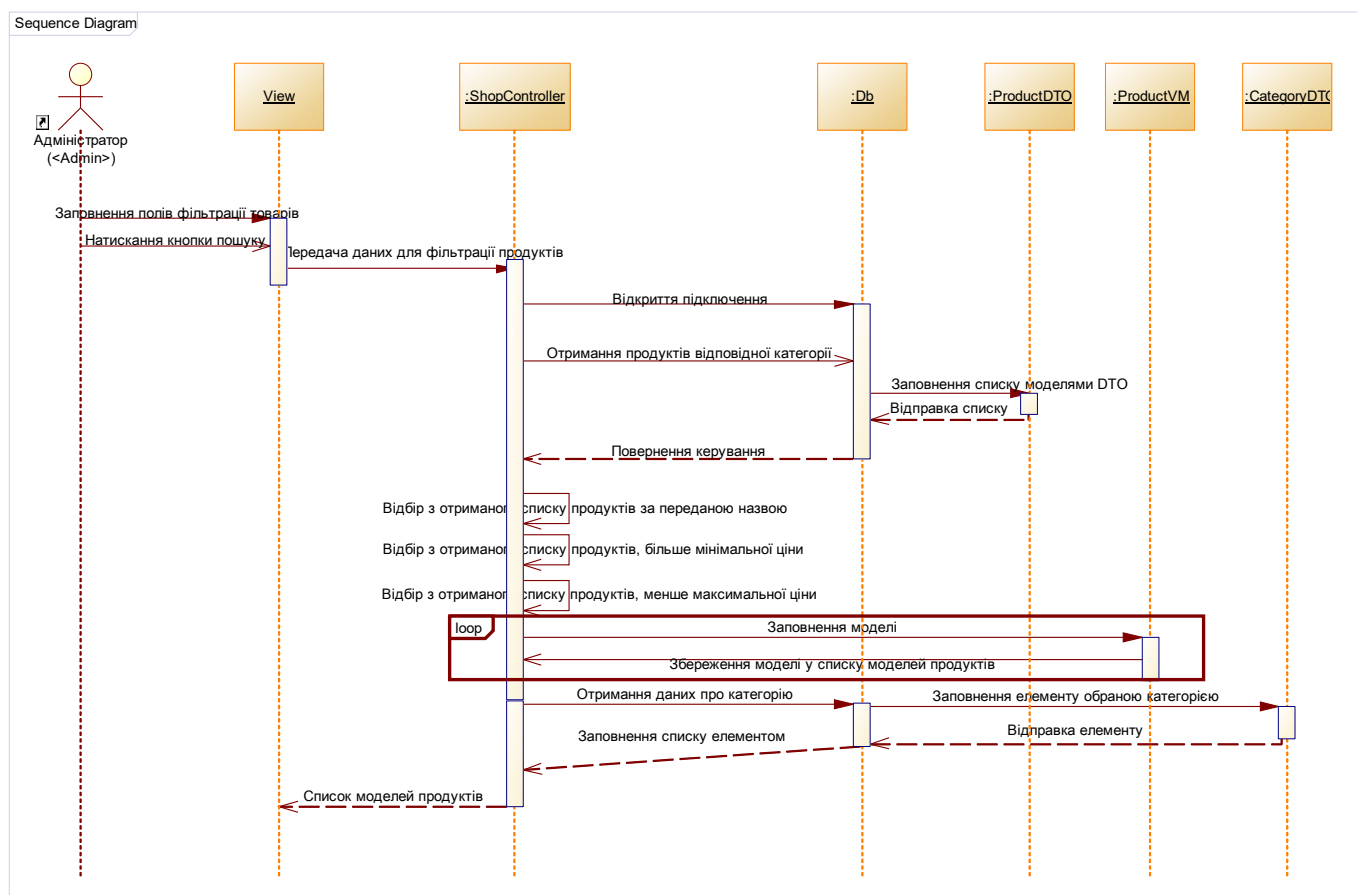


Рисунок 3.9 – Діаграма послідовності дій, що описує динаміку системи при виконанні прецедентів вказаних на рис. 3.8

Висновки до розділу 3

В цьому розділі було описано внутрішню структуру проекту та схему взаємодії класів між собою (рис. 3.8).

В розділі представлено кольорову палітру для різних компонентів сайту в залежності від їх ролі та ступеня привернення уваги користувача.

Було проаналізовано варіанти діалогу з користувачем й обрано найбільш доречний для сайту.

Також в розділі описуються підтримуванні браузерери та їх версії.

4 ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

Для наочного відображення фізичної структури проекту виконаємо проектування діаграми компонентів (рис. 4.1 – 4.7).

Оскільки сайт міститиме багато компонентів, діаграми будуть розроблені відносно контролерів сторінок, це дає змогу детально проаналізувати залежності між файлами та дотримуватись принципу MVC, оскільки компоненти будуть розташовуватись згідно рівні патерну. Для наочного відображення різниці між файлами позначимо різними кольорами:

- контролери - темно-червоним кольором;
- представлення - світло-зеленим;
- компоненти які напряду пов'язані зі збереженням даних – світло-помаранчевим. До таких компонентів відносяться: моделі представлення (VM), моделі збереження даних рядку таблиці (DTO) та компонент, що забезпечує роботу з базами даних (DB).

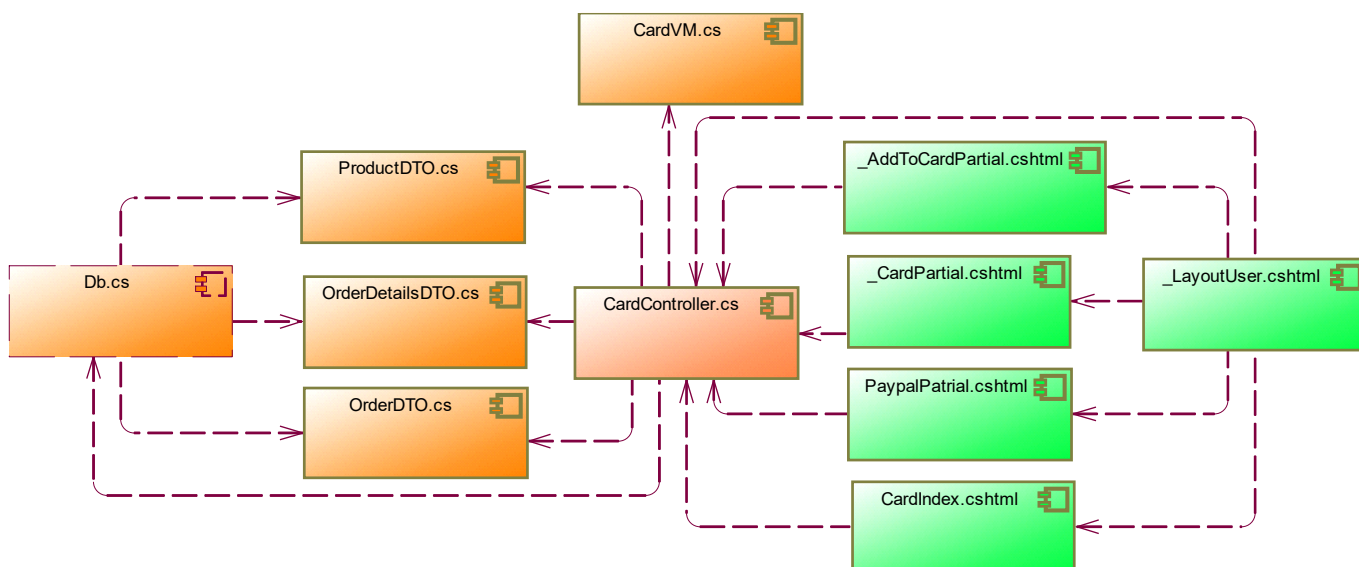


Рисунок 4.1 – Діаграма артефактів (компонентів) відносно контролеру списку покупок користувацької частини

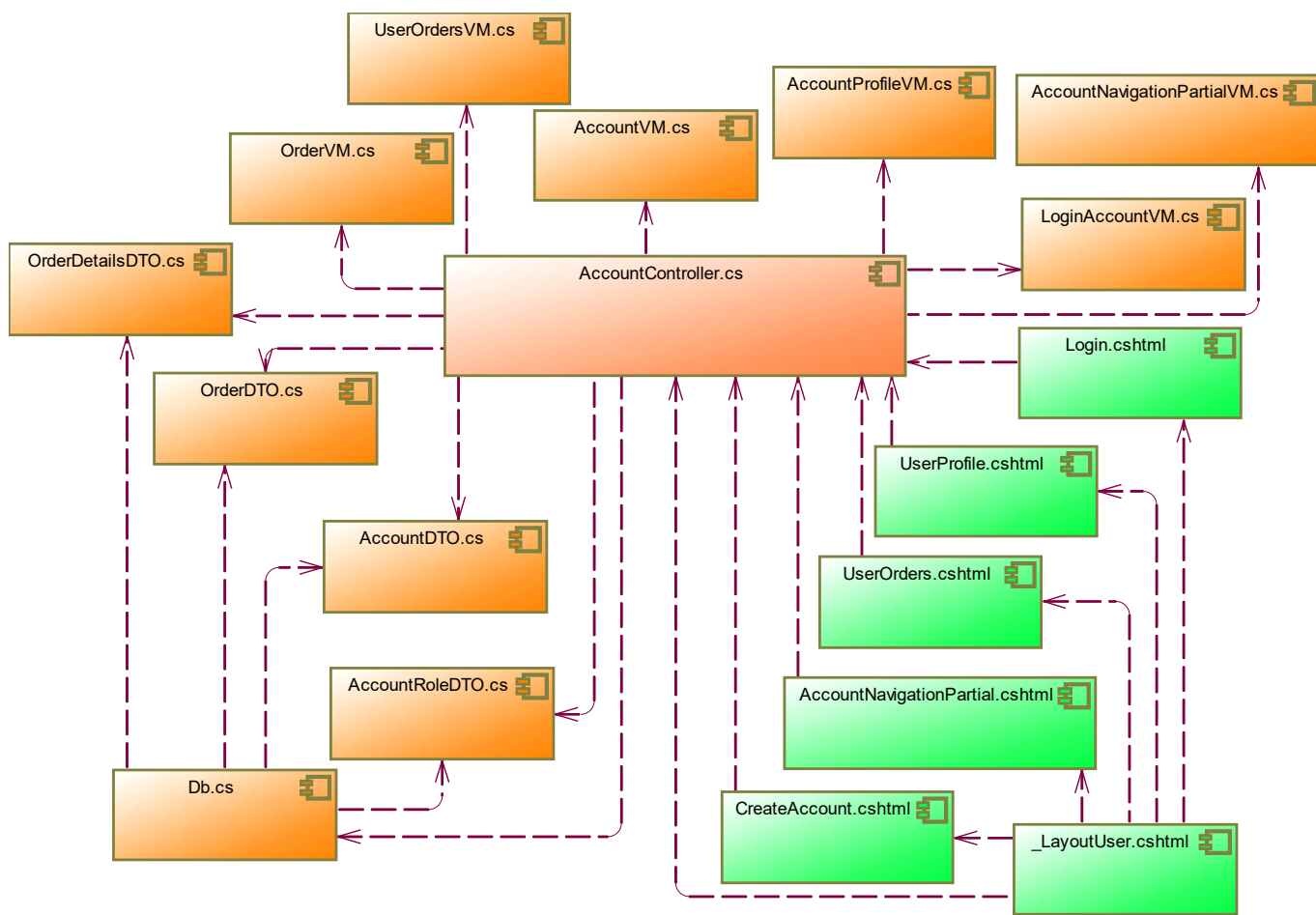


Рисунок 4.2 – Діаграма артефактів (компонентів) відносно контролеру акаунтів користувачької частини

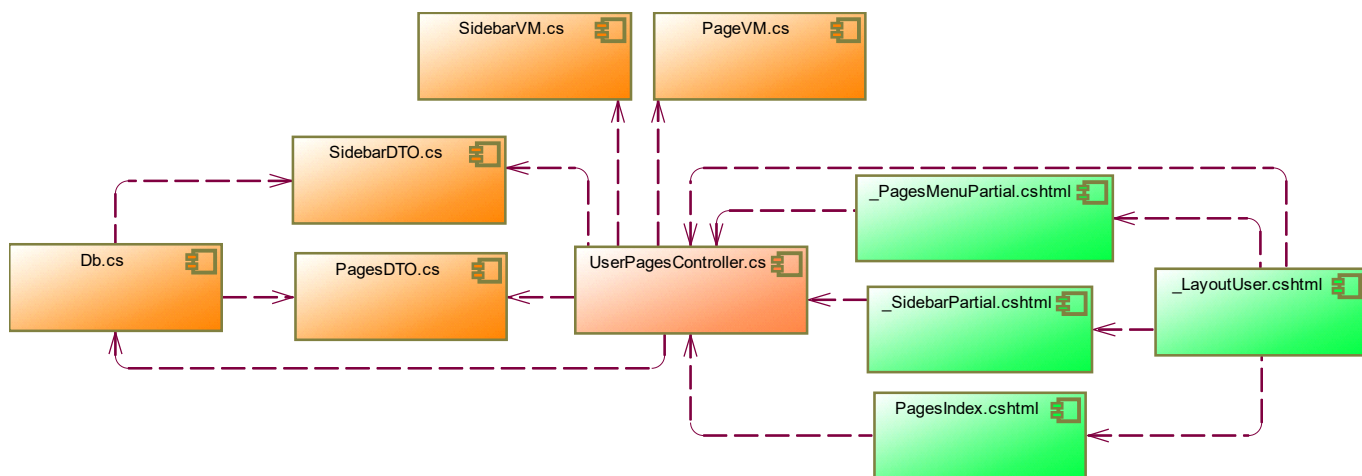


Рисунок 4.3 – Діаграма артефактів (компонентів) відносно контролеру сторінок користувачької частини

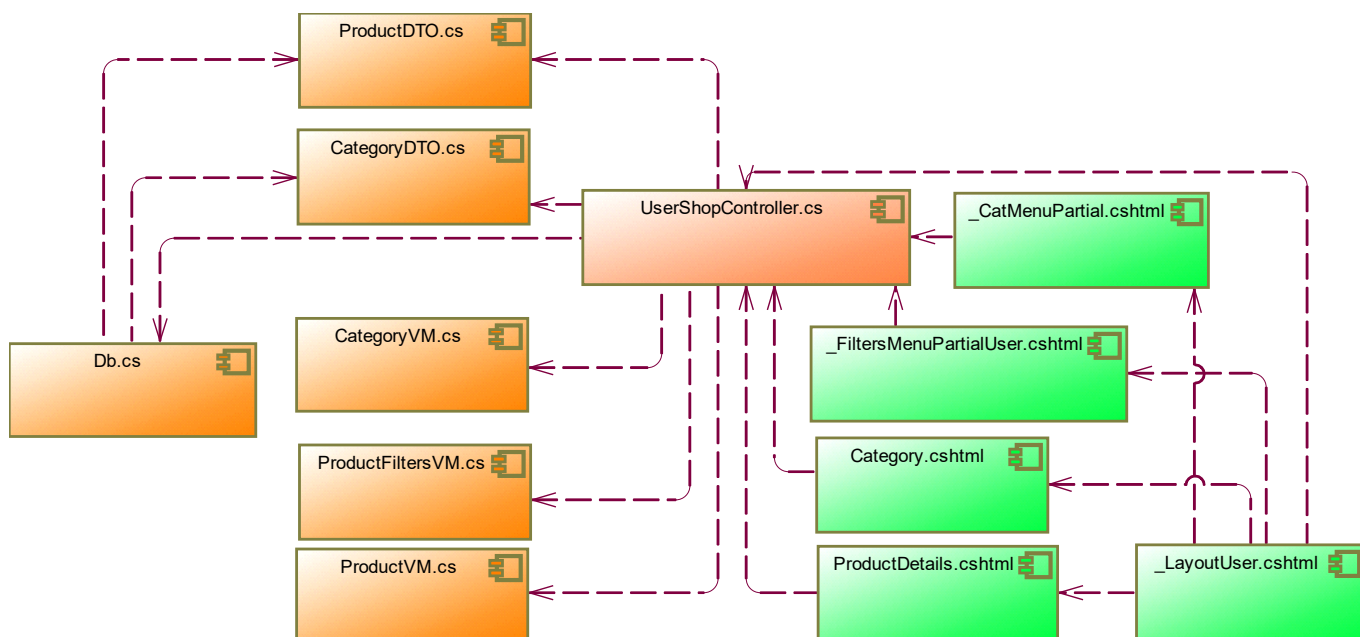


Рисунок 4.4 – Діаграма артефактів (компонентів) відносно контролеру роботи над покупками, товарами та категоріями товарів, для користувацької частини

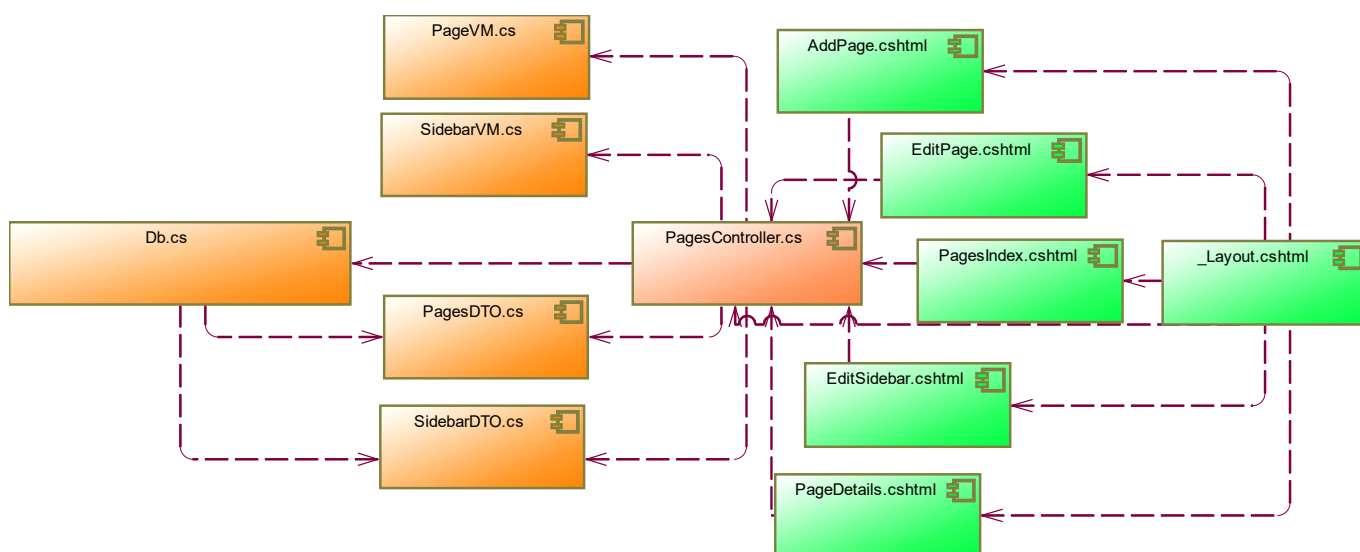


Рисунок 4.5 – Діаграма артефактів (компонентів) відносно контролеру сторінок адміністративної частини

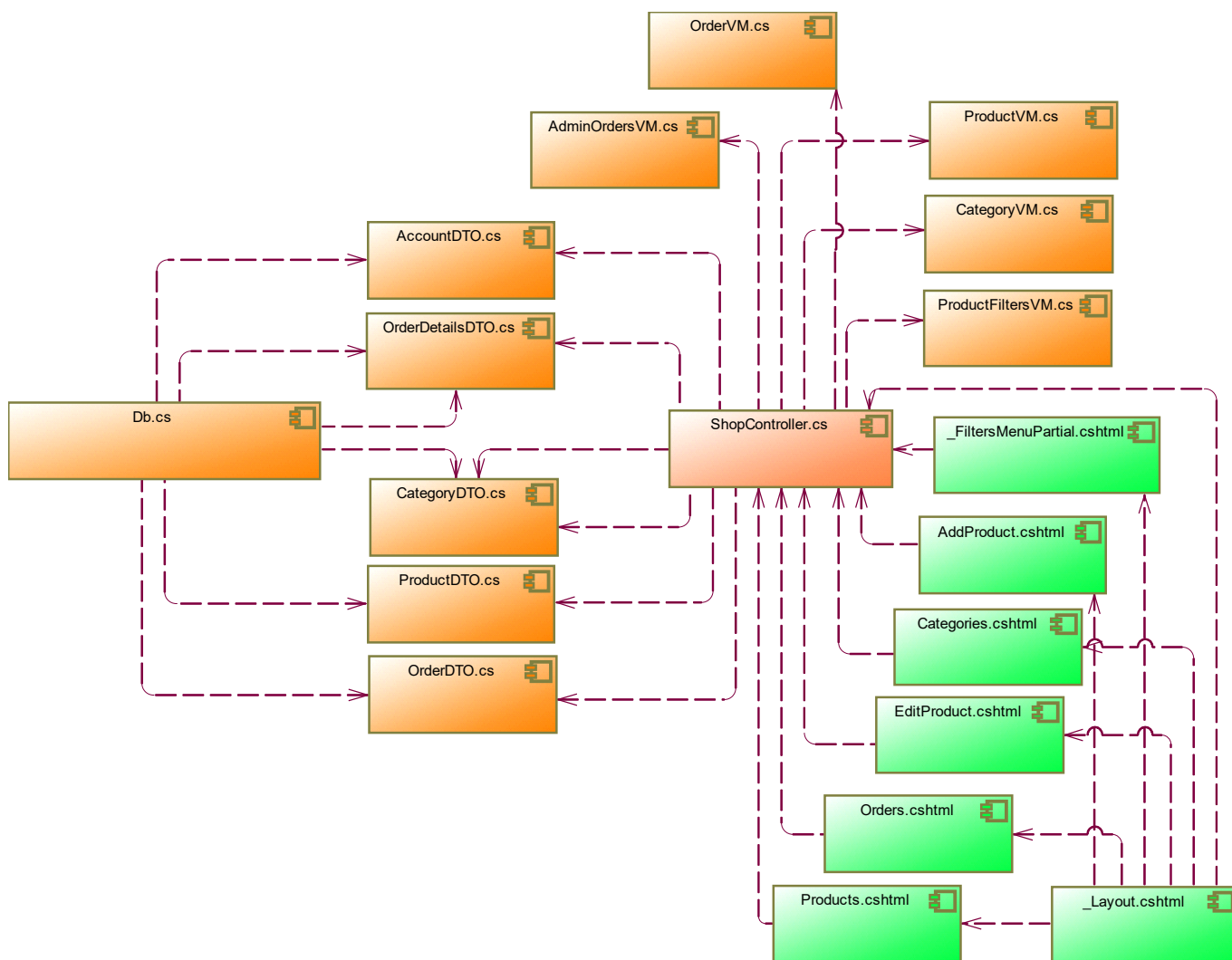


Рисунок 4.6 – Діаграма артефактів (компонентів) відносно контролеру роботи над покупками, товарами та категоріями товарів, для адміністративної частини

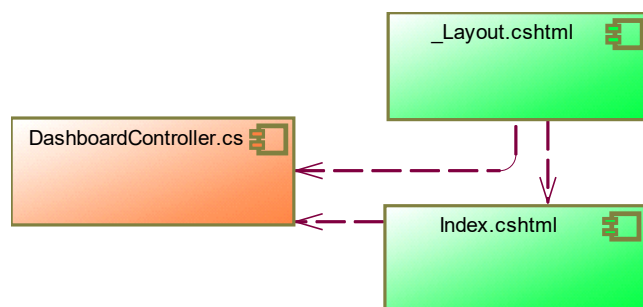


Рисунок 4.7 – Діаграма артефактів (компонентів) відносно контролеру головної сторінки адміністративної частини

В розробці програмного забезпечення важливим є детальне проектування бази даних, оскільки відносно її структури надалі будуть будуватись інші компоненти системи методи та засоби їх взаємодії. На рис 4.8 представлено ER-модель розробленої бази даних.

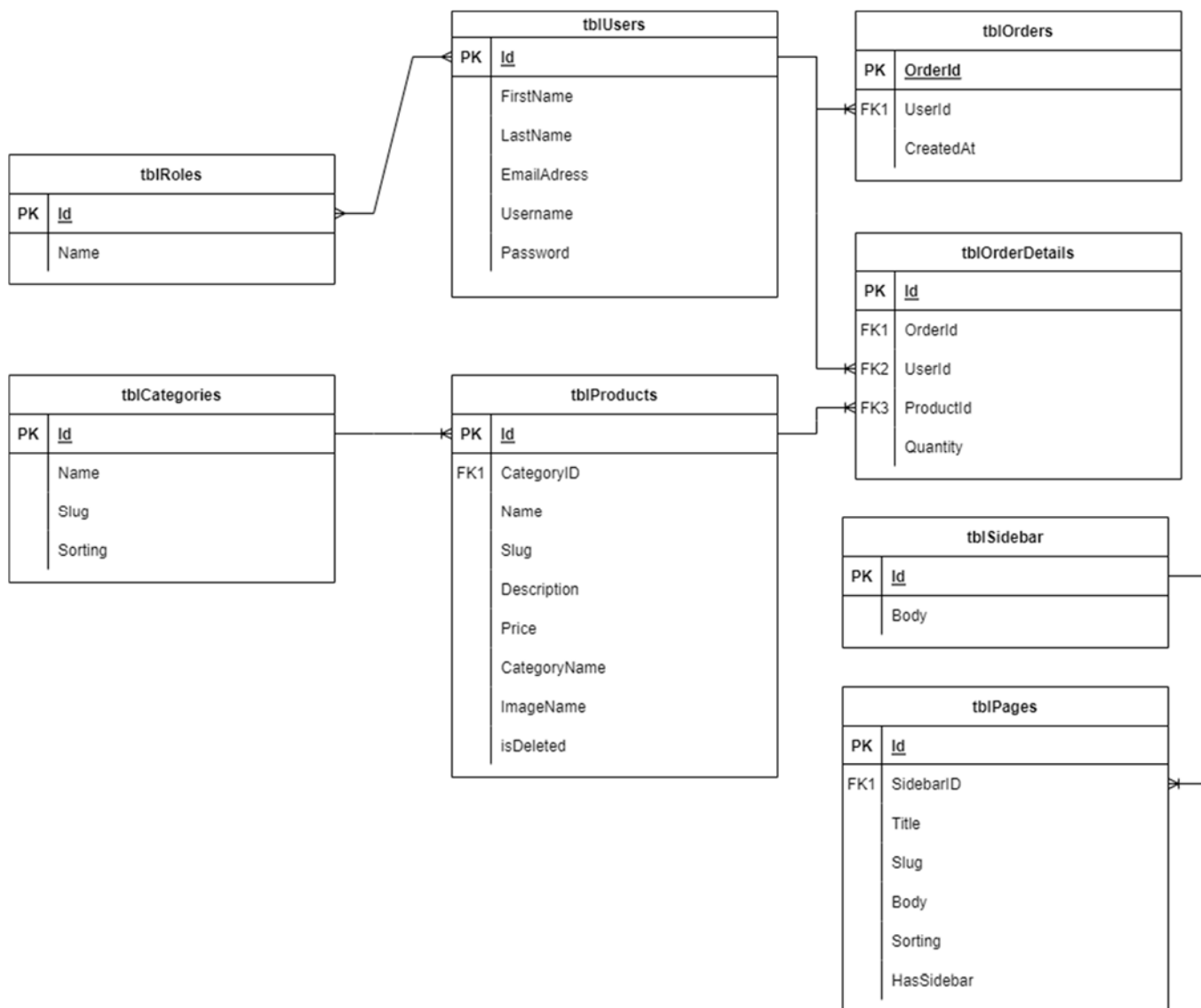


Рисунок 4.8 – ER-модель розробленої бази даних

Для роботи з базою даних програма використовує систему управління реляційними базами даних – Microsoft SQL Server, після розробки ER-моделі в середовищі Visual studio було розроблено відповідні таблиці бази даних (табл. 4.1 – 4.9).

Таблиця 4.1 – Відношення tblSidebar

№	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	Id	Унікальний код бічної панелі	Int	4	+
2.	Body	Тіло бічної панелі, представляє собою html код	ntext	2147483646	-

Таблиця 4.2 – Відношення tblCategories

№	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	Id	Унікальний код категорії	int identity	4	+
2.	Name	Назва категорії	nvarchar	100	-
3.	Slug	Коротка назва категорії, позначає назву, що використовується в URL сторінки	nvarchar	100	-
4.	Sorting	Число, що позначає номер по порядку у списку категорій	int	4	-

Таблиця 4.3 – Відношення tblProducts

№	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1	2	3	4	5	6
1.	Id	Унікальний код продукту	int identity	4	+
2.	Name	Назва продукту	nvarchar	100	-
3.	Slug	Коротка назва продукту, позначає назву, що використовується в URL сторінки	nvarchar	100	-
4.	Description	Опис товару	ntext	2147483646	-
5.	Price	Ціна	numeric	20	-
6.	CategoryName	Ім'я категорії до якої відноситься товар (використовується для зменшення кількості запитів на сервер)	nvarchar	100	-
7.	CategoryID	Унікальний код категорії	int	4	-

Продовження таблиці 4.3

1	2	3	4	5	6
8.	ImageName	Шлях до картинки товару	nvarchar	200	-
9.	isDeleted	Логічне значення, що позначає статус видалення продукту, це поле використовується для відображення продуктів користувачу, якщо поле рівне true, то товар не відображається в каталозі товарів і навпаки якщо false - відображається	bit	1	-

Таблиця 4.4 – Відношення tblRoles

	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	Id	Унікальний код ролі	int identity	4	+
2.	Name	Назва ролі	nvarchar	100	+

Таблиця 4.5 – Відношення tblUserRoles

	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	UserId	Унікальний код користувача	int	4	+
2.	RoleId	Унікальний код ролі	int	4	-

Таблиця 4.6 – Відношення tblOrders

	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	OrderId	Унікальний код замовлення	int identity	4	+
2.	UserID	Унікальний код користувача, що зробив замовлення	int	4	-
3.	CreatedAt	Дата та час створення замовлення	datetime2	54	-

Таблиця 4.7 – Відношення tblOrderDetails

	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	Id	Унікальний код запису про деталі замовлення	int identity	4	+
2.	OrderId	Унікальний код замовлення	int	4	-
3.	UserId	Унікальний код користувача, що зробив замовлення	int	4	-
4.	ProductId	Унікальний код продукту доданого в замовлення	int	4	-
5.	Quantity	Кількість продукту замовлення	int	4	-

Таблиця 4.8 – Відношення tblUsers

	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	Id	Унікальний код користувача	int identity	4	+
2.	FirstName	Ім'я користувача	nvarchar	100	-
3.	LastName	Прізвище користувача	nvarchar	100	-
4.	EmailAdresses	Електронна пошта користувача	nvarchar	100	-
5.	Username	Логін	nvarchar	100	-
6.	Password	Пароль	nvarchar	100	-

Таблиця 4.9 – Відношення tblPages

	Стовпець	Опис поля	Тип даних	Розмір	Ключ
1.	Id	Унікальний код сторінки	int identity	4	+
2.	Title	Заголовок сторінки, що відображається на вкладці у браузері	nvarchar	100	-
3.	Slug	Коротка назва сторінки, позначає назву, що використовується в URL сторінки	nvarchar	100	-
4.	Body	Тіло сторінки	ntext	2147483646	-
5.	Sorting	Число, що позначає номер по порядку у списку сторінок	int	4	-
6.	HasSidebar	Логічне значення, що означає наявність бічної панелі у сторінки	bit	1	-
7.	SidebarID	Унікальний код бічної сторінки	int	4	-

Висновки до розділу 4

В цьому розділі була описана взаємодія компонентів на фізичному рівні, побудовано структури їх взаємодії на різних рівнях проекту, основними елементами виступали контролери. На діаграмах компонентів за допомогою різних кольорів представлено поділ модулів відповідно до їх ролей.

Також було спроектовано структуру відношень бази даних. Відношення спроектовані відносно поточної ітерації проекту й можуть покращуватись та розширюватись на інших ітераціях.

5 РОЗРОБКА ПРОГРАМИ

Для розробки програми було обрано мову C# та фреймворк ASP.Net MVC.

Обраний фреймворк незалежний від системи управління базою даних, що дає змогу легко обрати необхідний ресурс для розміщення сайту.

При побудові додатку використовується мова HTML-розмітки Razor, що дозволяє гнучко будувати сторінки сайту з будь-яким набором даних.

Мова C# є компільованою мовою, що дає помітний приріст в продуктивності сайту, вона має автоматичне прибирання сміття. Ця можливість дозволяє не думати про алокацію пам'яті в більшості завдань та надає змогу зосередитися на інших аспектах розробки системи.

Також сайти на ASP.Net MVC легко розміщаються на рішеннях в Azure або IIS. За бажанням розміщення сайту можна зробити прямо з Visual Studio у два кліки і таким чином не витратити час на розгортання.

Програма розробляється методом покрокової деталізації, тобто програма розробляється зверху вниз.

Покрокова деталізація представлена у вигляді списку.

1. Розподілення зон.

1.1. Зона адміністратора призначена для налаштування сторінок, товарів та категорій.

1.1.1. Частина сайту згідно процесу розробки через фреймворк з використанням Razor.

1.1.1.1. Блок Header.

1.1.1.1.1. Перехід до перегляду зони користувача у вигляді демонстрації.

1.1.1.1.2. Головна сторінка.

1.1.1.1.3. Сторінка роботи з сторінками сайту.

1.1.1.1.4. Сторінка роботи з бічною панеллю.

1.1.1.1.5. Сторінка роботи з категоріями.

1.1.1.1.6. Сторінка роботи з продуктами.

1.1.1.1.7. Сторінка роботи замовленнями.

1.1.1.2. Блок Main.

1.1.1.2.1. Головна сторінка.

1.1.1.2.1.1. Вступна інформація для адміністраторів.

1.1.1.2.2. Сторінка роботи з сторінками сайту.

1.1.1.2.2.1. Сторінка додання.

1.1.1.2.2.2. Сторінка редагування.

1.1.1.2.2.3. Видалення.

1.1.1.2.2.4. Сорткування сторінок.

1.1.1.2.3. Сторінка роботи з бічною панеллю.

1.1.1.2.3.1. Редагування бічної панелі.

1.1.1.2.4. Сторінка роботи з категоріями.

1.1.1.2.4.1. Додання.

1.1.1.2.4.2. Редагування.

1.1.1.2.4.3. Видалення.

1.1.1.2.4.4. Сорткування категорій.

1.1.1.2.5. Сторінка роботи з продуктами.

1.1.1.2.5.1. Сторінка додання, окрім основних полів, функція додання заставки товару.

1.1.1.2.5.2. Сторінка редагування, окрім основних полів, функція додання заставки товару.

1.1.1.2.5.3. Видалення.

1.1.1.2.5.4. Блок фільтрації даних.

1.1.1.2.5.4.1. За назвою категорії.

1.1.1.2.5.4.2. За назвою продукту.

1.1.1.2.5.4.3. За ціною.

1.1.1.2.6. Сторінка роботи замовленнями.

1.1.1.2.6.1. Перегляд замовлень користувачів.

1.1.1.3. Блок footer.

1.1.1.3.1. Контактні дані.

1.2. Зона адміністратора для перегляду даних призначених для покупця.

1.2.1. Частина сайту згідно процесу розробки через фреймворк з

використанням Razor.

1.2.1.1. Блок Header.

1.2.1.1.1. Головна сторінка.

1.2.1.1.2. Меню сторінок для користувача (випадаючий список).

1.2.1.1.3. Вихід з акаунту.

1.2.1.1.4. Перехід до зони адміністратора.

1.2.1.2. Блок Main.

1.2.1.2.1. Головна сторінка.

1.2.1.2.2. Перегляд інформативних сторінок.

1.2.1.2.3. Перегляд категорії.

1.2.1.2.4. Перегляд товарів.

1.2.1.2.4.1. Перегляд інформації про товар.

1.2.1.2.4.1.1. Перегляд галереї.

1.2.1.2.5. Блок фільтрації даних.

1.2.1.2.5.1. За назвою категорії.

1.2.1.2.5.2. За назвою продукту.

1.2.1.2.5.3. За ціною.

1.2.1.3. Блок footer.

1.2.1.3.1. Контактні дані.

1.3. Зона авторизованого Покупця.

1.3.1. Частина сайту згідно процесу розробки через фреймворк з використанням Razor.

1.3.1.1.1. Блок Header.

1.3.1.1.1.1. Головна сторінка.

1.3.1.1.1.2. Інформаційні сторінки.

1.3.1.1.1.3. Вихід.

1.3.1.1.1.4. Редагування профілю.

1.3.1.1.1.5. Історія замовлень.

1.3.1.1.1.6. Корзина.

1.3.1.1.2. Блок Main.

- 1.3.1.1.2.1. Головна сторінка.
- 1.3.1.1.2.2. Перегляд інформативних сторінок.
- 1.3.1.1.2.3. Перегляд категорії.
- 1.3.1.1.2.4. Перегляд корзини.
 - 1.3.1.1.2.4.1. Збільшення кількості товару.
 - 1.3.1.1.2.4.2. Зменшення кількості товару.
 - 1.3.1.1.2.4.3. Видалення товару.
 - 1.3.1.1.2.4.4. Оплата замовлення.
- 1.3.1.1.2.5. Перегляд товарів.
 - 1.3.1.1.2.5.1. Перегляд інформації про товар.
 - 1.3.1.1.2.5.1.1. Перегляд галереї.
 - 1.3.1.1.2.5.1.2. Додання до корзини.
- 1.3.1.1.2.6. Блок фільтрації даних.
 - 1.3.1.1.2.6.1. За назвою категорії.
 - 1.3.1.1.2.6.2. За назвою продукту.
 - 1.3.1.1.2.6.3. За ціною.
- 1.3.1.1.3. Блок footer.
 - 1.3.1.1.3.1. Контакти.

1.4. Зона неавторизованого Покупця.

- 1.4.1. Частина сайту згідно процесу розробки через фреймворк з використанням Razor.
 - 1.4.1.1.1. Блок Header.
 - 1.4.1.1.1.1. Головна сторінка.
 - 1.4.1.1.1.2. Інформаційні сторінки.
 - 1.4.1.1.1.3. Авторизація.
 - 1.4.1.1.2. Блок Main.
 - 1.4.1.1.2.1. Головна сторінка.
 - 1.4.1.1.2.2. Перегляд інформативних сторінок.
 - 1.4.1.1.2.3. Авторизація.
 - 1.4.1.1.2.4. Реєстрація.

1.4.1.1.2.5. Перегляд категорії.

1.4.1.1.2.6. Перегляд товарів.

1.4.1.1.2.6.1. Перегляд інформації про товар.

1.4.1.1.2.6.1.1. Перегляд галереї.

1.4.1.1.2.7. Блок фільтрації даних.

1.4.1.1.2.7.1. За назвою категорії.

1.4.1.1.2.7.2. За назвою продукту.

1.4.1.1.2.7.3. За ціною.

1.4.1.1.3. Блок footer.

1.4.1.1.3.1. Контакти.

Алгоритм роботи для кожного модулю схожий. Рівень представлення надає користувачу дії, які він може виконати з додатком, для заповнення полів він використовує моделі представлення, які відображають ці поля. Для обробки подій кожне представлення посилається на відповідний контролер, який виконує обробку введеної інформації й надалі при необхідності передає або отримує дані що зберігаються в базі даних через об'єкти передачі даних, які представляють собою моделі розроблених в базі даних таблиць.

Також для авторизації користувача використовується фреймворк ASP.Net Identity, який виконує більшу частину логіки по обробці даних користувача для входу у систему та надання користувачу можливостей роботи з нею в залежності від ролі.

Висновки до розділу 5

В цьому розділі було відображено рівні деталізації проекту, розбиття кожної окремої задачі на під задачі, було обґрунтовано обрання мови програмування та фреймворків для розробки сайту й описано загальний алгоритм роботи модулів спроектованих за архітектурним патерном MVC.

6 ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ

При тестуванні програми доцільно використати метод тестування чорної скриньки — припущення про помилку, оскільки проект використовує додаткові бібліотеки, що відкидає можливість побудови класів еквівалентності для тестування, також інші методи не зможуть дати достатню якість тестування сайту.

Разом з методом припущення про помилку для налагодження програми буде використовуватись метод індукції та зворотного відстеження.

При використанні методу індукції було проаналізовано випадки помилки та дані чи дії що її спричиняють. Гіпотезами було отримано набір можливих причин помилки.

Під час тестування деякі помилки призводили до зупину програми в конкретному місці, для налагодження програми було використано метод зворотного відстеження.

Припущення про помилку:

1. Що буде з відправкою повідомлення на пошту якщо на сервері зникне інтернет?
2. Що буде якщо видалити категорію, яка використовується в товарі?
3. Що буде при переході на адміністративні сторінки сайту неавторизованим користувачем? (Помилка відсутня, користувач посилається на сторінки доступні йому, такі як авторизація чи головна сторінка).
4. Якщо адміністратор перейде до перегляду товару, чи зможе він його додати?
5. Чи може один користувач мати декілька ролей та використовувати їх можливості?
6. Що буде якщо видалили категорію товару, що використовується при відображенні історії замовлень?
7. Що буде якщо видалити товар, який був замовлений?
8. Що буде якщо внаслідок дій над базою даних було пошкоджено дані користувача й він був видалений?

Протокол налагодження програми був записаний до табл. 6.1.

Таблиця 6.1 – Протокол налагодження програми

Опис помилки	Опис ситуації	Способи усунення	Дії, що були застосовані для усунення
1	2	3	4
Помилка з'єднання з SMTP сервером	На сервері зникло інтернет з'єднання	Обробка події відправки повідомлення та встановлення додаткової перевірки на можливість підключення	Блок роботи з SMTP сервером було обгорнуто в блок <code>try {} catch() {}</code> , що дає змогу піймати помилку з'єднання з сервером і не призводить до критичного пошкодження роботи сайту
Звернення до товару, категорії не знайдено	Оскільки категорії за замовчуванням товару не присвоюється, при видаленні категорії, товар має дані які не відповідають дійсності, й не може їх отримати	Додання категорії за замовчуванням, попередня перевірка категорії на наявність у системі	Перед кожним зверненням до категорії відбувається перевірка даних на значення null, Якщо товар редагується, за замовчуванням привласнюється перша з доступних категорій
Адміністратор перейшов до перегляду товару де було виявлено кнопку додати товар	Адміністратор не маючи права покупця отримав доступ до додання товарів до кошику	Перевірити доступ до функцій роботи з доданням товару, перевірити умови показу кнопки додання товару	Виявлено помилку в розташуванні умовних конструкцій, тому перевірка не спрацьовувала, блок було переміщено за межі блоку, що не давав правильно реагувати на роль користувача системи

Продовження таблиці 6.1

1	2	3	4
Відсутня кнопка переходу до адміністративної зони при наявності ролі користувача та покупця на одному акаунті	Адміністратор має роль користувача (покупця) та роль адміністратора, проте кнопка переходу до адміністративної зони недоступна, хоча через пряме посилання доступ до адміністративної зони можливий	Перевірити дані відображення кнопки переходу	Виявлено помилку в ієрархії блоків, через що перевірка на роль адміністратора була недосяжною умовою й кнопка не відображалась, блок було переміщено в місце, що йде за блоком кошику
Перегляд історії замовлень користувача, помилка виведення інформації замовлення, користувацька зона	Товар, що використовувався був видалений з бази даних	Контроль наявності даного товару у базі даних, заборона видалення товару з бази даних, створення резервної копії бази даних для відновлення видаленої інформації	Коли контролер отримує дані замовлення відбувається перевірка значення на null й формується текст про відсутність товару
Перегляд історії замовлень користувача, помилка виведення інформації замовлення, зона адміністратора	Користувач, що використовувався був видалений з бази даних	Контроль наявності даного користувача у базі даних, створення резервної копії бази даних для відновлення видаленої інформації	Коли контролер отримує дані замовлення відбувається перевірка значення на null й формується текст про відсутність користувача

Висновки до розділу 6

В цьому розділі обґрунтовуються обрані методи тестування та налагодження програми. Також розділ містить протокол налагодження програми, що описує помилки, які виникали під час тестування та засоби їх вирішення.

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

Під час виконання дипломної роботи було отримано практичні навички побудови сайту за архітектурним патерном MVC, досліджено методи та технології розробки сайтів на мові C#.

Під час дослідження методів розробки сайтів було обрано найбільш доречний до них стек технологій.

При побудові проекту використовувались такі фреймворки та бібліотеки:

ASP.Net MVC v.5 – фреймворк для побудови сайтів на мові C#;

Entity Framework v.6.4.4 – об'єктно-реляційним трансфер даних, для зв'язку с базою даних;

CKEditor v.3.6.4 – текстовий редактор, який використовується всередині веб-сторінок, для формування веб сторінок через графічні компоненти;

Dropzone версії 4.3.0 – бібліотека, яка забезпечує завантаження файлів методом drag and drop з попереднім переглядом зображень;

Fancybox версії 2.1.5 – інструмент для відображення картинок у спеціальній формі з використанням ефекту каруселі (перехід від картинки до картинки);

jQuery v.3.6.0 – бібліотека JavaScript, яка спрощує роботу з документом HTML, обробку подій та анімацію для швидкої веб-розробки;

Bootstrap версії 4.1.3 – найпопулярніший інтерфейсний фреймворк для розробки адаптивних сайтів.

Проект виконаний згідно поставлених до нього вимог у розділі 1 цього документу. Експлуатаційне призначення вважаю досягнутим, оскільки на поточній ітерації сайт повністю забезпечує основні вимоги для його функціонування, й дозволяє виконувати задачі поставленні до нього, – продавець має можливості розмістити свій товар для продажу на інтернет ресурсі, адміністратор має можливості для налаштування контенту сайту та даних товарів, а покупець має можливості для ознайомлення з товарами продавця, їх пошуком та покупкою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Freeman, A. Pro ASP.NET Core MVC. – NY.: Apress, 2016. – 1018 с.
2. Freeman, A. Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. – NY.: Apress, 2022. – 1286 с.
3. Jeffrey, R. CLR via C# (Developer Reference). – NY.: Microsoft Press, 2012. – 1443 с.
4. Freeman, A. Pro jQuery 2.0 (Expert's Voice in Web Development) . – NY.: Apress, 2013. – 1651 с.

ДОДАТКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Перший проректор Українського
державного університету науки і
технологій

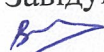
Анатолій РАДКЕВИЧ

18.02.22

ВЕБ-ПОРТАЛ ASP.NET З
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

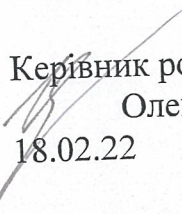
Технічне завдання
ЛИСТ ЗАТВЕРДЖЕННЯ
44165850.01253-01-ЛЗ

Представники
підприємства-розробника
Завідувач кафедри КІТ

 Вадим ГОРЯЧКІН

18.02.22

Керівник розробки

 Олександр ІВАНОВ

18.02.22

Виконавець

 Олександр ГЕТМАНЕНКО

18.02.22

Нормоконтролер

 Олена КУРОП'ЯТНИК

18.02.22

ЗАТВЕРДЖЕНО
44165850.01253-01-ЛЗ

ВЕБ-ПОРТАЛ ASP.NET З
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

Технічне завдання

44165850.01253-01

Листів 16

АНОТАЦІЯ

Документ 44165850.01253-01 «ВЕБ-ПОРТАЛ ASP.NET З ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО ПАТЕРНУ MVC. Технічне завдання» входить до складу програмної документації на програму, яка представляє собою сайт інтернет-магазину.

У даному документі представлені призначення та область застосування програмного продукту, основні вимоги, стадії та строки виконання проекту.

ЗМІСТ

Вступ.....	4
1 Підстави для розробки.....	6
2 Призначення розробки.....	7
3 Вимоги до програмного продукту.....	8
3.1 Вимоги до функціональних характеристик.....	8
3.2 Вимоги до надійності.....	10
3.3 Умови експлуатації.....	10
3.4 Вимоги до складу та параметрів технічних засобів.....	11
3.5 Вимоги до інформаційної та програмної сумісності.....	11
3.6 Вимоги до маркування і упаковки.....	11
3.7 Вимоги до транспортування і зберігання.....	12
4 Вимоги до програмної документації.....	13
5 Стадії і етапи розробки.....	14
6 Порядок контролю і приймання.....	15
Список використаної літератури.....	16

ВСТУП

Програмний додаток «Веб-портал asp.net з використанням архітектурного патерну mvc», що розробляється призначений для продажу товарів покупцям.

В сучасному світі досить часто виникає потреба в розробці веб-застосунків різної складності та масштабів. Майже кожен розвинений магазин приходить до необхідності в розробці сайту, який його рекламуватиме та, найбільш необхідне, надаватиме користувачам можливість ознайомитись з асортиментом товарів й придбати їх.

При побудові веб-застосунків на мові C# використовують декілька спеціальних фреймворків, до них належать ASP.Net MVC та ASP.Net core, які розширюють платформу розробників .NET інструментами та бібліотеками спеціально для створення веб-застосунків.

Також до їх складу входить веб-платформа Blazor, що дозволяє створювати інтерактивні веб-інтерфейси, використовуючи C# замість JavaScript для швидкого створення вмісту сторінок за допомогою коду вбудованого в html розмітку сторінок. Програми Blazor складаються з багаторазових компонентів веб-інтерфейсу, реалізованих за допомогою C#, HTML і CSS.

При використанні архітектури MVC сайт може легко поліпшуватись та розширюватись, всі рівні системи розділені між собою, що дозволяє розробляти їх окремо один від одного.

Особливістю даного фреймворку є те, що він надає зрозумілу адресу ресурсу, що дозволяє легко орієнтуватись у сайті й швидко визначати на якій сторінці знаходиться користувач.

Архітектура ASP.Net MVC є часто використовуваною при розробці сайтів, що підтверджує актуальність її вибору. Паралельно до неї набуває актуальності фреймворк ASP.Net core, проте здебільшого він має схожу до обраної архітектури структуру сайту. На даний момент цей фреймворк лише набуває популярності, тому його використання має менші переваги щодо готових програмних рішень та бібліотек що додаються до них. Зважаючи на вище зазначені недоліки фреймворку ASP.Net

Core було обрано фреймворк ASP.Net MVC.

При роботі сайту використовується фреймворк Bootstrap 4.6.1, який надає можливості для швидкої розробки адаптивних сторінок сайтів, також використовується бібліотека JavaScript – JQuery, що дозволяє легко отримувати доступ до елементів сторінки та їх даних, а також керувати ними, за допомогою цієї бібліотеки можливо легко оперувати іншими елементами сайту, наприклад, можливо легко обробляти зміст URL (адреси ресурсу), що значно полегшує виконання запитів та перехід до сторінок сайту за допомогою скриптів.

JQuery надає потужні можливості по обробці інтерфейсової частини на стороні клієнту й за необхідності виконання запитів до серверу.

Область застосувань: програмний продукт призначений для власників магазинів та користувачів інтернет-магазинів.

Експлуатаційне призначення – за допомогою інтернет-магазину продавець подає свій товар в зручному для покупців вигляді, а покупці в свою чергу отримують доступ до повного асортименту товарів магазину, мають можливості ознайомлення з їх детальними характеристиками, швидкого пошуку згідно наданих критеріїв та за потреби можливість придбання обраних товарів.

1 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є наказ від 08.12.2021 р. «Про призначення керівників та затвердження бакалаврських робіт», затверджений ректором Українського державного університету науки і технологій проф. Пшіньком О. М.

Тема дипломної роботи – «Веб-портал asp.net з використанням архітектурного патерну mvc». Керівник – доцент Іванов О. П.

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Функціональне призначення полягає в організації роботи магазину через інтернет ресурс.

Експлуатаційне призначення – за допомогою інтернет-магазину продавець подає свій товар в зручному для покупців вигляді, а покупці в свою чергу отримують доступ до повного асортименту товарів магазину, мають можливості ознайомлення з їх детальними характеристиками, швидкого пошуку згідно наданих критеріїв та за потреби можливість придбання обраних товарів.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Вимоги до функціональних характеристик

Функціональні вимоги – програмний продукт надаватиме наступні можливості в залежності від рівня доступу до роботи сайту.

Покупець:

- додання товарів до кошику;
- видалення товарів з кошику;
- перегляд замовлення;
- редагування замовлення (збільшення кількості товарів, зменшення кількості товарів, видалення товарів у замовленні);
- перегляд історії замовлень;
- перегляд товарів;
- пошук товарів за назвою;
- фільтрація за мінімальною та максимальною ціною товару;
- фільтрація за категорією;
- детальний перегляд інформації про характеристики товару;
- оплата товару;
- робота з профілем: створення акаунту та редагування даних профілю;

Адміністратор:

- додання товарів та інформації про них;
- редагування інформації про товари;
- видалення товарів;
- додання категорій товарів;
- редагування категорій товарів;
- видалення категорій товарів.
- пошук товарів за назвою;
- фільтрація за мінімальною та максимальною ціною товару;
- фільтрація за категорією;
- додання сторінок для пунктів меню;

- видалення сторінок для пунктів меню;
- робота з виглядом головної сторінки для користувача;
- робота з бічною частиною сторінки.

Обмеження системи та додаткові можливості:

- головна сторінка не повинна видалятися;
- сторінки редагуються як в коді html так і за допомогою редактору сторінок;
- при видаленні товарів вони не знищуються, а лише помічаються як видалені й надалі ці товари можуть використовуватись у перегляді історії замовлень;
- відсутня можливість обрання ролей адміністраторів, тобто існує стільки адміністраторів скільки буде необхідно замовнику, їх дані не змінюються;
- адміністратори не можуть редагувати інформацію про інших користувачів;
- при видаленні категорій, товари не мають категорії за замовченням, відповідно й відображатись вони не будуть;
- валідація даних та відповідні повідомлення про некоректність введеної інформації;
- розділення сторінок на ролі: тобто певні сторінки відповідають тільки ролі адміністратору, чи користувачу. Користувач не повинен отримати доступ до сторінки адміністратора лише за допомогою коректного для адміністративної зони url, так само адміністратор не може отримати доступ до додання товарів у корзину відповідно й видаляти він їх не може, проте якщо адміністратор захоче переглянути корзину він має на це право, проте необхідно вивести відповідне повідомлення про відсутність товару;
- при виході з акаунту корзина очищується;
- перегляд галереї товарів за допомогою спеціального вікна;
- додання картинок до галереї товару за допомогою перетягування файлів;
- фільтрація файлів картинок, допустимі розширення: .jpg, .jpeg, .gif, .png;
- критерії для фільтрації повинні зберігатися після виконання пошуку.

3.2 Вимоги до надійності

Вимоги до надійності наступні:

- для полів ведення необхідно забезпечити контроль ведених даних;
- при оновленні даних забезпечити показ відповідних повідомлень про стан роботи програми та результати виконання операцій;
- наявність архівної копії тексту програми;
- наявність резервної копії бази даних на зовнішньому носії (сервері де розміщена база даних).

3.3 Умови експлуатації

Програмний продукт повинен використовуватись у приміщеннях, які відповідають умовам роботи ЕОМ, а саме мають такі кліматичні, санітарні та гігієнічні умови, які відповідають ДНАОП 0.00-1.31-99 (див. табл. 3.1).

Таблиця 3.1 – Кліматичні умови

Пора року	Категорія робіт згідно з ГОСТ 12.01-005-88	Температура повітря, град.С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
		Оптимальна	Оптимальна	Оптимальна
Холодна	легка-1-а	22-24	40-60	0,1
	легка-1-б	21-23	40-60	0,1
Тепла	легка-1-а	23-25	40-60	0,1
	легка-1-б	22-24	40-60	0,2

Працювати з програмою може людина, що має навички роботи з персональним комп'ютером та ознайомлена з керівництвом користувача.

3.4 Вимоги до складу та параметрів технічних засобів

Розроблюваний програмний продукт повинен використовуватись на ЕОМ, що мають:

- 2 гігабайти оперативної пам'яті;
- браузер.

Та мобільних пристроях, що мають:

- 2 гігабайти оперативної пам'яті;
- операційну систему android версії 9+.

Сервер повинен мати:

- 2 гігабайти оперативної пам'яті;
- процесор: 32 або 64 розрядний процесор із тактовою частотою 1 ГГц або вище з набором інструкцій SSE2;
- 50 гігабайт простору на жорсткому диску;
- засоби контролю (миша, віддалений робочий стіл, клавіатура);
- CD-R;
- монітор.

3.5 Вимоги до інформаційної та програмної сумісності

Програмний продукт розробляється для всіх видів операційних систем сімейства Windows 95 та наступні версії а також для операційних систем Android версії 9.0 та вище. Середовище розробки: Visual Studio 2022 professional. Система керування базами даних MS SQL Server 2017, інтернет-браузер.

3.6 Вимоги до маркування і упаковки

Упаковка програмного продукту, включаючи документацію повинна бути захищена від пошкоджень різного роду (механічних, кліматичних). На упаковці повинно бути вказана назва продукту, номер версії (якщо вона змінювалась), мінімальні системні вимоги. На зворотній стороні упаковки вказується розробник та його юридична адреса.

На рис. 3.1. представлено приклад маркування.

<p>Програмний додаток «Інтернет-магазин»</p> <p>Мінімальні системні вимоги:</p> <ul style="list-style-type: none">– 2 гігабайти оперативної пам'яті;– 50 гігабайт простору на жорсткому диску;	<p>Розробник: Гетманенко О. Р. Кафедра «КІТ», УДУНТ м. Дніпро, вул. Лазаряна 2 2022</p>
---	---

Рисунок 3.1 – Маркування

3.7 Вимоги до транспортування і зберігання

Транспортування повинне забезпечувати збереження програмного продукту, його цілісність і запобігання несанкціонованого доступу до нього. Програмний виріб міститься на оптичному носії даних типу CD-R і повинно мати відповідну упаковку для захисту від механічних ушкоджень та атмосферного впливу (пластиковий футляр або паперовий конверт).

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Вся документація до програмного додатку повинна задовольняти вимоги до програмної документації.

До складу програмної документації має входити технічне завдання та робочий проект.

До складу робочого проекту мають входити:

- специфікація;
- текст програми;
- опис програми.

Вся документація до програми повинна задовольняти вимогам державного стандарту до оформлення програмних документів [3].

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

В табл. 5.1 приведені стадії та етапи розробки програмного продукту.

Таблиця 5.1 – Стадії та етапи розробки

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, вибір та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	31.01.22 - 18.02.22
Робочий проект	Програмування та відладка програми.	19.02.22 - 01.05.22
	Тестування програми	02.05.22 - 24.05.22
	Розробка, узгодження і затвердження програмної документації.	25.05.22 - 12.06.22

6 ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ

Контроль здійснюється за допомогою виконання набору тестів з метою знаходження помилок в програмному продукті та його специфікації. Контроль виконання роботи забезпечується головним керівником розробки.

Прийом програмного продукту здійснюється уповноваженою комісією.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Freeman, A. Pro ASP.NET Core MVC. – NY.: Apress, 2016. – 1018 с.
2. Freeman, A. Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. – NY.: Apress, 2022. – 1286 с.
3. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю.М.Івченко, В.І.Шинкаренко, В.Г.Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В.Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В.Лазаряна, 2009. – 38 с.

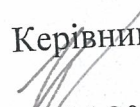
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

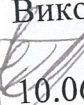
ЗАТВЕРДЖУЮ
Перший проректор Українського
державного університету
науки і технологій
Анатолій РАДКЕВИЧ
10.06.2022


ВЕБ-ПОРТАЛ ASP.NET 3
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

Специфікація
ЛИСТ ЗАТВЕРДЖЕННЯ
44165850.01253-01-ЛЗ

Представники
підприємства-розробника
Завідувач кафедри КІТ
 Вадим ГОРЯЧКІН
10.06.2022

Керівник розробки
 Олександр ІВАНОВ
10.06.2022

Виконавець
 Олександр ГЕТМАНЕНКО
10.06.2022

Нормоконтролер
 Олена КУРОП'ЯТНИК
10.06.2022

ЗАТВЕРДЖЕНО
44165850.01253-01-ЛЗ

ВЕБ-ПОРТАЛ ASP.NET З
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

Специфікація

44165850.01253-01

Листів 2

Позначення	Найменування	Примітка
	Документація	
44165850.01253-01-ЛЗ	Лист затвердження	
44165850.01253-01 12 01-ЛЗ	Лист затвердження	
44165850.01253-01 12 01	Текст програми	
44165850.01253-01 13 01-ЛЗ	Лист затвердження	
44165850.01253-01 13 01	Опис програми	

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ
Перший проректор Українського
державного університету
науки і технологій
Анатолій РАДКЕВИЧ
10.06.22

ВЕБ-ПОРТАЛ ASP.NET З
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

Текст програми
ЛИСТ ЗАТВЕРДЖЕННЯ
44165850.01253-01 12 01-ЛЗ

Представники
підприємства-розробника
Завідувач кафедри КІТ
Вадим ГОРЯЧКІН
10.06.22

Керівник розробки
Олександр ІВАНОВ
10.06.22

Виконавець
Олександр ГЕТМАНЕНКО
10.06.22

Нормоконтролер
Олена КУРОП'ЯТНИК
10.06.22

ЗАТВЕРДЖЕНО
44165850.01253-01 12 01-ЛЗ

ВЕБ-ПОРТАЛ ASP.NET З
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

Текст програми

44165850.01253-01 12 01

Листів 37

АНОТАЦІЯ

Документ 44165850.01253-01 12 01 «ВЕБ-ПОРТАЛ ASP.NET З ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО ПАТЕРНУ MVC. Текст програми» входить до складу програмної документації на програму, яка представляє собою сайт інтернет-магазину.

У даному документі представлений текст програми та схема взаємодії модулів програми. Програма реалізована на мові C# у програмному середовищі Visual studio 2022, з використанням фреймворку ASP.Net.

1	Схема взаємодії модулів.....	5
2	Текст програми.....	9
2.1	Модуль Index.cshtml (admin zone/Dashboard).....	9
2.2	Модуль AddPage.cshtml (Admin zone/Pages)	10
2.3	Модуль EditPage.cshtml (Admin zone/Pages)	11
2.4	Модуль EditSidebar.cshtml (Admin zone/Pages).....	12
2.5	Модуль Index.cshtml (Admin zone/Pages).....	13
2.6	Модуль PageDetails.cshtml (admin zone/Pages).....	14
2.7	Модуль _Layout.cshtml (admin zone/Shared).....	15
2.8	Модуль _FiltersMenuPartial.cshtml (admin zone/Shop)	17
2.9	Модуль AddProduct.cshtml (admin zone/Shop).....	18
2.10	Модуль Categories.cshtml (admin zone/Shop).....	19
2.11	Модуль EditProduct.cshtml (admin zone/Shop)	20
2.12	Модуль Orders.cshtml (admin zone/Shop).....	22
2.13	Модуль Products.cshtml (admin zone/Shop).....	23
2.14	Модуль AccountNavigationPartial.cshtml (buyer zone/Account).....	24
2.15	Модуль CreateAccount.cshtml (buyer zone/Account)	25
2.16	Модуль Login.cshtml (buyer zone/Account).....	26
2.17	Модуль Orders.cshtml (buyer zone/Account).....	27
2.18	Модуль UserProfile.cshtml (buyer zone/Account)	28
2.19	Модуль _AddToCardPartial.cshtml (buyer zone/Card)	29
2.20	Модуль _CardPartial.cshtml (buyer zone/ Card)	29
2.21	Модуль Index.cshtml (buyer zone/ Card)	29
2.22	Модуль _PagesMenuPartial.cshtml (buyer zone/Pages).....	31
2.23	Модуль _SidebarPartial.cshtml (buyer zone/ Pages).....	32
2.24	Модуль Index.cshtml (buyer zone/ Pages)	32
2.25	Модуль _Layout.cshtml (buyer zone/Shared).....	32
2.26	Модуль _CatMenuPartial.cshtml (buyer zone/Shop).....	35
2.27	Модуль _FiltersMenuPartial.cshtml (buyer zone/Shop).....	35

2.28	Модуль Category.cshtml (buyer zone/Shop)	35
2.29	Модуль ProductDetails.cshtml (buyer zone/ Shop).....	36

1 СХЕМА ВЗАЄМОДІЇ МОДУЛІВ

На рис. 1.1 – 1.7 приведена схема взаємодії модулів програми.

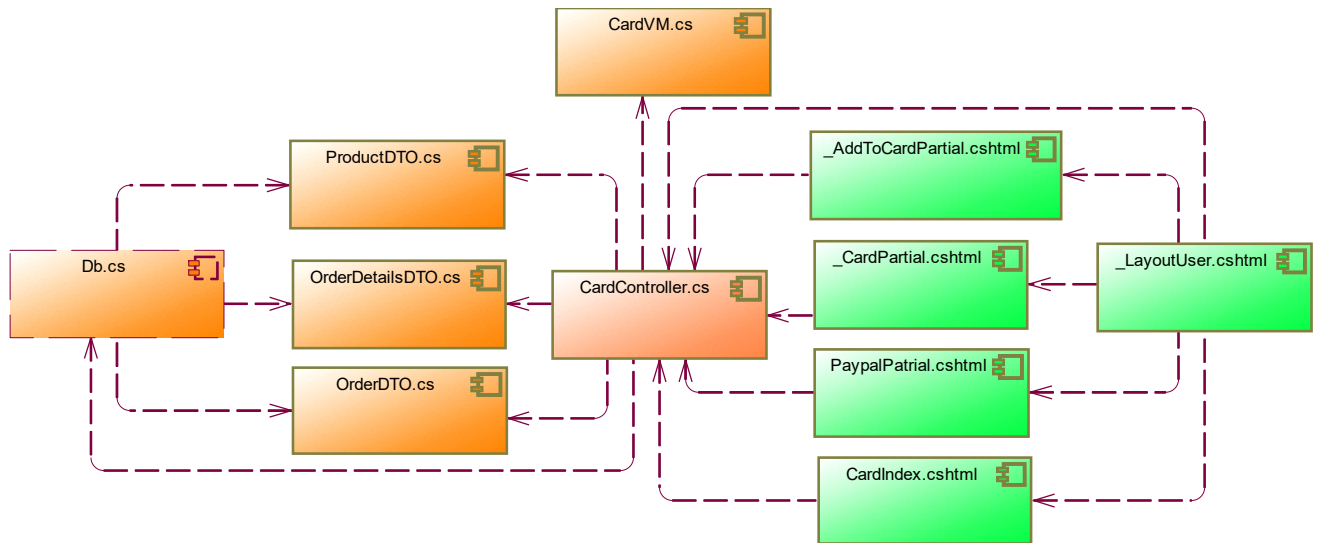


Рисунок 1.1 – Схема взаємодії модулів відносно контролеру списку покупок користувачької частини

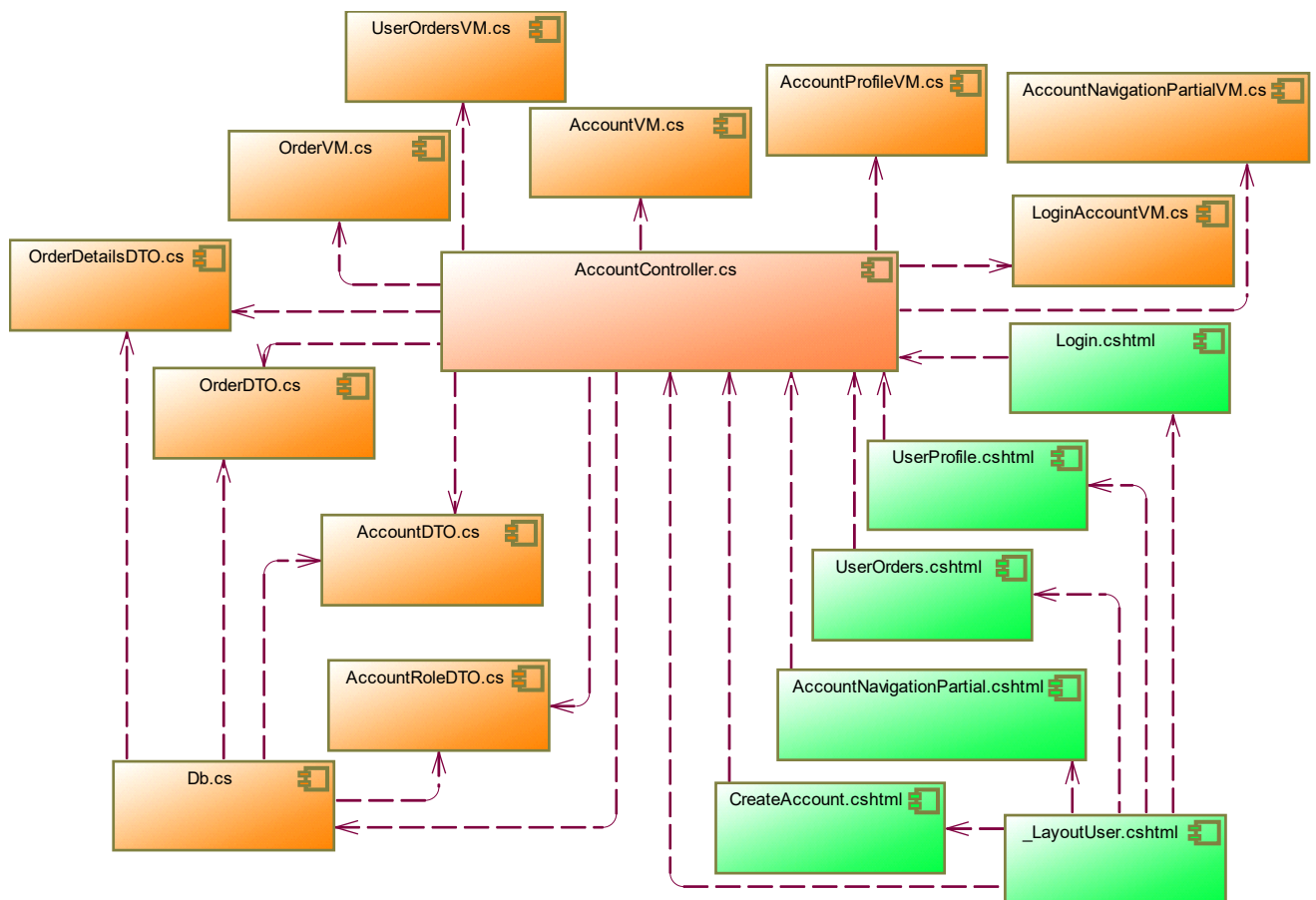


Рисунок 1.2 – Схема взаємодії модулів відносно контролеру акаунтів користувачької частини

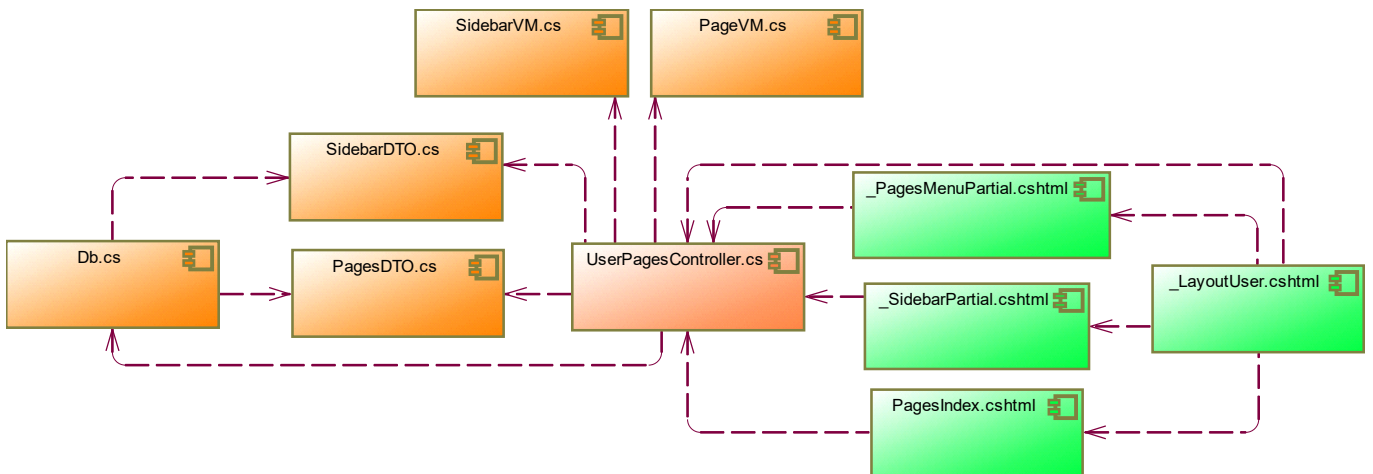


Рисунок 1.3 – Схема взаємодії модулів відносно контролеру сторінок користувачької частини

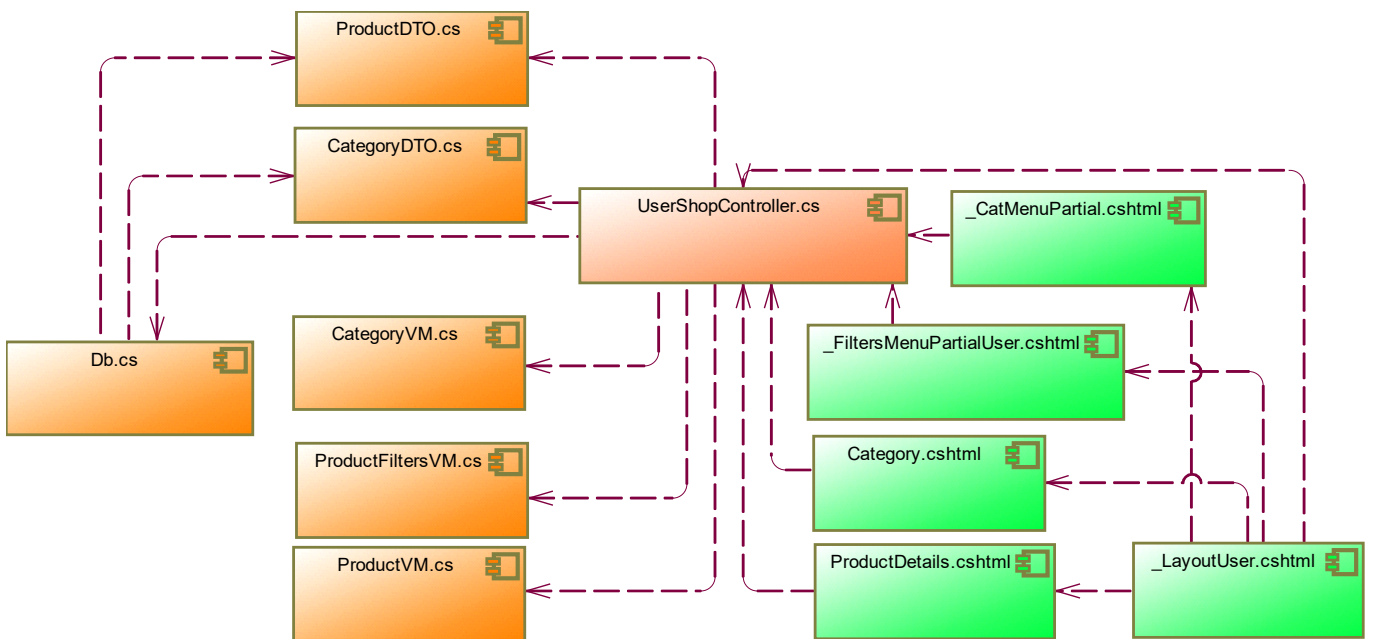


Рисунок 1.4 – Схема взаємодії модулів відносно контролеру роботи над покупками, товарами та категоріями товарів, для користувачької частини

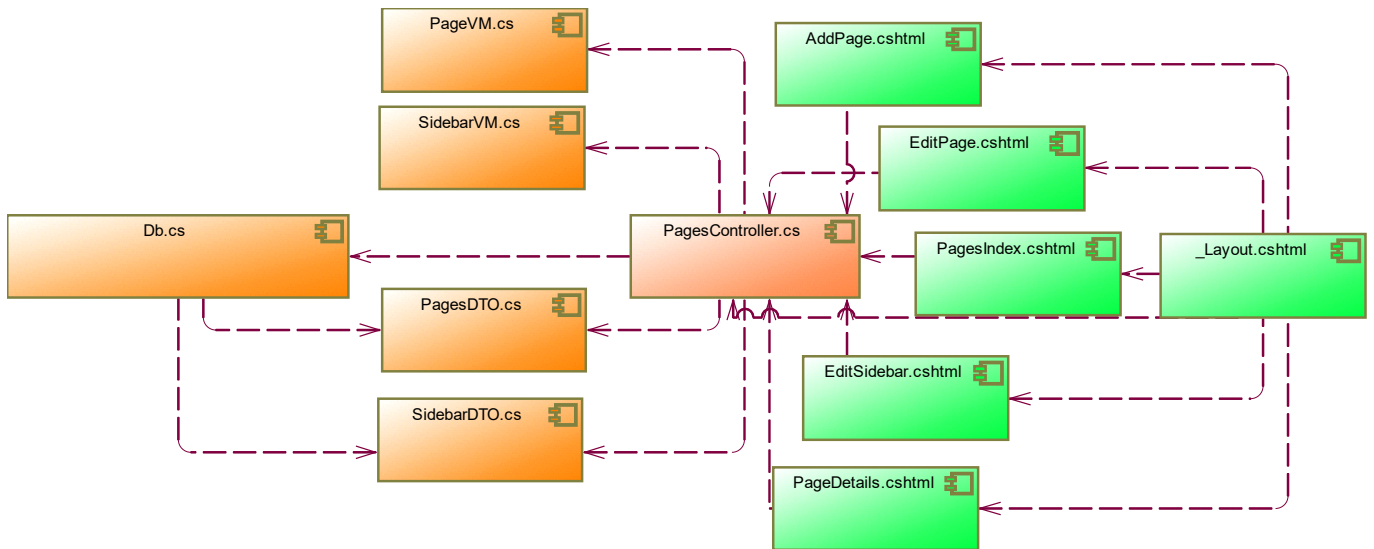


Рисунок 1.5 – Схема взаємодії модулів відносно контролеру сторінок адміністративної частини

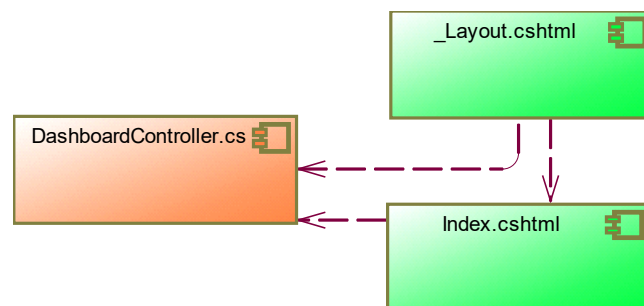


Рисунок 1.6 – Схема взаємодії модулів відносно контролеру головної сторінки адміністративної частини

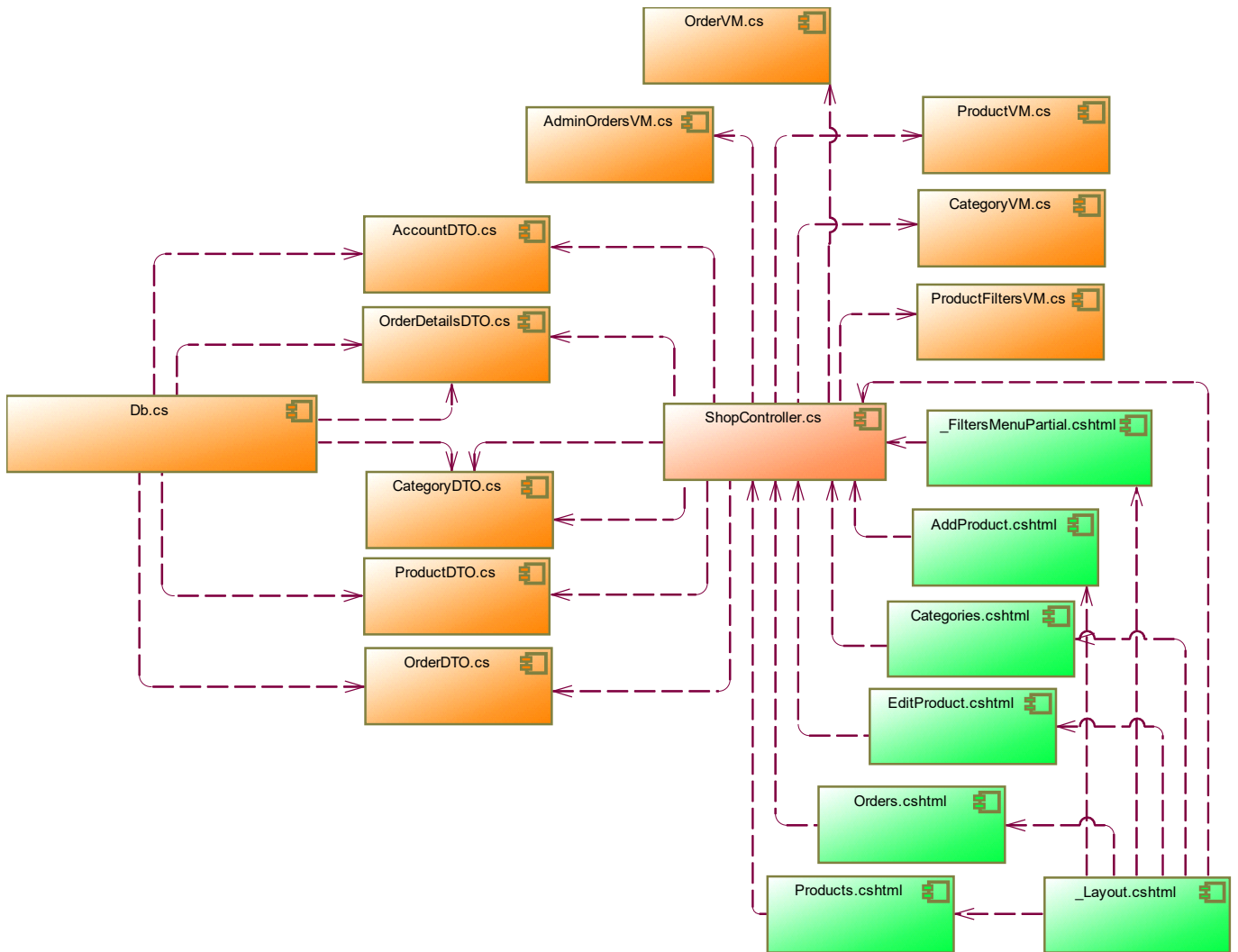


Рисунок 1.7 – Схема взаємодії модулів відносно контролеру роботи над покупками, товарами та категоріями товарів, для адміністративної частини

2 ТЕКСТ ПРОГРАММЫ

2.1 Модуль Index.cshtml (admin zone/Dashboard)

```

@{
    ViewBag.Title = "Dashboard";
}

<h2>Dashboard</h2>

<div class="row">
    <div class="card col-md-4 col-sm-12" style="width: 20rem;">
        <div class="text-center">
            
        </div>
        <div class="card-body text-justify">
            <p class="card-text">
                It is a long established fact that a reader will be distracted by the
                readable content of a page when looking at its layout. The point of
using
                Lorem Ipsum is that it has a more-or-less normal distribution of
letters, as opposed to using
                'Content here, content here', making it look like readable English.
                Many desktop publishing packages and web page editors now use Lorem
Ipsum as their default model text,
                and a search for 'lorem ipsum' will uncover many web sites still in
their infancy. Various versions have
                evolved over the years, sometimes by accident, sometimes on purpose
(injected humour and the like).
            </p>
        </div>
    </div>
    <div class="card col-md-4 col-sm-12" style="width: 20rem;">
        <div class="text-center">
            
        </div>
        <div class="card-body text-justify">
            <p class="card-text">
                It is a long established fact that a reader will be distracted by the
                readable content of a page when looking at its layout. The point of
using
                Lorem Ipsum is that it has a more-or-less normal distribution of
letters, as opposed to using
                'Content here, content here', making it look like readable English.
                Many desktop publishing packages and web page editors now use Lorem
Ipsum as their default model text,
                and a search for 'lorem ipsum' will uncover many web sites still in
their infancy. Various versions have
                evolved over the years, sometimes by accident, sometimes on purpose
(injected humour and the like).
            </p>
        </div>
    </div>
    <div class="card col-md-4 col-sm-12" style="width: 20rem;">
        <div class="text-center">
            
        </div>
        <div class="card-body text-justify">
            <p class="card-text">

```


It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of

using

 Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using
 'Content here, content here', making it look like readable English.
 Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text,
 and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have
 evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

```

        </p>
    </div>
</div>
</div>

```

2.2 Модуль AddPage.cshtml (Admin zone/Pages)

```
@model MVCStore.Models.ViewModels.Pages.PageVM
```

```
@{
    ViewBag.Title = "Add Page";
}
```

```
<h2>Add New Page</h2>
```

```
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Title, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Title, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Title, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Slug, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Slug, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Slug, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Body, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.TextAreaFor(model => model.Body, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Body, "", new { @class = "text-danger" })
            </div>
        </div>
    </div>
}
```

```

        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.HasSidebar, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            <div class="checkbox">
                @Html.EditorFor(model => model.HasSidebar)
                @Html.ValidationMessageFor(model => model.HasSidebar, "", new {
@class = "text-danger" })
            </div>
        </div>
    </div>

    <div class="form-group">
        <div class="offset-2 col">
            <input type="submit" value="Add Page" class="btn btn-default" />
        </div>
    </div>
</div>
}

<div>
    @Html.ActionLink("Back to Pages", "Index", null, new { @class="btn btn-info" })
</div>

@section Scripts
{
    <script>
        CKEDITOR.replace("Body")
    </script>
}

```

2.3 Модуль EditPage.cshtml (Admin zone/Pages)

```

@model MVCStore.Models.ViewModels.Pages.PageVM

@{
    ViewBag.Title = "Edit Page";
}

<h2>Edit Page</h2>

@if (TempData["SM"] != null)
{
    <div class="alert alert-success">
        @TempData["SM"]
    </div>
}

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.Title, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">

```

```

    @Html.EditorFor(model => model.Title, new { htmlAttributes = new { @class = "form-control" } })
    @Html.ValidationMessageFor(model => model.Title, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Slug, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @if (Model != null && Model.Slug == "home")
        {
            @Html.EditorFor(model => model.Slug, new { htmlAttributes = new { @class = "form-control", @readonly =
"readonly" } })
        }
        else
        {
            @Html.EditorFor(model => model.Slug, new { htmlAttributes = new { @class = "form-control" } })
        }
        @Html.ValidationMessageFor(model => model.Slug, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Body, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.TextAreaFor(model => model.Body, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Body, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.HasSidebar, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        <div class="checkbox">
            @Html.EditorFor(model => model.HasSidebar)
            @Html.ValidationMessageFor(model => model.HasSidebar, "", new { @class = "text-danger" })
        </div>
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Edit page" class="btn btn-default" />
    </div>
</div>
</div>
}

<div>
    @Html.ActionLink("Back to Pages", "Index", null, new { @class = "btn btn-info" })
</div>

```

2.4 Модуль EditSidebar.cshtml (Admin zone/Pages)

```
@model MVCStore.Models.ViewModels.Pages.SidebarVM
```

```
@{
    ViewBag.Title = "Edit Sidebar";
}
```

```
<h2>Edit Sidebar</h2>
```

```
@if (TempData["SM"] != null)
{
```

```

<div class="alert alert-success">
    @TempData["SM"]
</div>
}

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            <div class="col-md-10">
                @Html.TextAreaFor(model => model.Body, new { @class = "form-control" } )
                @Html.ValidationMessageFor(model => model.Body, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Edit" class="btn btn-default" />
            </div>
        </div>
    </div>
}

```

2.5 Модуль Index.cshtml (Admin zone/Pages)

```

@model IEnumerable<MVCStore.Models.ViewModels.Pages.PageVM>

@{
    ViewBag.Title = "Pages";
}

<h2>Pages</h2>

@if (TempData["SM"] != null)
{
    <div class="alert alert-success">
        @TempData["SM"]
    </div>
}

<p>
    @Html.ActionLink("Create New Page", "AddPage", "", new { @class = "btn btn-success" })
</p>

@if (!Model.Any())
{
    <h1 class="text-center"> There are no pages </h1>
}
else
{
    <table class="table sorting" id="pages">
        <tr class="home">
            <th>
                @Html.DisplayNameFor(model => model.Title)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Slug)
            </th>

```

```

</th>
<th>
    @Html.DisplayNameFor(model => model.HasSidebar)
</th>
<th></th>
</tr>

@foreach (var item in Model)
{
    <tr id="id_@item.Id" class="@item.Slug">
        <td>
            @if (item.Slug != "home")
            {
                <span class="ui-icon ui-icon-arrowthick-2-n-s"></span>
            }
            @Html.DisplayFor(modelItem => item.Title)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Slug)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.HasSidebar)
        </td>
        <td>
            @Html.ActionLink("Edit", "EditPage", new { id = item.Id }, new { @class = "btn btn-info" }) |
            @Html.ActionLink("Details", "PageDetails", new { id = item.Id }, new { @class = "btn btn-info" })

            <!-- If Slug is Home, dont show the delete button -->
            @if (item.Slug != "home")
            {
                string slash = "| ";
                @slash;
                @Html.ActionLink("Delete", "DeletePage", new { id = item.Id }, new { @class = "btn btn-danger delete" })
            }
        </td>
    </tr>
}
</table>
}

```

2.6 Модуль PageDetails.cshtml (admin zone/Pages)

```
@model MVCStore.Models.ViewModels.Pages.PageVM
```

```
@{
    ViewBag.Title = "PageDetails";
}
```

```
<h2>PageDetails</h2>
```

```

<div>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Title)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Title)
        </dd>

        <dt>

```

```

        @Html.DisplayNameFor(model => model.Slug)
    </dt>

    <dd>
        @Html.DisplayFor(model => model.Slug)
    </dd>

    <dt>
        @Html.DisplayNameFor(model => model.Body)
    </dt>

    <dd>
        @Html.Raw(Model.Body)
    </dd>

</dl>
</div>
<p>
    @Html.ActionLink("Edit", "EditPage", new { id = Model.Id }) |
    @Html.ActionLink("Back to Pages", "Index")
</p>

```

2.7 Модуль _Layout.cshtml (admin zone/Shared)

```

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - Admin Area</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
<header role="banner">
    <nav class="navbar navbar-expand-lg navbar-dark bg-main-menu fixed-top" style="height: 70px;">
        @Html.ActionLink("Dashboard", "Index", "Dashboard", "", new { @class = "navbar-brand" })
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText" aria-
controls="navbarText" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarText">
            <ul class="navbar-nav mr-auto">
                <li>
                    <div class="dropdown nav-item">
                        <div class="btn nav-link bg-main-menu dropdown-toggle text-left" type="button" data-toggle="dropdown">
                            Menu
                            <span class="caret"></span>
                        </div>
                        <ul class="dropdown-menu bg-main-menu">
                            <li class="bg-main-menu">
                                @Html.ActionLink("Pages", "Index", "Pages", "", new { @class = "nav-link" })
                            </li>
                            <li class="bg-main-menu">
                                @Html.ActionLink("Sidebar", "EditSidebar", "Pages", "", new { @class = "nav-link" })
                            </li>
                            <li class="bg-main-menu">
                                @Html.ActionLink("Categories", "Categories", "Shop", "", new { @class = "nav-link" })
                            </li>
                            <li class="bg-main-menu">
                                @Html.ActionLink("Products", "Products", "Shop", "", new { @class = "nav-link" })
                            </li>
                        </ul>
                    </div>
                </li>
            </ul>
        </div>
    </nav>

```

```

        <li class="bg-main-menu">
            @Html.ActionLink("Orders", "Orders", "Shop", "", new { @class = "nav-link" })
        </li>
    </ul>
</div>
</li>
</ul>
<ul class="navbar-nav ml-auto">
    <li class="bg-main-menu">
        @Html.ActionLink("Shopping card", "Index", "Pages", new { area = "" }, new { @class = "btn active nav-link
links" })
    </li>
</ul>
</div>
</nav>
</header>
<main class="body-content" role="main">
    @if (ViewBag.Title == "Products")
    {
        <div class="row mx-0 mt-5 pt-5 mb-5">
            <div class="col-sm-12 col-md-3 px-4 list-group text-dark">
                <div class="row">
                    <div class="col-sm-12 col-md-12 px-4 list-group text-dark">
                        @Html.Action("FiltersMenuPartial", "Shop", new { @class = "nav-link" })
                    </div>
                </div>
            <div class="col-sm-12 col-md-9 container body-content mx-auto mx-0 text-dark">
                @RenderBody()
            </div>
        </div>
    }
    else
    {
        <div class="row mx-0 mt-5 pt-5 mb-5 mx-auto">
            <div class="col-sm-10 col-md-10 container body-content mx-auto mx-0 text-dark">
                @RenderBody()
            </div>
        </div>
    }
</main>
<footer role="contentinfo" class="bg-main-menu">
    <div class="row py-4 bg-main-menu m-0">
        <div class="col-sm-12 col-md-3 my-auto">
            
        </div>
        <div class="col-sm-12 col-md-9">
            <h4 class="text-center">
                Contacts
            </h4>
            <hr style="background: whitesmoke;" />
            <div class="px-3">
                <div class="mb-2">
                    
                    <a class="link-cl-ws" href="tel:+380506726443">+38 (099) 329-57-10</a>
                </div>
                <div class="mb-2">
                    
                    <a class="link-cl-ws" href="https://goo.gl/maps/iQC4cjSe4JbKQA3z7">Kropyvnytskyi region, Olexandria, street
Cossack Way 89</a>
                </div>
                <div class="mb-2">

```

```

        
        <a class="link-cl-ws" href="mailto:agrosnabmarket@gmail.com">agrosnabmarket@gmail.com</a>
    </div>
    <div class="col-md-2 col-sm-6 p-0">
        <button href="tel:+380506726443" class="btn btn-success btn-block">
            
            &nbsp;   Advice
        </button>
    </div>
</div>
<hr style="background: whitesmoke;" />
</div>
</div>
</footer>

```

```

<!-- Connect bundle to project-->
@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/jqueryui")
@Scripts.Render("~/bundles/ckeditor")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>
</html>

```

2.8 Модуль _FiltersMenuPartial.cshtml (admin zone/Shop)

```
@model MVCStore.Models.ViewModels.Shop.ProductFiltersVM
```

```

<h3>Filters</h3>
<div class="row mx-auto">
    <div class="row m-1 col-12 px-0" width="100%">
        <div class="col-5 px-0">
            @Html.LabelFor(model => model.ProductName, htmlAttributes: new { @class = "control-label" })
        </div>
        <div class="col-7 px-0">
            @Html.EditorFor(model => model.ProductName, new { htmlAttributes = new { @class = "form-control filters" } })
        </div>
    </div>
    <div class="row m-1 col-12 px-0">
        <div class="col-5 px-0">
            @Html.LabelFor(model => model.minPrice, htmlAttributes: new { @class = "control-label" })
        </div>
        <div class="col-7 px-0">
            @Html.EditorFor(model => model.minPrice, new { htmlAttributes = new { @class = "form-control filters" } })
        </div>
    </div>
    <div class="row m-1 col-12 px-0">
        <div class="col-5 px-0">
            @Html.LabelFor(model => model.maxPrice, htmlAttributes: new { @class = "control-label" })
        </div>
        <div class="col-7 px-0">
            @Html.EditorFor(model => model.maxPrice, new { htmlAttributes = new { @class = "form-control filters" } })
        </div>
    </div>
    <div class="row m-1 save-btn col-12 px-0">
        <div class="btn btn-success col-12">
            Search
        </div>
    </div>
</div>

```


2.9 Модуль AddProduct.cshtml (admin zone/Shop)

```
@model MVCStore.Models.ViewModels.Shop.ProductVM
```

```
@{
    ViewBag.Title = "Add Product";
}
```

```
<h2>Add Product</h2>
```

```
@if (TempData["SM"] != null)
{
    <div class="alert alert-success">
        @TempData["SM"]
    </div>
}
```

```
@using (Html.BeginForm("AddProduct", "Shop", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
```

```
    @Html.AntiForgeryToken()
```

```
    <div class="form-horizontal">
```

```
        <hr />
```

```
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
```

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
```

```
            <div class="col-md-10">
```

```
                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
```

```
                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
```

```
            </div>
```

```
        </div>
```

```
    <div class="form-group">
```

```
        @Html.LabelFor(model => model.Description, htmlAttributes: new { @class = "control-label col-md-2" })
```

```
        <div class="col-md-10">
```

```
            @Html.TextAreaFor(model => model.Description, new { htmlAttributes = new { @class = "form-control" } })
```

```
            @Html.ValidationMessageFor(model => model.Description, "", new { @class = "text-danger" })
```

```
        </div>
```

```
    </div>
```

```
    <div class="form-group">
```

```
        @Html.LabelFor(model => model.Price, htmlAttributes: new { @class = "control-label col-md-2" })
```

```
        <div class="col-md-10">
```

```
            @Html.EditorFor(model => model.Price, new { htmlAttributes = new { @class = "form-control" } })
```

```
            @Html.ValidationMessageFor(model => model.Price, "", new { @class = "text-danger" })
```

```
        </div>
```

```
    </div>
```

```
    <div class="form-group">
```

```
        @Html.LabelFor(model => model.CategoryID, htmlAttributes: new { @class = "control-label col-md-2" })
```

```
        <div class="col-md-10">
```

```
            @Html.DropDownListFor(model => model.CategoryID, Model.Categories, "Select Category", new { htmlAttributes = new { @class = "form-control" } })
```

```
            @Html.ValidationMessageFor(model => model.CategoryID, "", new { @class = "text-danger" })
```

```
        </div>
```

```
    </div>
```

```
    <div class="form-group">
```

```
        <label class="control-label col-md-2">
```

```
            Product Image
```

```
        </label>
```

```
        <div class="col-md-10">
```

```
            <input type="file" accept=".jpg,.jpeg,.gif,.png" maxlength="" name="file" id="imageUpload" />
```

```

        
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Add product" class="btn btn-success" />
    </div>
</div>
</div>
}
<div>
    @Html.ActionLink("Back to Products", "Products")
</div>

```

2.10 Модуль Categories.cshtml (admin zone/Shop)

```

@model IEnumerable<MvcStore.Models.ViewModels.Shop.CategoryVM>

@{
    ViewBag.Title = "Categories";
}

<h2>Categories</h2>

@if (TempData["SM"] != null)
{
    <div class="alert alert-success">
        @TempData["SM"]
    </div>
}

<div class="new-cat">
    <div class="input-group mb-3">
        <div class="input-group-prepend">
            <span class="input-group-text" id="inputGroup-sizing-default">
                Category Name
            </span>
            <input class="form-control" type="text" id="newcatname" name="name" />
        </div>
    </div>
    <p>
        <a href="#" id="newcata" class="btn btn-success">Add a new category</a>
        <span class="ajax-text">
            
        </span>
    </p>
</div>

@if (!Model.Any())
{
    <h1 class="text-center">There are no categories here.</h1>
}
else
{
    <table class="table sorting" id="pages">
        <tr class="home">
            <th>
                @Html.DisplayNameFor(model => model.Name)
            </th>

            <th></th>
        </tr>

```

```

@foreach (var item in Model)
{
    <tr id="id_@item.Id">
        <td>
            <div class="container">
                <div class="row">
                    <div class="col">
                        <div class="ajax-div-td">

                            </div>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col-0">
                            <div class="ui-icon ui-icon-arrowthick-2-n-s" style="margin-top:10px;"></div>
                        </div>
                        <div class="col">
                            @Html.EditorFor(modelItem => item.Name, new { HtmlAttributes = new { @class = "form-control",
@readonly = "readonly" } })
                        </div>
                    </div>
                </div>
            </td>
            <td>
                @Html.ActionLink("Delete", "DeleteCategory", new { id = item.Id }, new { @class = "delete btn btn-danger" })
            </td>
        </tr>
    }
}
</table>
}

```

2.11 Модуль EditProduct.cshtml (admin zone/Shop)

```

@model MVCStore.Models.ViewModels.Shop.ProductVM

@{
    ViewBag.Title = "Edit Product";
}

<h2>Edit Product</h2>

@if (TempData["SM"] != null)
{
    <div class="alert alert-success">
        @TempData["SM"]
    </div>
}

@using (Html.BeginForm("EditProduct", "Shop", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
            </div>
        </div>
    </div>
}

```

```

    @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Description, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.TextAreaFor(model => model.Description, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Description, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Price, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Price, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Price, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.CategoryID, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownListFor(model => model.CategoryID, Model.Categories, new { htmlAttributes = new { @class =
"form-control" } })
        @Html.ValidationMessageFor(model => model.CategoryID, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.ImageName, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.ImageName, new { htmlAttributes = new { @class = "form-control", @readonly =
"readonly" } })
        @Html.ValidationMessageFor(model => model.ImageName, "", new { @class = "text-danger" })

        <br />

        @if (@Model.ImageName != null)
        {
            
        }

        <br />
        <br />

        
        <div class="col-md-10">
            <input type="file" accept=".jpg,.jpeg,.gif,.png" name="file" id="imageUpload" />
        </div>
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.isDeleted, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.CheckBoxFor(model => model.isDeleted, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.isDeleted, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">

```

```

        <input type="submit" value="Edit products" class="btn btn-success" />
    </div>
</div>
</div>
}

<hr/>
@if (!Model.GalleryImages.Any())
{
    <h3>There are no images</h3>
}
<form action="/admin/shop/SaveGalleryImages" method="post" enctype="multipart/form-data" class="dropzone"
id="dropzonebox">
    <div class="fallback">
        <input type="file" name="file" multiple="" />
        <input type="submit" value="Upload" />
    </div>
</form>

<br/>
<br/>

<div class="d-flex align-content-stretch flex-wrap">
    @foreach (var item in Model.GalleryImages)
    {
        <div class="card" style="width: 16rem;">
            
            <ul class="list-group list-group-flush">
                <li class="list-group-item">@Html.ActionLink("Delete", "DeleteImage", "Shop", new { @class = "delete-image btn
btn-danger btn-block", data_name = item })</li>
            </ul>
        </div>
    }
</div>

<div>
    @Html.ActionLink("Back to products", "Products")
</div>

<link href="~/Scripts/dropzone/basic.css" rel="stylesheet" />
<link href="~/Scripts/dropzone/dropzone.css" rel="stylesheet" />

```

2.12 Модуль Orders.cshtml (admin zone/Shop)

```

@model IEnumerable<MvcStore.Areas.Admin.Models.ViewModels.Shop.AdminOrdersVM>

@{
    ViewBag.Title = "Orders";
}

<h2>Orders</h2>

@if (!Model.Any())
{
    <h2 class="text-center">There are no orders</h2>
}
else
{
    <table class="table">
        <tr>
            <th>
                Order Number
            </th>

```

```

        <th>
            User name
        </th>
        <th>
            Order details
        </th>
        <th>
            Created At
        </th>
        <th>
            Total
        </th>
    </tr>

    @foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.OrderID)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.UserName)
        </td>
        <td>
            @foreach (var productDetails in item.ProductsAndQuantity)
            {
                @productDetails.Key @:x @productDetails.Value <br/>
            }
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.CreatedAt)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Total)
        </td>
    </tr>
    }

    </table>

}

```

2.13 Модуль Products.cshtml (admin zone/Shop)

```

@model IEnumerable<MvcStore.Models.ViewModels.Shop.ProductVM>

@{
    ViewBag.Title = "Products";
}

@using MvcStore.Models.Data
@using PagedList.Mvc;
@using PagedList;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />

<h2>Products</h2>
<select id="SelectCategory" name="SelectCategory">
    <option value="0">All</option>
    @foreach (var item in ViewBag.Categories)
    {
        if (ViewBag.SelectedCat == item.Value)
        {
            <option selected="selected" value="@item.Value">@item.Text</option>
        }
    }

```

```

        else
        {
            <option value="@item.Value">@item.Text</option>
        }
    }
</select>

<br />
<br />

<p>
    @Html.ActionLink("Add new product", "AddProduct", "", new { @class = "btn btn-success" })
</p>

<div class="d-flex align-content-stretch flex-wrap">
    @foreach (var item in ViewBag.OnePageOfProducts)
    {
        <div class="card" style="width: 12rem;">
            <a href="/admin/shop/EditProduct/@item.Id"></a>
            @if (@item.ImageName == null)
            {
                
            }
            else
            {
                
            }
            <div class="card-body p-3">
                <h5 class="card-title">@item.Name</h5>
                @if (item.isDeleted == true)
                {
                    <p class="card-text text-cropped-3 text-danger m-0">UNAVAILABLE</p>
                }
                else
                {
                    <p class="card-text text-cropped-3 text-success m-0">AVAILABLE</p>
                }
                <p class="card-text text-cropped-3 m-0">@item.Description</p>
            </div>
            <ul class="list-group list-group-flush">
                <li class="list-group-item">@item.Price</li>
                <li class="list-group-item">
                    <div class="row text-center center-block">
                        @Html.ActionLink("Edit", "EditProduct", new { id = item.Id }, new { @class = "btn btn-info col-4 p-1" })
                        &nbsp;
                        @Html.ActionLink("DISABLE", "DeleteProduct", new { id = item.Id }, new { @class = "btn btn-danger delete
col-6 p-1" })
                    </div>
                </li>
            </ul>
        </div>
    }
</div>

@Html.PagedListPager((IPagedList)ViewBag.onePageOfProducts, page => Url.Action("Products", new { page, catId =
ViewBag.SelectedCat }))

```

2.14 Модуль AccountNavigationPartial.cshtml (buyer zone/Account)

@model MVCStore.Models.ViewModels.Account.AccountNavigationPartialVM

```
<li class="navbar-text nav-link"><a href="/account/user-profile">&nbsp;&nbsp;&(@Model.FirstName @Model.LastName)</a></li>
```

2.15 Модуль CreateAccount.cshtml (buyer zone/Account)

```
@model MVCStore.Models.ViewModels.Account.AccountVM
```

```
@{
    ViewBag.Title = "Create Account";
}
```

Create Account

```

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.FirstName, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.FirstName, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.FirstName, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.LastName, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.LastName, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.LastName, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.EmailAddress, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.EmailAddress, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.EmailAddress, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Username, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Username, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Username, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.PasswordFor(model => model.Password, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.ConfirmPassword, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.PasswordFor(model => model.ConfirmPassword, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.ConfirmPassword, "", new { @class = "text-danger" })
            </div>
        </div>
    </div>
}

```



```

    @Html.PasswordFor(model => model.ConfirmPassword, new { htmlAttributes = new { @class = "form-control" } })
  })
  @Html.ValidationMessageFor(model => model.ConfirmPassword, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Create" class="btn btn-default" />
  </div>
</div>
</div>
}

```

2.16 Модуль Login.cshtml (buyer zone/Account)

```
@model MVCStore.Models.ViewModels.Account.LoginAccountVM
```

```

@{
  ViewBag.Title = "Login";
}

```

```
<h2>Login</h2>
```

```

@if (TempData["SM"] != null)
{
  <div class="alert alert-success">
    @TempData["SM"]
  </div>
}

```

```

@using (Html.BeginForm())
{
  @Html.AntiForgeryToken()

  <div class="form-horizontal">
    <hr />
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    <div class="form-group">
      @Html.LabelFor(model => model.Login, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.EditorFor(model => model.Login, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Login, "", new { @class = "text-danger" })
      </div>
    </div>

    <div class="form-group">
      @Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        @Html.PasswordFor(model => model.Password, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger" })
      </div>
    </div>

    <div class="form-group">
      @Html.LabelFor(model => model.RememberMe, htmlAttributes: new { @class = "control-label col-md-2" })
      <div class="col-md-10">
        <div class="checkbox">
          @Html.EditorFor(model => model.RememberMe)
          @Html.ValidationMessageFor(model => model.RememberMe, "", new { @class = "text-danger" })
        </div>
      </div>
    </div>
  </div>
}

```

```

<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <input type="submit" value="Login" class="btn btn-success"/>
    <a href="/account/create-account">Create an account</a>
  </div>
</div>
</div>
}

```

2.17 Модуль Orders.cshtml (buyer zone/Account)

@model IEnumerable<MvcStore.Models.ViewModels.Account.UserOrdersVM>

```

@{
    ViewBag.Title = "Orders";
}

<h2>Orders</h2>

@if (!Model.Any())
{
    <h2 class="text-center">There are no orders</h2>
}
else
{
    <table class="table">
        <tr>
            <th>
                Order Number
            </th>
            <th>
                Order details
            </th>
            <th>
                Created At
            </th>
            <th>
                Total
            </th>
        </tr>

        @foreach (var item in Model) {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.OrderID)
                </td>
                <td>
                    @foreach (var productDetails in item.ProductsAndQuantity)
                    {
                        @productDetails.Key @:x @productDetails.Value <br/>
                    }
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.CreatedAt)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Total)
                </td>
            </tr>
        }
    }
}

```

</table>

}

2.18 Модуль UserProfile.cshtml (buyer zone/Account)

@model MVCStore.Models.ViewModels.Account.AccountProfileVM

@{

ViewBag.Title = "Profile";

}

<h2>Your Profile</h2>

@if (TempData["SM"] != null)

{

<div class="alert alert-success" role="alert">

@TempData["SM"]

</div>

}

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.Id)

<div class="form-group">

@Html.LabelFor(model => model.FirstName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.FirstName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.FirstName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.LastName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.LastName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.LastName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.EmailAdress, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmailAdress, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmailAdress, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Username, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Username, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Username, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

```

    @Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.PasswordFor(model => model.Password, new { htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.ConfirmPassword, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.PasswordFor(model => model.ConfirmPassword, new { htmlAttributes = new { @class = "form-control" } })
    </div>
    @Html.ValidationMessageFor(model => model.ConfirmPassword, "", new { @class = "text-danger" })
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Edit" class="btn btn-default" />
    </div>
</div>
</div>
}

```

2.19 Модуль _AddToCardPartial.cshtml (buyer zone/Card)

```

@model MVCStore.Models.ViewModels.CardVM.CardVM

<div class="ajax-card">
    Your card: @Model.Quantity item(s) - @Model.Price USD
    <br />
    @Html.ActionLink("Checkout", "Index")
</div>

```

2.20 Модуль _CardPartial.cshtml (buyer zone/ Card)

```

@model MVCStore.Models.ViewModels.CardVM.CardVM

<div class="ajax-card">
    Your card: @Model.Quantity item(s) - @Model.Price USD
    <br />
    @Html.ActionLink("Checkout", "Index")
</div>

```

2.21 Модуль Index.cshtml (buyer zone/ Card)

```

@model IEnumerable<MVCStore.Models.ViewModels.CardVM.CardVM>

@{
    ViewBag.Title = "Shop Card";
    int count = 1;
}

<h2>Shop card</h2>

@if (ViewBag.Message != null)
{
    <h3>@ViewBag.Message</h3>
}
else
{
    <div class="ajax-bg">
        <span>

```

```

        
    </span>
</div>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.ProductName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Quantity)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Price)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Image)
        </th>
        <th>

        </th>
        <th>
            @Html.DisplayNameFor(model => model.Total)
        </th>
        <th>

    </th>
</tr>

    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(itemM => item.ProductName)
            </td>
            <td class="qty"@item.ProductID">
                @Html.DisplayFor(itemM => item.Quantity)
            </td>
            <td>
                @Html.DisplayFor(itemM => item.Price)
            </td>
            <td>
                @if (item.Image != null)
                {
                    
                }
                else
                {
                    
                }
                @Html.DisplayFor(itemM => item.Image)
            </td>
            <td class="row">
                <a class="inc-product btn btn-info p-1 col-12 m-1" style="max-width: 45px" href="#" data-
id="@item.ProductID">+</a>
                <a href="#" class="dec-product btn btn-info p-1 col-12 m-1" style="max-width: 45px" data-
id="@item.ProductID">-</a>
                <a href="#" class="remove-product btn btn-danger p-1 px-2 col-12 m-1" data-
id="@item.ProductID">Remove</a>
            </td>
            <td class="total"@item.ProductID">
                @Html.DisplayFor(itemM => item.Total) USD
            </td>
            <td>

```

```

        </td>
    </tr>
}
<tr>
    <td>
    </td>
    <td>
    </td>
    <td>
    </td>
    <td>
    </td>
    <td>
    </td>
    <td>
    </td>
    <td class="grand-total">
        <b>
            Grand total:
        </b>
        <span>
            @ViewBag.GrandTotal USD
        </span>
    </td>
</tr>
</table>

<div class="paypaldiv">
    <form class="paypal-form" action="https://www.sandbox.paypal.com/us/cgi-bin/webscr" method="post">
        <input type="hidden" name="cmd" value="_cart">
        <input type="hidden" name="upload" value="1" />
        <input type="hidden" name="business" value="sb-xairy16426561@business.example.com">

        @foreach (var item in Model)
        {
            <input type="hidden" name="item_name_@count" value="@item.ProductName">
            <input type="hidden" name="amount_@count" value="@item.Price">
            <input type="hidden" name="quantity_@count" value="@item.Quantity">
            count++;
        }

        <input type="hidden" name="currency_code" value="USD">
        <div style="display: none">
            <input type="image" src="http://www.paypal.com/en_US/i/btn/x-click-but01.gif" name="submit" alt="Make
payments with PayPal - it's fast, free and secure!">
        </div>
    </form>
</div>
<br />
<a href="#" class="btn btn-primary float-right place-order">Checkout</a>
}

```

2.22 Модуль _PagesMenuPartial.cshtml (buyer zone/Pages)

```
@model IEnumerable<MvcStore.Models.ViewModels.Pages.PageVM>
```

```

@foreach (var item in Model) {
    <li class="bg-main-menu nav-item">
        <a class="nav-link" href="/@item.Slug">@item.Title</a>
    </li>
}

```

2.23 Модуль _SidebarPartial.cshtml (buyer zone/ Pages)

@model MVCStore.Models.ViewModels.Pages.SidebarVM

@Html.Raw(Model.Body)

2.24 Модуль Index.cshtml (buyer zone/ Pages)

@model MVCStore.Models.ViewModels.Pages.PageVM

@{
ViewBag.Title = Model.Title;

}
@Html.Raw(Model.Body)

2.25 Модуль _Layout.cshtml (buyer zone/Shared)

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
</head>
<body>
  <header role="banner">
    <nav class="navbar navbar-expand-lg navbar-dark bg-main-menu fixed-top" style="height: 70px;">
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText" aria-
controls="navbarText" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse bg-main-menu" id="navbarText">
        <ul class="navbar-nav mr-auto bg-main-menu">
          <li class="bg-main-menu">
            <a class="navbar-brand" href="/">Home</a>
          </li>
          <li class="dropdown nav-item">
            <div class="btn nav-link bg-main-menu dropdown-toggle text-left" type="button" data-toggle="dropdown">
              Pages
              <span class="caret"></span>
            </div>
            <ul class="dropdown-menu bg-main-menu">
              @Html.Action("PagesMenuPartial", "Pages", new { @class = "nav-link" })
            </ul>
          </li>
          @if (Request.IsAuthenticated)
          {
            <li> @Html.ActionLink("Logout", "Logout", "Account", null, new { @class = "nav-item navbar-text nav-link
bg-main-menu" }) </li>
          }
          else
          {
            if (ViewBag.Title.Equals("Login"))
            {
              <li> @Html.ActionLink("Login", "Login", "Account", null, new { @class = "nav-item nav-link bg-main-
menu active" }) </li>
            }
            else
            {
```

```

        <li>@Html.ActionLink("Login", "Login", "Account", null, new { @class = "nav-item nav-link bg-main-
menu" })</li>
    }
}

@if (Request.IsAuthenticated && User.IsInRole("User"))
{
    if (ViewBag.Title == "Profile")
    {
        <li class="active bg-main-menu">@Html.Action("AccountNavigationPartial", "Account", new { area = "" })
    }
    else
    {
        <li class="bg-main-menu">@Html.Action("AccountNavigationPartial", "Account", new { area = "" }) </li>
    }
}
@if (Request.IsAuthenticated && User.IsInRole("User"))
{
    @Html.ActionLink("Orders", "Orders", "Account", new { area = "" }, new { @class = "nav-link bg-main-
menu" })
}
</ul>

@if (Request.IsAuthenticated && User.IsInRole("User") && ViewBag.Title != "Shop Card")
{
    if (!User.IsInRole("Admin"))
    {
        <span class="navbar-text bg-main-menu">
            @Html.Action("CardPartial", "Card")
        </span>
    }
    else
    {
        <span class="navbar-text bg-main-menu mr-4">
            @Html.Action("CardPartial", "Card")
        </span>
    }
}
@if (Request.IsAuthenticated && User.IsInRole("Admin"))
{
    <span class="navbar-text bg-main-menu">
        @Html.ActionLink("Admin Area", "Index", "Dashboard", new { area = "Admin" }, new { @class = "btn active
nav-link links bg-main-menu" })
    </span>
}
</div>
</nav>
</header>
<main class="body-content" role="main">
    <div class="row mx-0 px-0 mb-5 mt-5 pt-5 text-dark">

        <div class="col-sm-12 col-md-3 px-4 list-group text-dark">
            <div class="row">
                @if (ViewBag.Title == "Categories")
                {
                    <div class="col-sm-12 col-md-12 px-4 list-group text-dark">
                        @Html.Action("FiltersMenuPartial", "Shop", new { @class = "nav-link" })
                    </div>
                }
                <div class="col-sm-12 col-md-12 px-4 list-group text-dark">
                    @Html.Action("CatMenuPartial", "Shop")
                </div>
            </div>
        </div>
    </div>

```



```

    </div>
  </div>
</div>

```

```

    @if (ViewBag.Sidebar == "Yes")
    {
      <div class="mt-sm-3 mt-md-0 col-sm-12 col-md-6 px-4">
        @RenderBody()
      </div>
      <div class="col-sm-12 col-md-3 px-4">
        @Html.Action("SidebarPartial", "Pages")
      </div>
    }
  else
  {
    <div class="col-sm-12 col-md-9 px-4">
      @RenderBody()
    </div>
  }
</div>
</main>
<footer role="contentinfo" class="bg-main-menu">
  <div class="row py-4 bg-main-menu m-0">
    <div class="col-sm-12 col-md-3 my-auto">
      
    </div>
    <div class="col-sm-12 col-md-9">
      <h4 class="text-center">
        Contacts
      </h4>
      <hr style="background: whitesmoke;" />
      <div class="px-3">
        <div class="mb-2">
          
          <a class="link-cl-ws" href="tel:+380506726443">+38 (099) 329-57-10</a>
        </div>
        <div class="mb-2">
          
          <a class="link-cl-ws" href="https://goo.gl/maps/iQC4cjSe4JbKQA3z7">Kropyvnytskyi region, Olexandria,
street Cossack Way 89</a>
        </div>
        <div class="mb-2">
          
          <a class="link-cl-ws" href="mailto:agrosnabmarket@gmail.com">agrosnabmarket@gmail.com</a>
        </div>
        <div class="col-md-2 col-sm-6 p-0">
          <button href="tel:+380506726443" class="btn btn-success btn-block">
            
            &nbsp;Advice
          </button>
        </div>
      </div>
      <hr style="background: whitesmoke;" />
    </div>
  </div>
</footer>

```

```

@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/scripts")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)

```

```
</body>
</html>
```

2.26 Модуль _CatMenuPartial.cshtml (buyer zone/Shop)

```
@model IEnumerable<MVCStore.Models.ViewModels.Shop.CategoryVM>
```

```
<h3>Categories</h3>
```

```
@if (!Model.Any())
{
    <h5 class="text-danger">There are no categories</h5>
}

```

```
@foreach (var item in Model)
{
    @Html.ActionLink(@item.Name, $"/category/{@item.Slug.ToLower()}", "", new { @class = "list-group-item list-group-item-action" })
}

```

2.27 Модуль _FiltersMenuPartial.cshtml (buyer zone/Shop)

```
@model MVCStore.Models.ViewModels.Shop.ProductFiltersVM
```

```
<h3>Filters</h3>
<div class="row mx-auto">
    <div class="row m-1 col-12 px-0" width="100%">
        <div class="col-5 px-0">
            @Html.LabelFor(model => model.ProductName, htmlAttributes: new { @class = "control-label" })
        </div>
        <div class="col-7 px-0">
            @Html.EditorFor(model => model.ProductName, new { htmlAttributes = new { @class = "form-control filters" } })
        </div>
    </div>
    <div class="row m-1 col-12 px-0">
        <div class="col-5 px-0">
            @Html.LabelFor(model => model.minPrice, htmlAttributes: new { @class = "control-label" })
        </div>
        <div class="col-7 px-0">
            @Html.EditorFor(model => model.minPrice, new { htmlAttributes = new { @class = "form-control filters" } })
        </div>
    </div>
    <div class="row m-1 col-12 px-0">
        <div class="col-5 px-0">
            @Html.LabelFor(model => model.maxPrice, htmlAttributes: new { @class = "control-label" })
        </div>
        <div class="col-7 px-0">
            @Html.EditorFor(model => model.maxPrice, new { htmlAttributes = new { @class = "form-control filters" } })
        </div>
    </div>
    <div class="row m-1 save-btn col-12 px-0">
        <div class="btn btn-success col-12">
            Search
        </div>
    </div>
</div>
```

2.28 Модуль Category.cshtml (buyer zone/Shop)

```
@model IEnumerable<MVCStore.Models.ViewModels.Shop.ProductVM>
```

```
@{
```

```

    ViewBag.Title = ViewBag.SiteTitle;
}

<h2>@ViewBag.CategoryName</h2>
@if (!Model.Any())
{
    <h4 class="text-center text-danger">There are no products here.</h4>
}
else
{
    <div class="d-flex align-content-stretch flex-wrap">
        @foreach (var item in Model)
        {
            <div class="card" style="width: 12rem;">
                @if (@item.ImageName == null)
                {
                    
                }
                else
                {
                    
                }
                <div class="card-body p-3">
                    <h5 class="card-title">@item.Name</h5>
                    <p class="card-text text-cropped-3 m-0">@item.Description</p>
                </div>
                <ul class="list-group list-group-flush">
                    <li class="list-group-item">@item.Price USD</li>
                    <li class="list-group-item">
                        <div class="row text-center center-block">
                            @Html.ActionLink("Details", "Product-details", "Shop", new { name = @item.Slug }, new { @class = "btn
btn-info col-5 p-1" })
                        </div>
                    </li>
                </ul>
            </div>
        }
    </div>
}

@section scripts
{
    <script src="/Content/ShopJS/FiltersJS.js"></script>
}

```

2.29 Модуль ProductDetails.cshtml (buyer zone/ Shop)

```

@model MVCStore.Models.ViewModels.Shop.ProductVM

@{
    ViewBag.Title = Model.Name;
}

<h2>@Model.Name</h2>

<div class="row">
    <div class="col-sm-12 col-md-5">
        @if (@Model.ImageName != null)
        {
            
        }
    </div>
    <div class="col-sm-12 col-md-7">
        @if (@Model.Description != null)
        {
            <p>@Model.Description</p>
        }
        <ul class="list-group">
            <li>@Model.Price USD</li>
            <li>@Html.ActionLink("Details", "Product-details", "Shop", new { name = @Model.Slug }, new { @class = "btn
btn-info col-5 p-1" })</li>
        </ul>
    </div>
</div>

```

```

    }
    else
    {
        
    }
</div>
<div class="col-sm-12 col-md-7">
    @Html.DisplayFor(model => model.Description)
    <br />
    @Html.DisplayFor(model => model.Price) USD

    @if (Request.IsAuthenticated && User.IsInRole("User"))
    {
        <div class="addtocard">
            <a href="#" class="addtocard btn btn-success">Add to shop list</a>
            <span class="loader">
                
            </span>
            <div class="ajax-msg">
                That product has been added to your card.
            </div>
        </div>
    }
    else if (!User.IsInRole("Admin"))
    {
        <br />
        @Html.ActionLink("First you need Log in as our User!", "Login", "Account", null, new { @class = "btn btn-info mt-1"
    })
    }
</div>

</div>
<hr />

<div class="row">
    <h3>Gallery</h3>
    <div class="col-sm-12 fancyboxdiv">
        <br />
        <br />
        <div class="row">
            @foreach (var item in Model.GalleryImages)
            {
                <div>
                    <a class="fancybox" rel="gallery1" href="/Images/Uploads/Products/@Model.Id/Gallery/@item">
                        
                    </a>
                </div>
            }
        </div>
    </div>
</div>

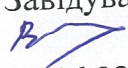
```

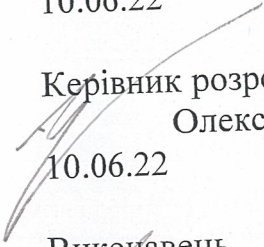
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

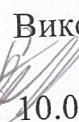
ЗАТВЕРДЖУЮ
Перший проректор Українського
державного університету
науки і технологій
Анатолій РАДКЕВИЧ
10.06.22

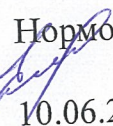
ВЕБ-ПОРТАЛ ASP.NET З
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

Опис програми
ЛИСТ ЗАТВЕРДЖЕННЯ
44165850.01253-01 13 01-ЛЗ

Представники
підприємства-розробника
Завідувач кафедри КІТ
 Вадим ГОРЯЧКІН
10.06.22

Керівник розробки
 Олександр ІВАНОВ
10.06.22

Виконавець
 Олександр ГЕТМАНЕНКО
10.06.22

Нормоконтролер
 Олена КУРОП'ЯТНИК
10.06.22

ЗАТВЕРДЖЕНО
44165850.01253-01 13 01-ЛЗ

ВЕБ-ПОРТАЛ ASP.NET З
ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО
ПАТЕРНУ MVC

Опис програми
44165850.01253-01 13 01

Листів 43

АНОТАЦІЯ

Документ 44165850.01253-01 13 01 «ВЕБ-ПОРТАЛ ASP.NET З ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО ПАТЕРНУ MVC. Опис програми» входить до складу програмної документації на програму, яка представляє собою сайт інтернет-магазину.

У даному документі представлено опис програми та її функціональних можливостей. Програма реалізована на мові C# у програмному середовищі Visual studio 2022, з використанням фреймворку ASP.Net.

ЗМІСТ

1	Загальні відомості	4
2	Функціональне призначення	5
3	Опис логічної структури.....	6
	3.1 Алгоритм програми.....	6
	3.2 Використані методи	6
	3.3 Структура програми.....	7
	3.4 Зв'язки програми з іншими програмами	15
4	Використані технічні засоби	16
5	Виклик завантаження.....	17
6	Вхідні дані.....	18
7	Вихідні дані.....	19
8	Опис призначеного для користувача інтерфейсу	20
	8.1 Опис станів програми	20
	8.2 Опис керування діалогом	23
	8.3 Формування екранів.....	25
9	Порядок роботи з програмою.....	39
10	Повідомлення.....	40

1 ЗАГАЛЬНІ ВІДОМОСТІ

Програмний продукт «Веб-портал asp.net з використанням архітектурного патерну mvc», що розробляється призначений для організації роботи магазину через інтернет ресурс.

Програмне забезпечення необхідне для функціонування:

- бібліотека .Net framework v.4.5 або вище;
- фреймворк Entity Framework v.6.4.4 або вище;
- інтернет браузер;
- фреймворк ASP.Net MVC v.5 або вище.

Мови програмування:

- SQL;
- C#.

2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Функціональне призначення полягає в організації роботи магазину через інтернет ресурс.

Функціональним обмеженням сайту, є те, що він повинен використовуватись у браузерях:

- Google Chrome v.101.0.4951.67+;
- Opera v.86.0.4363.59+;
- Microsoft Edge v.101.0.1210.53+.

3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ

3.1 Алгоритм програми

Алгоритм роботи для кожного модулю схожий. Рівень представлення надає користувачу дії, які він може виконати з додатком, для заповнення полів він використовує моделі представлення, які відображають ці поля. Для обробки подій кожне представлення посилається на відповідний контролер, який виконує обробку введеної інформації й надалі при необхідності передає або отримує дані що зберігаються в базі даних через об'єкти передачі даних, які представляють собою моделі розроблених в базі даних таблиць.

Також для авторизації користувача використовується фреймворк ASP.Net Identity, який виконує більшу частину логіки по обробці даних користувача для входу у систему та надання користувачу можливостей роботи з нею в залежності від ролі.

3.2 Використані методи

Під час розробки додатку було використано методологію розробки програм з використанням архітектурного патерну Model View Controller.

Кожна частина такої архітектури має притаманні лише їй обов'язки:

- логіка моделей призначена для роботи на логічному рівні, моделі забезпечують збереження даних користувача (так звані View Model) або дані предметної області (Data transfer object), їх перегляд у представленні та додаткові можливості перевірки їх вмісту чи відображення у представленні (так, наприклад, у C# є спеціальні види анотацій, які задають певні правила відображення полів даних та їх перевірку);
- логіка представлення - дані моделі повинні мати відповідне представлення, яке представляє собою засоби відображення цих даних, тобто інтерфейс користувача;
- логіка контролера, контролер є сполучником між моделями та представленнями, його обов'язок полягає в обробці подій користувача та отриманні необхідних даних для подальшого переходу до представлення, що відповідає отриманій моделі.

Також під час розробки було застосовано такі принципи об'єктно-орієнтованого програмування, як: поліморфізм, інкапсуляція та наслідування.

3.3 Структура програми

Для наочного відображення складових частин структури програми та їх взаємодії було розроблено діаграми класів (рис. 3.1 – 3.6).

На діаграмах представлено взаємодію класів між собою, через велику кількість класів, діаграми були представлені відносно контролерів сторінок, оскільки контролери є посередником між рівнями даних та представлення, й надають можливість найбільш повно охопити взаємодію між іншими модулями.

Призначення класів:

- DB – використовується для зв'язку з базою даних та отримання даних таблиць, що в ній зберігаються;
- AccountDTO – використовується для зв'язку з таблицею користувачів;
- AccountRoleDTO – використовується для зв'язку з таблицею ролей акаунтів ;
- CategoryDTO – використовується для зв'язку з таблицею категорій;
- OrderDetailsDTO – використовується для зв'язку з таблицею деталей замовлення;
- OrderDTO – використовується для зв'язку з таблицею замовлень;
- PagesDTO – використовується для зв'язку з таблицею сторінок;
- ProductDTO – використовується для зв'язку з таблицею продуктів;
- RoleDTO – використовується для зв'язку з таблицею можливих ролей для користувачів;
- SidebarDTO – використовується для зв'язку з таблицею ролей акаунтів бази даних;
- AccountNavigationPartialVM – використовується для зчитування/запису логіну й паролю з представлення;
- AccountProfileVM – використовується для зчитування/запису даних користувача з представлення;

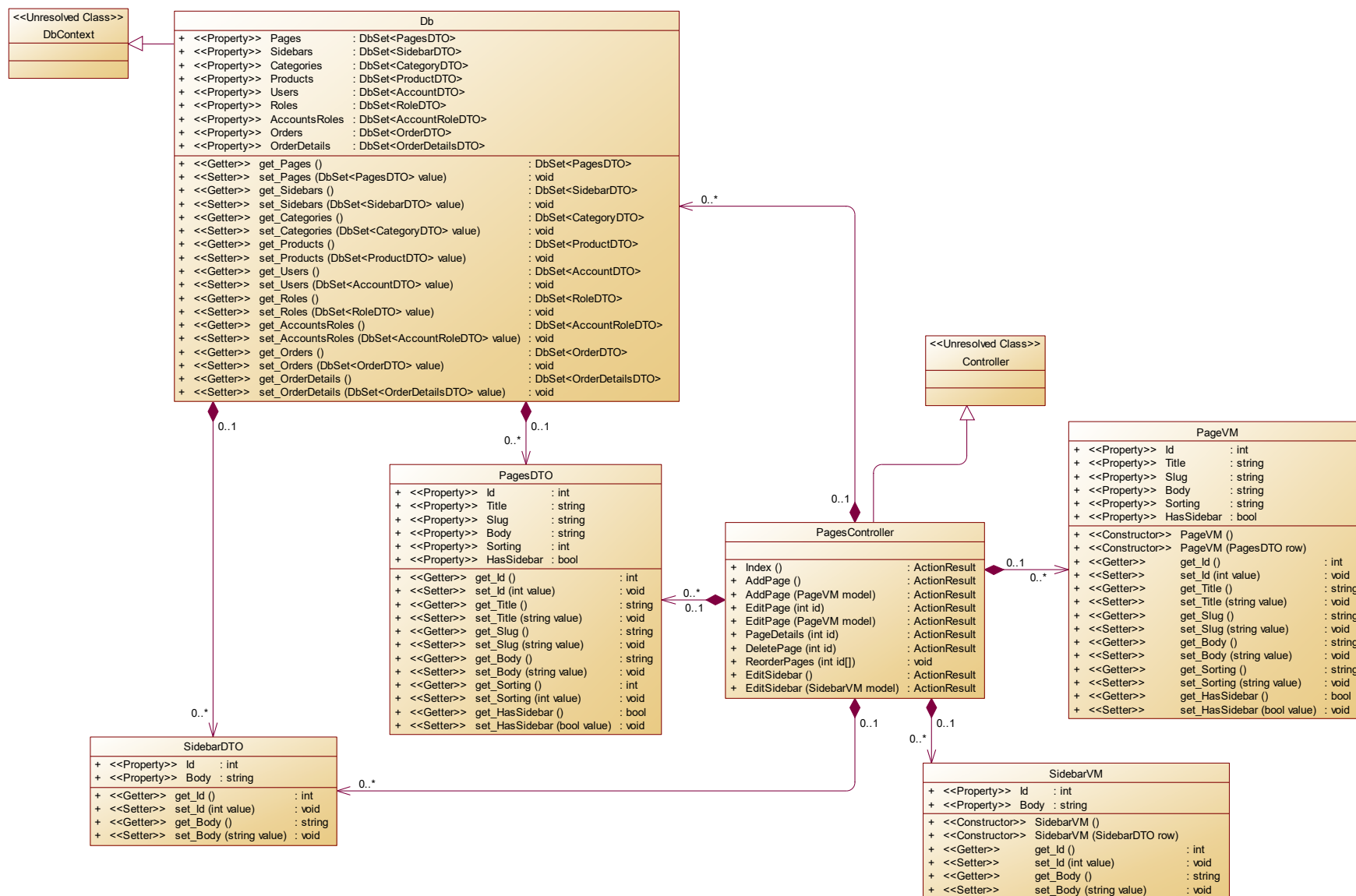


Рисунок 3.1 – Діаграма класів, відносно контролеру сторінок адміністративної зони

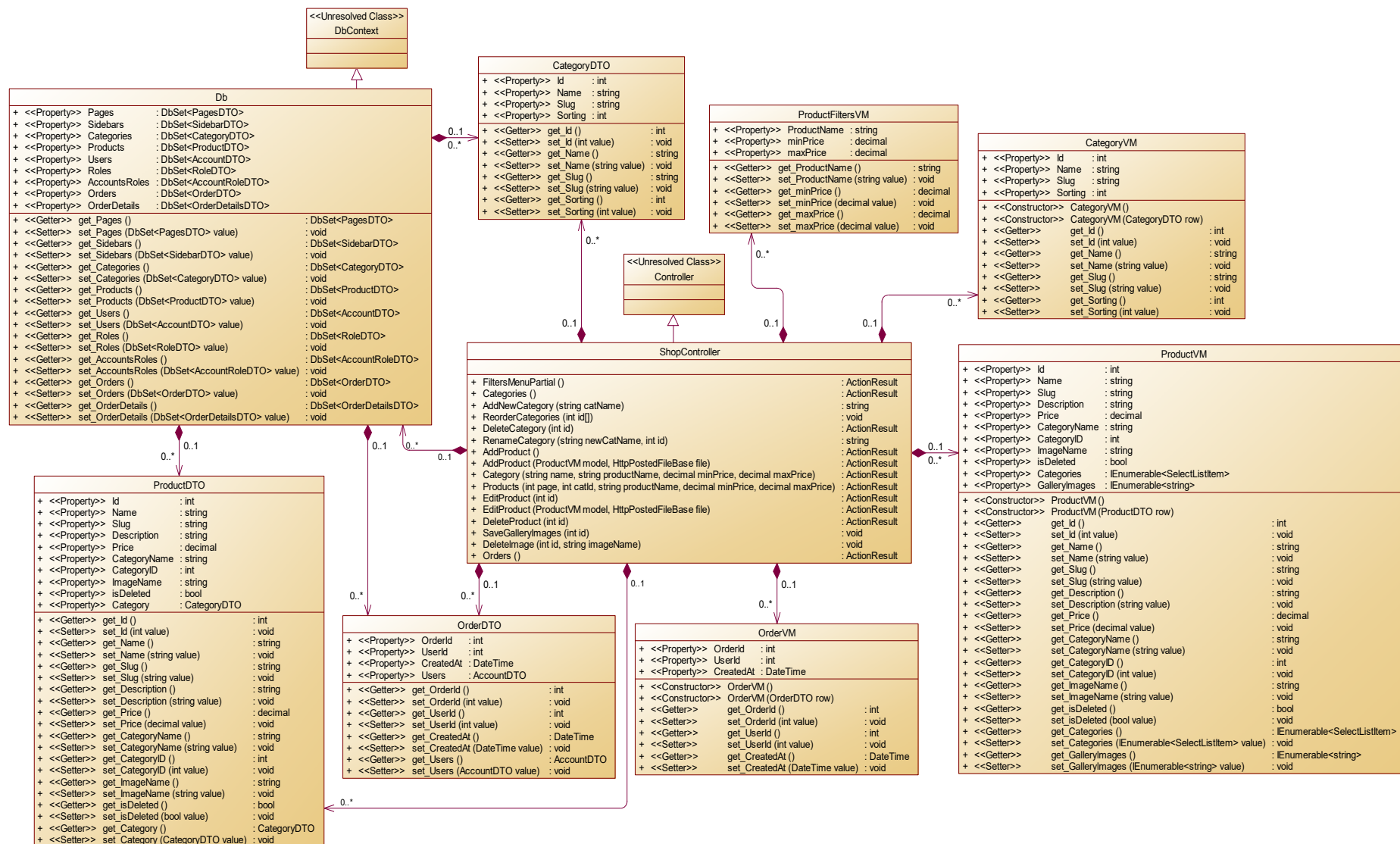


Рисунок 3.2 – Діаграма класів, відносно контролеру продуктів та категорій адміністративної зони

```

classDiagram
    class Db {
        <<Unresolved Class>>
        DbContext
        Users
    }
    class AccountDTO {
        <<Unresolved Class>>
        Users
        Id : int
        FirstName : string
        LastName : string
        EmailAddress : string
        Usemame : string
        Password : string
        get_Id() : int
        set_Id(int value) : void
        get_FirstName() : string
        set_FirstName(string value) : void
        get_LastName() : string
        set_LastName(string value) : void
        get_EmailAddress() : string
        set_EmailAddress(string value) : void
        get_Usemame() : string
        set_Usemame(string value) : void
        get_Password() : string
        set_Password(string value) : void
    }
    class AccountNavigationPartialVM {
        <<Unresolved Class>>
        Users
        FirstName : string
        LastName : string
        get_FirstName() : string
        set_FirstName(string value) : void
        get_LastName() : string
        set_LastName(string value) : void
    }
    class OrderVM {
        <<Unresolved Class>>
        Users
        OrderId : int
        UserId : int
        CreatedAt : DateTime
        OrderVM()
        OrderVM(OrderDTO row)
        get_OrderId() : int
        set_OrderId(int value) : void
        get_UserId() : int
        set_UserId(int value) : void
        get_CreatedAt() : DateTime
        set_CreatedAt(DateTime value) : void
    }
    class UserOrdersVM {
        <<Unresolved Class>>
        Users
        OrderID : int
        Total : decimal
        ProductsAndQuantity : Dictionary<string,int>
        CreatedAt : DateTime
        get_OrderID() : int
        set_OrderID(int value) : void
        get_Total() : decimal
        set_Total(decimal value) : void
        get_ProductsAndQuantity() : Dictionary<string,int>
        set_ProductsAndQuantity(Dictionary<string,int> value) : void
        get_CreatedAt() : DateTime
        set_CreatedAt(DateTime value) : void
    }
    class AccountController {
        <<Unresolved Class>>
        Users
        CreateCommonList(Queue<string> queue, Stack<string> stack) : List<string>
        Index() : ActionResult
        CreateAccount() : ActionResult
        Login() : ActionResult
        CreateAccount(AccountVM accountVM) : ActionResult
        Login(LoginAccountVM accModel) : ActionResult
        Logout() : ActionResult
        AccountNavigationPartial() : ActionResult
        UserProfile() : ActionResult
        EditProfile(AccountProfileVM accountVM) : ActionResult
        Orders() : ActionResult
    }
    class ProductDTO {
        <<Unresolved Class>>
        Users
        Id : int
        Name : string
        Slug : string
        Description : string
        Price : decimal
        CategoryName : string
        CategoryID : int
        ImageName : string
        isDeleted : bool
        Category : CategoryDTO
        get_Id() : int
        set_Id(int value) : void
        get_Name() : string
        set_Name(string value) : void
        get_Slug() : string
        set_Slug(string value) : void
        get_Description() : string
        set_Description(string value) : void
        get_Price() : decimal
        set_Price(decimal value) : void
        get_CategoryName() : string
        set_CategoryName(string value) : void
        get_CategoryID() : int
        set_CategoryID(int value) : void
        get_ImageName() : string
        set_ImageName(string value) : void
        get_isDeleted() : bool
        set_isDeleted(bool value) : void
        get_Category() : CategoryDTO
        set_Category(CategoryDTO value) : void
    }
    class OrderDetailsDTO {
        <<Unresolved Class>>
        Users
        Id : int
        OrderId : int
        UserId : int
        ProductId : int
        Quantity : int
        Orders : OrderDTO
        Users : AccountDTO
        Products : ProductDTO
        get_Id() : int
        set_Id(int value) : void
        get_OrderId() : int
        set_OrderId(int value) : void
        get_UserId() : int
        set_UserId(int value) : void
        get_ProductId() : int
        set_ProductId(int value) : void
        get_Quantity() : int
        set_Quantity(int value) : void
        get_Orders() : OrderDTO
        set_Orders(OrderDTO value) : void
        get_Users() : AccountDTO
        set_Users(AccountDTO value) : void
        get_Products() : ProductDTO
        set_Products(ProductDTO value) : void
    }
    class AccountVM {
        <<Unresolved Class>>
        Users
        Id : int
        FirstName : string
        LastName : string
        EmailAddress : string
        Usemame : string
        Password : string
        ConfirmPassword : string
        AccountVM()
        AccountVM(AccountDTO row)
        get_Id() : int
        set_Id(int value) : void
        get_FirstName() : string
        set_FirstName(string value) : void
        get_LastName() : string
        set_LastName(string value) : void
        get_EmailAddress() : string
        set_EmailAddress(string value) : void
        get_Usemame() : string
        set_Usemame(string value) : void
        get_Password() : string
        set_Password(string value) : void
        get_ConfirmPassword() : string
        set_ConfirmPassword(string value) : void
    }
    class AccountProfileVM {
        <<Unresolved Class>>
        Users
        Id : int
        FirstName : string
        LastName : string
        EmailAddress : string
        Usemame : string
        Password : string
        ConfirmPassword : string
        AccountProfileVM()
        AccountProfileVM(AccountDTO row)
        get_Id() : int
        set_Id(int value) : void
        get_FirstName() : string
        set_FirstName(string value) : void
        get_LastName() : string
        set_LastName(string value) : void
        get_EmailAddress() : string
        set_EmailAddress(string value) : void
        get_Usemame() : string
        set_Usemame(string value) : void
        get_Password() : string
        set_Password(string value) : void
        get_ConfirmPassword() : string
        set_ConfirmPassword(string value) : void
    }
    Db --> AccountDTO
    Db --> AccountNavigationPartialVM
    Db --> OrderVM
    Db --> UserOrdersVM
    Db --> AccountController
    Db --> ProductDTO
    Db --> OrderDetailsDTO
    Db --> AccountVM
    Db --> AccountProfileVM
    AccountDTO --> AccountController : 0..1
    AccountNavigationPartialVM --> AccountController : 0..1
    OrderVM --> AccountController : 0..1
    UserOrdersVM --> AccountController : 0..1
    AccountController --> AccountVM : 0..1
    AccountController --> AccountProfileVM : 0..1
    AccountController --> ProductDTO : 0..1
    AccountController --> OrderDetailsDTO : 0..1
    AccountController --> AccountVM : 0..1
    AccountController --> AccountProfileVM : 0..1
    AccountVM --> AccountProfileVM : 0..1
    ProductDTO --> OrderDetailsDTO : 0..1
    OrderDetailsDTO --> AccountVM : 0..1
    OrderDetailsDTO --> AccountProfileVM : 0..1
    AccountVM --> AccountProfileVM : 0..1
    
```



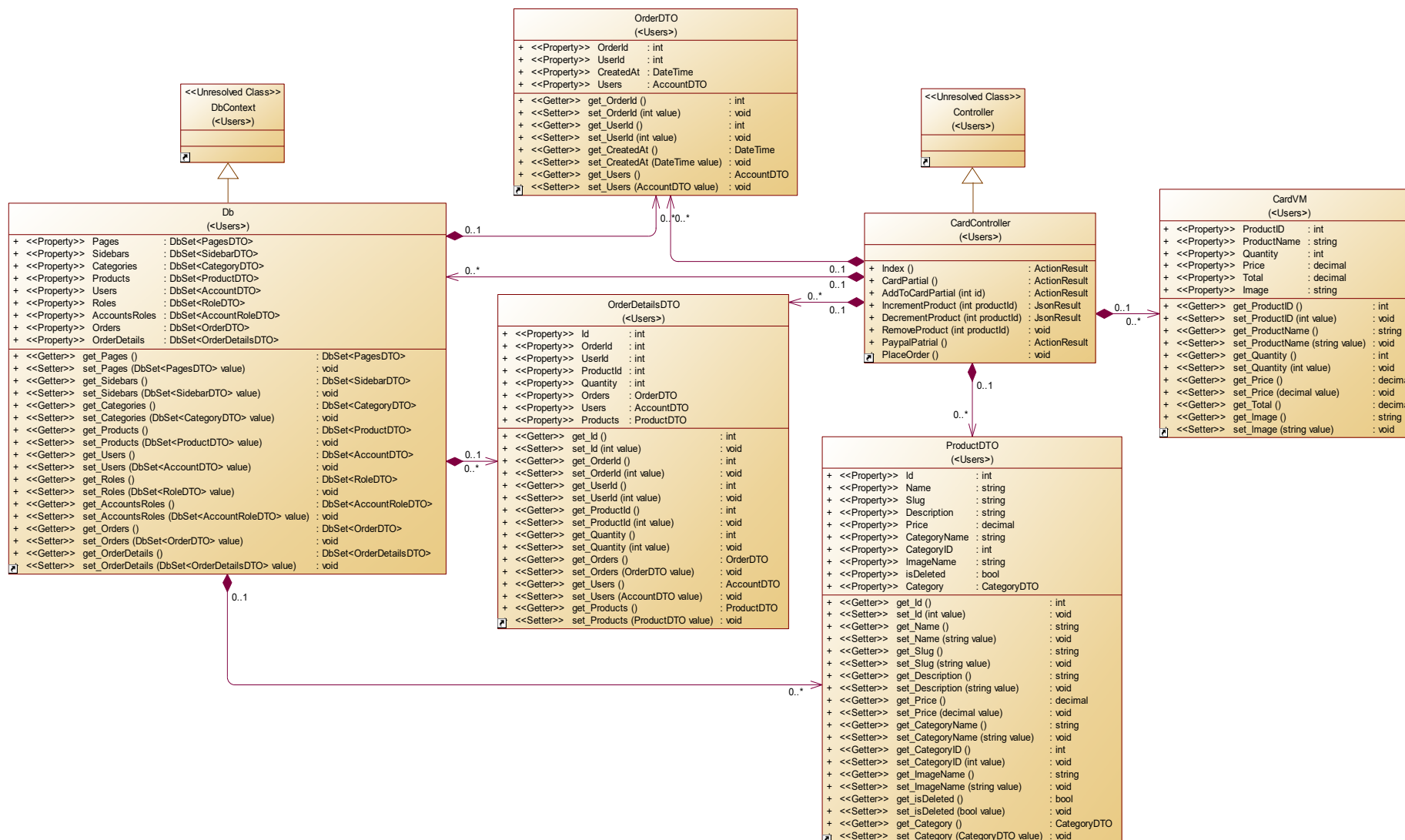



Рисунок 3.5 – Діаграма класів, відносно контролеру кошику товарів користувацької зони

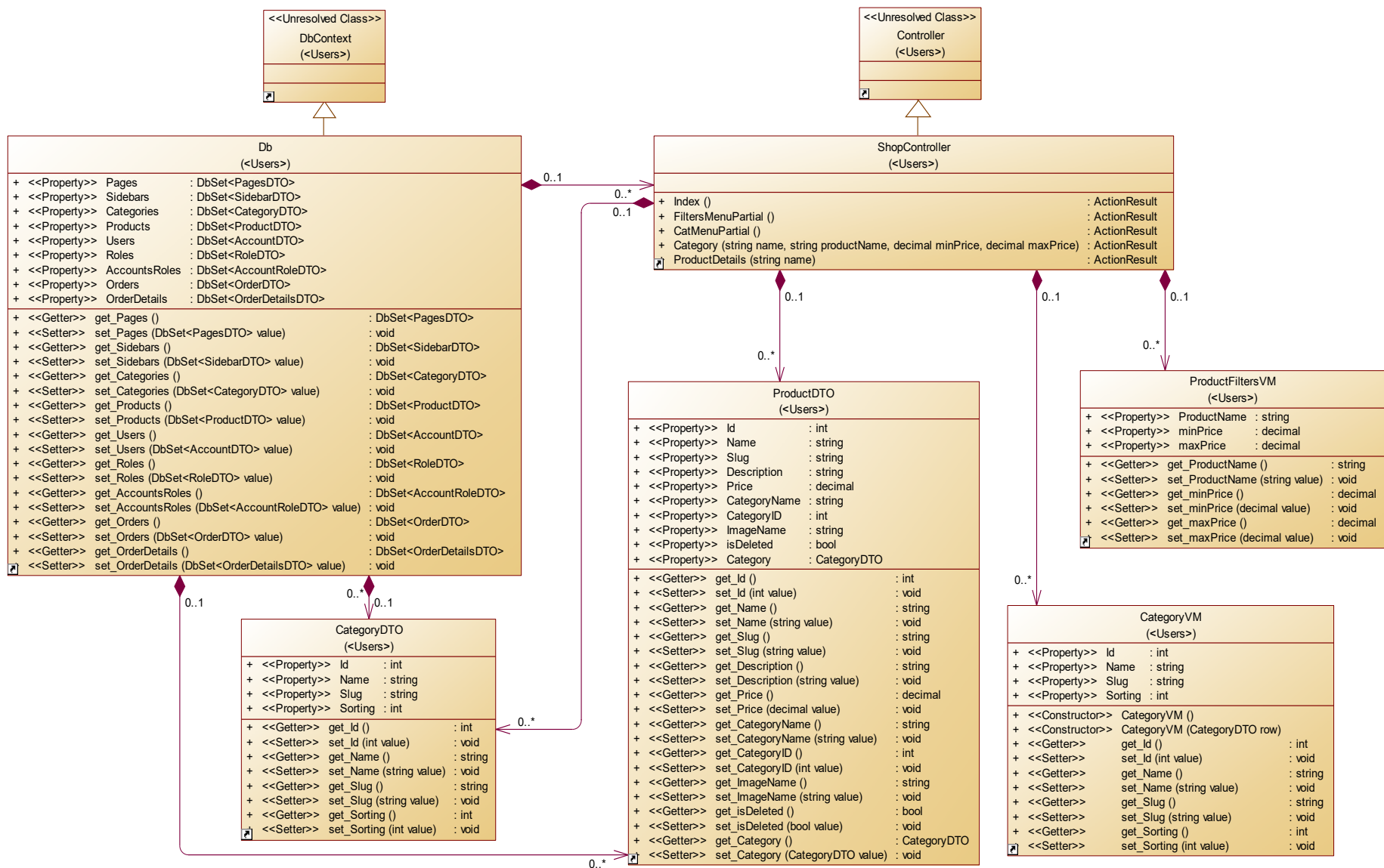


Рисунок 3.6 – Діаграма класів, відносно контролеру продуктів та категорій користувацької зони

- AccountVM – використовується для зчитування/запису даних користувача з представлення;
- LoginAccountVM – використовується для зчитування/запису даних користувача з представлення;
- UserOrdersVM – використовується для зчитування/запису даних про замовлення користувача з представлення;
- CardVM – використовується для зчитування/запису продуктів у списку покупок з представлення;
- PageVM – використовується для зчитування/запису даних про сторінку з представлення;
- SidebarVM – використовується для зчитування/запису даних про бічну сторінку з представлення;
- CategoryVM – використовується для зчитування/запису даних категорії з представлення;
- OrderVM – використовується для зчитування/запису даних про замовлення з представлення;
- ProductFiltersVM – використовується для зчитування/запису даних фільтрів продуктів з представлення;
- ProductVM – використовується для зчитування/запису даних продукту представлення;
- PagesController.cs (Admin) – виконує обробку подій у представленнях інформаційних сторінок та бічної панелі;
- ShopController.cs (Admin) – виконує обробку подій у представленнях, продуктів категорій, замовлень та фільтрів продуктів;
- AccountController.cs – виконує обробку подій у представленнях, пов'язаних з даними профілю користувача та його замовленнями;
- CardController.cs – виконує обробку подій у представленні списку покупок;
- PagesController.cs – виконує обробку подій у представленнях інформаційних сторінок та бічної панелі;

- ShopController.cs – виконує обробку подій у представленнях продуктів та категорій.

3.4 Зв'язки програми з іншими програмами

Для роботи сайту використовується спеціальний сервер запуску сайтів, що має назву IIS Server. IIS розроблений для роботи та адміністрування сайтів під управлінням операційних систем сімейства Microsoft. Веб-сервер інтегрований як компонент у windows і має різні функції аутентифікації.

4 ВИКОРИСТАНІ ТЕХНІЧНІ ЗАСОБИ

Розроблюваний програмний продукт повинен використовуватись на ЕОМ, що мають:

- 2 гігабайти оперативної пам'яті;
- браузер.

Та мобільних пристроях, що мають:

- 2 гігабайти оперативної пам'яті;
- операційну систему android версії 9+.

Сервер повинен мати:

- 2 гігабайти оперативної пам'яті;
- процесор: 32 або 64 розрядний процесор із тактовою частотою 1 ГГц або вище з набором інструкцій SSE2;
- 50 гігабайт простору на жорсткому диску;
- клавіатуру;
- мишу;
- CD-R;
- монітор.

5 ВИКЛИК ЗАВАНТАЖЕННЯ

Для того, щоб користуватись програмним продуктом необхідно перейти на сайт за посиланням: <http://alexiss-001-site1.ctempurl.com/>.

6 ВХІДНІ ДАНІ

Вхідними даними в програмі є атрибути сутностей опис яких представимо у вигляді ER-діаграми (рис. 6.1).

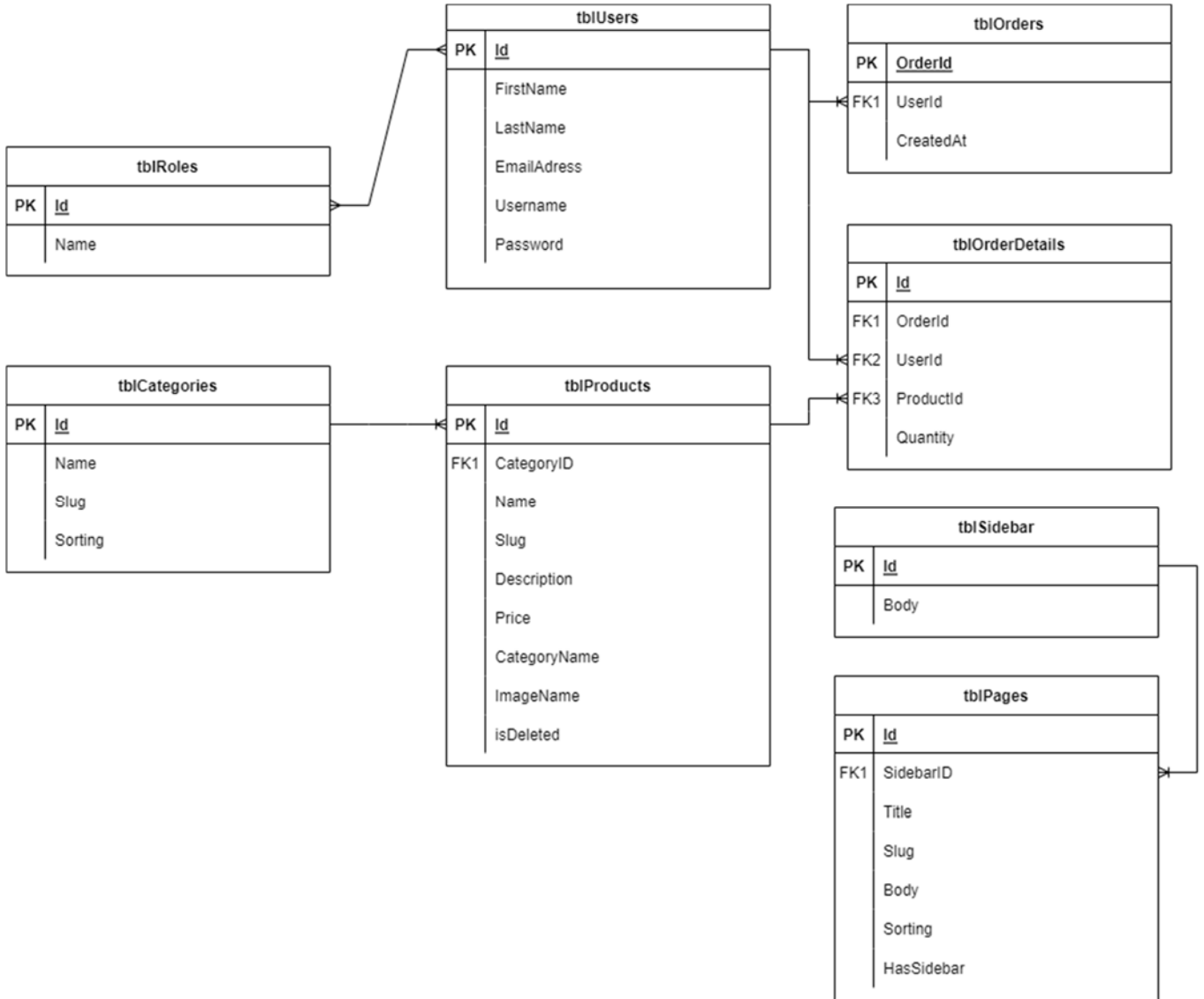


Рисунок 6.1 – ER-модель, що відображає сутності та їх атрибути
Дані вводяться з клавіатури.

7 ВИХІДНІ ДАНІ

Вихідними даними в програмі є:

- представлення, що використовуються для відображення даних сутностей бази даних;
- повідомлення програми про коректність введеної користувачем інформації, про помилки програми та про стан її виконання;
- анімації елементів сторінки;
- записи бази даних після проведення операцій над ними.

8 ОПИС ПРИЗНАЧЕНОГО ДЛЯ КОРИСТУВАЧА ІНТЕРФЕЙСУ

8.1 Опис станів програми

Стани, в яких може знаходитись програма наведено у табл. 8.1.

Таблиця 8.1 – Стани програми

№	Стан	Опис стану	Рекомендовані дії
1	2	3	4
1	Завантаження програми	Програма завантажується на клієнт користувача	Почекайте, доки сайт завантажиться
2	Відображення початкової сторінки, списку категорій, меню та футеру сайту	Програма формує початкову сторінку сайту, його меню та підвал. Після чого відображає сформовану сторінку користувачу	Ознайомитись з інтерфейсом сайту
3	Відображення списку, який випадає в головному меню	Програма формує список доступних інформаційних сторінок та відображає їх у вигляді елементів списку	Закрити або обрати сторінку що вам необхідна (наприклад сторінку новин)
4	Відображення інформативної сторінки	Програма формує обрану інформативну сторінку сайту	Переглянути контент сторінки
5	Відображення каталогу товарів	Програма формує список товарів та сторінку з ними	Переглянути товари, переглянути дані товару, відфільтрувати дані
6	Фільтрація даних	Після натискання кнопки фільтрації програма обробляє введені дані фільтрації й формує список товарів у вигляді їх карток	Дочекатись виведення товарів

1	2	3	4
7	Відображення деталей товару	Після переходу до деталей товару програма завантажує інформацію про нього та формує сторінку з цими даними	Додати товар до кошика, якщо користувач авторизований як покупець, перейти на сторінку авторизації, якщо користувач не авторизований, переглянути інформацію про товар, обрати фото з галереї товару для перегляду
8	Відображення сторінки авторизації	Програма формує сторінку авторизації та відображає її	Ввести логін та пароль користувача або перейти до створення нового користувача
9	Відображення сторінки створення користувача	Програма формує сторінку з полями профілю та відображає її	Створити нового користувача правильно заповнивши всі поля форми, або повернутись на сторінку авторизації
10	Перехід до адміністративної зони	Програма перевіряє роль користувача та надає йому доступ до адміністративної сторінки якщо у користувача є права адміністратора або повертає його на головну сторінку. Якщо користувач є адміністратором програма формує головну сторінку зони адміністратора та відображає її	Дочекатись завантаження сторінки адміністратора або головної сторінки. В разі отримання доступу до сторінки адміністратора ознайомитись з головним меню адміністративної зони
11	Перехід до корегування інформативних сторінок	Програма формує список наявних сторінок й відображає сторінку на якій представлено цей список та кнопки для керування ним	Дочекатись завантаження сторінки

1	2	3	4
12	Перехід до корегування бічної сторінки	Програма формує сторінку редагування бічної сторінки та відображає її	Дочекатись завантаження сторінки
13	Перехід до корегування категорій товару	Програма формує список наявних категорій й відображає сторінку на якій представлено цей список та кнопку для видалення категорій	Дочекатись завантаження сторінки
14	Перехід до корегування даних товарів	Програма формує список наявних товарів й відображає сторінку на якій представлено цей список та кнопку для видалення категорій, також вона формує представлення для фільтрації даних та відображає його	Дочекатись завантаження сторінки
15	Відображення списку замовлень	Програма формує список замовлень користувачів	Дочекатись завантаження сторінки
16	Перегляд списку інформативних сторінок	Після завантаження програма очікує на події, що викликає користувач	Змінити порядок сторінок у списку, додати нову сторінку, редагувати обрану, видалити обрану, Переглянути інформації сторінки, перейти до іншого пункту меню
17	Додання сторінки	Після заповнення полів про сторінку та формування її тіла користувача зберігає дані сторінки до бази даних	Дочекатись повідомлення про результат збереження даних

Продовження таблиці 8.1

1	2	3	4
18	Видалення запису	Програма видає повідомлення про підтвердження видалення	Підтвердити видалення, або скасувати його, дочекатись завантаження оновленої сторінки
19	Редагування обраної категорії	Користувач змінив дані рядку з назвою категорії	Дочекатись повідомлення про результат збереження даних
20	Додання категорії	Користувач ввів дані рядку з назвою категорії та натисну кнопку додати нову категорію	Дочекатись повідомлення про результат збереження даних
21	Видалення продукту	Програма видає повідомлення про підтвердження видалення	Підтвердити видалення, або скасувати його, дочекатись завантаження оновленої сторінки
22	Редагування продукту	Користувач заповнив поля про товар, додав його фото та галерею (за потреби) й натиснув кнопку збереження	Дочекатись повідомлення про результат збереження даних
23	Додання продукту	Користувач заповнив поля про товар, додав його фото та галерею (за потреби) й натиснув кнопку збереження	Дочекатись повідомлення про результат збереження даних

8.2 Опис керування діалогом

При переході за посиланням на сайт, користувачу відображається головна сторінка, меню та підвал сайту. Звідти користувач може переходити за посиланнями, що знаходяться у меню сайту, або звернутись за вказаними у підвалі сайту контактами.

На початку роботи в меню сайту користувачу доступні можливості переходу на інформаційні сторінки, що вказані в списку, що випадає, та на сторінку авторизації, чи головну сторінку. Головну сторінку й інформаційні сторінки заповнює адміністратор сайту, тому їх вміст та кількість інформаційних сторінок може змінюватись. Ще одним пунктом меню є посилання на сторінку авторизації користувача, при переході на нього відображається сторінка авторизації з полями

логіну та пароллю й можливостями авторизації чи реєстрації нового користувача. Також на всіх етапах роботи з програмою в зоні покупця користувачу доступні категорії товару, при переході на які відображається товар відповідної категорії.

На сторінці реєстрації користувач вводить дані про себе й підтверджує реєстрацію, після чого його повертають на сторінку авторизації.

При обранні перегляду товару можливості користувача при роботі з товарами змінюються в залежності від його авторизації, якщо він не увійшов до системи, то замість кнопки додати до корзини у деталях обраного товару буде відображено кнопку переходу на сторінку авторизації, якщо ж користувач зареєстрований, то до пунктів меню буде додано перехід до кошику, де будуть відображатись додані товари.

У кошику користувач може змінювати кількість обраних товарів та розрахуватись за нього, якщо оплата не виконана, але замовлення зареєстроване то користувач може зателефонувати за вказаними контактами й дізнатись про те як отримати товар та сплатити його вартість. Також коли користувач увійшов у систему йому доступні посилання на редагування профілю й посилання на історію його замовлень, де можна дізнатись номер замовлення за яким адміністратор зможе переглянути деталі по замовленню покупця.

Якщо користувач увійшов як адміністратор, то йому доступні всі можливості зареєстрованого користувача, крім редагування профілю, перегляду своїх замовлень, додання товарів до кошику та роботи з самим кошиком.

У адміністратора в меню з'являється додаткове посилання на перехід до адміністративної зони, при натисканні на яке він переходить на головну сторінку адміністративної зони.

Меню адміністративної зони містить посилання на перехід до зони покупця, роботу з інформаційними сторінками сайту, бічною сторінкою, каталогами, товарами, й замовленнями.

При роботі з інформаційними сторінками адміністратор може змінювати їх порядок, додавати нові, редагувати й видаляти існуючі. Виключенням є домашня сторінка, для якої можливо лише змінювати вміст.

Бічна сторінка є загальною для всіх й буде відображатись на всіх інформаційних сторінках в яких було обрано пункт її відображення.

При роботі з категоріями користувач може їх додавати, редагувати та видаляти, на відмінну від інших записів для редагування категорії користувачу достатньо натиснути на назву категорії та ввести нову її назву.

При роботі з товарами як в адміністративній зоні так і в зоні покупця виводиться блок фільтрації даних, в якому вказується мінімальна та максимальна ціна товару для пошуку, також якщо поле назва товару не заповнено, то пошук ведеться лише за поточною категорією товару й вказаною ціною, якщо поле назви заповнено, то будуть знайдені товари, які містять введений текст в своїй назві. Для виконання пошуку необхідно натиснути на кнопку пошуку.

При роботі з продуктами адміністратор отримує можливість додання нового товару, видалення та редагування існуючого. При редагуванні адміністратору доступне додаткове поле додання фотографій товару для галереї товару.

В пункті замовлень адміністратор може переглянути поточні замовлення, що виконували користувачі.

Вийти з сайту користувач може в будь-який момент, достатньо його закрити. Авторизація користувача буде зберігатись протягом певного проміжку часу, після чого його буде роз'єднано.

8.3 Формування екранів

Після опису всіх можливостей сайту відобразимо на рисунках 8.1 – 8.24 спроектовані сторінки сайту.



Рисунок 8.1 – Стартова сторінка сайту



Рисунок 8.2 – Інформативна сторінка сайту та список, що випадає

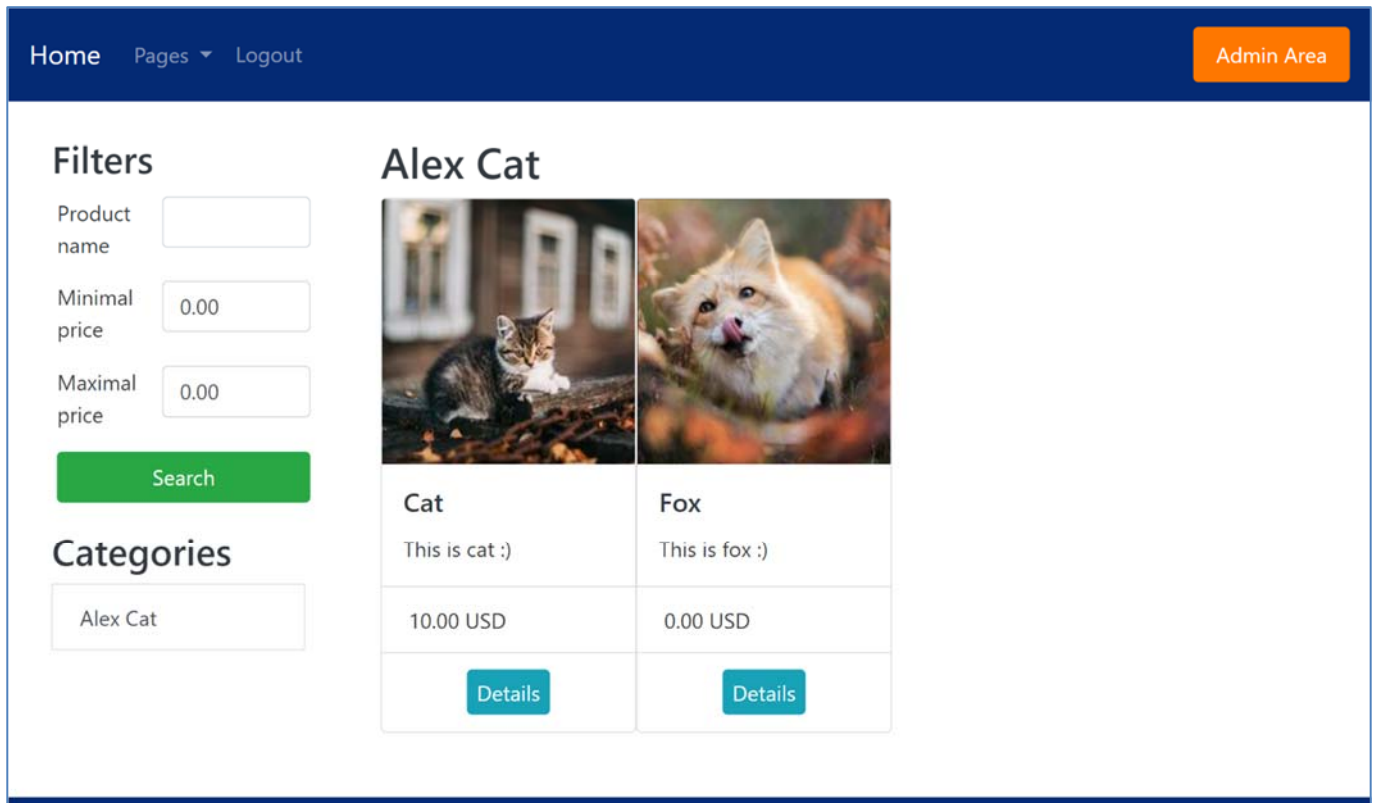


Рисунок 8.3 – Товари в магазині

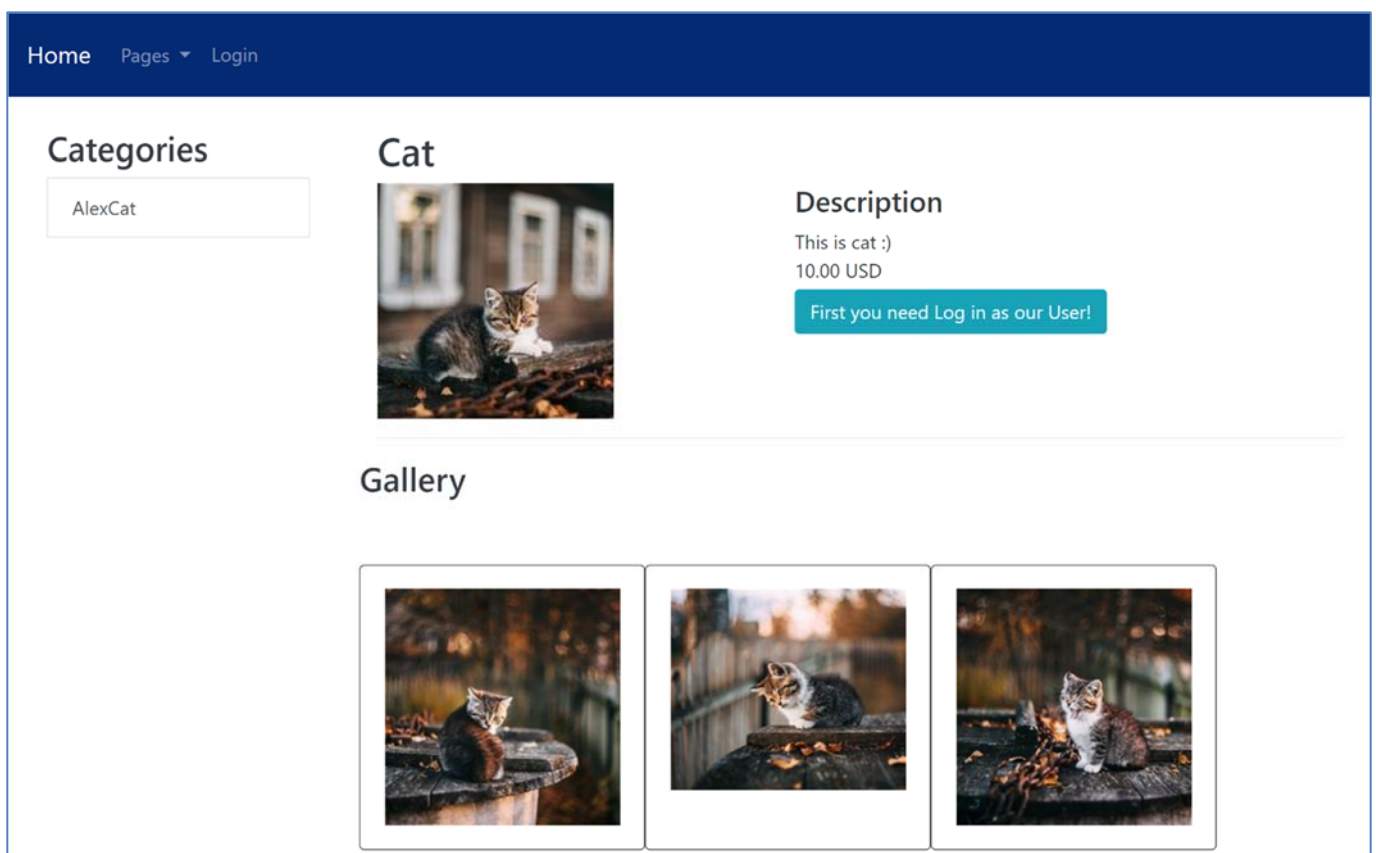


Рисунок 8.4 – Опис товару для незареєстрованих користувачів

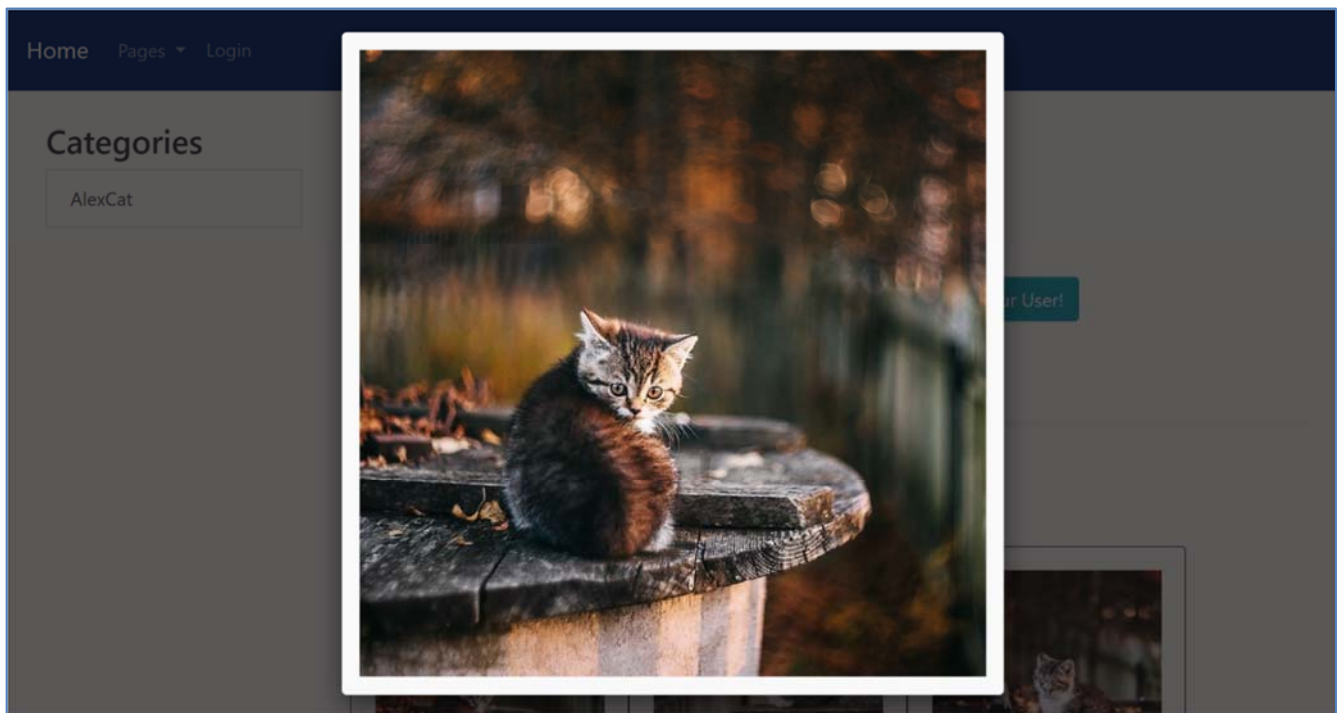


Рисунок 8.5 – Перегляд фото з галереї

Home Pages Login

Categories

AlexCat

Login

Login

Alex

Password

.....

Remember Me

☐

Login Create an account

Contacts

+38 (099) 329-57-10

Kropyvnytskyi region, Olexandria, street Cossack Way 89

agrosnabmarket@gmail.com

Advice

АГРОСНАБ

Рисунок 8.6 – Сторінка авторизації

Home Pages Login

Categories

AlexCat

Create Account

First name
Александр

Last name
Гетманенко

Email
alexandergetmanenko1@gmail.com

Login
New User

Password
.....

Confirm password
.....

Create

Рисунок 8.7 – Сторінка реєстрації користувача

Home Pages Login

Categories

AlexCat

Login

Successfully registered

Login

Password

Remember Me ☐

Login Create an account

Рисунок 8.8 – Повідомлення про успішну реєстрацію

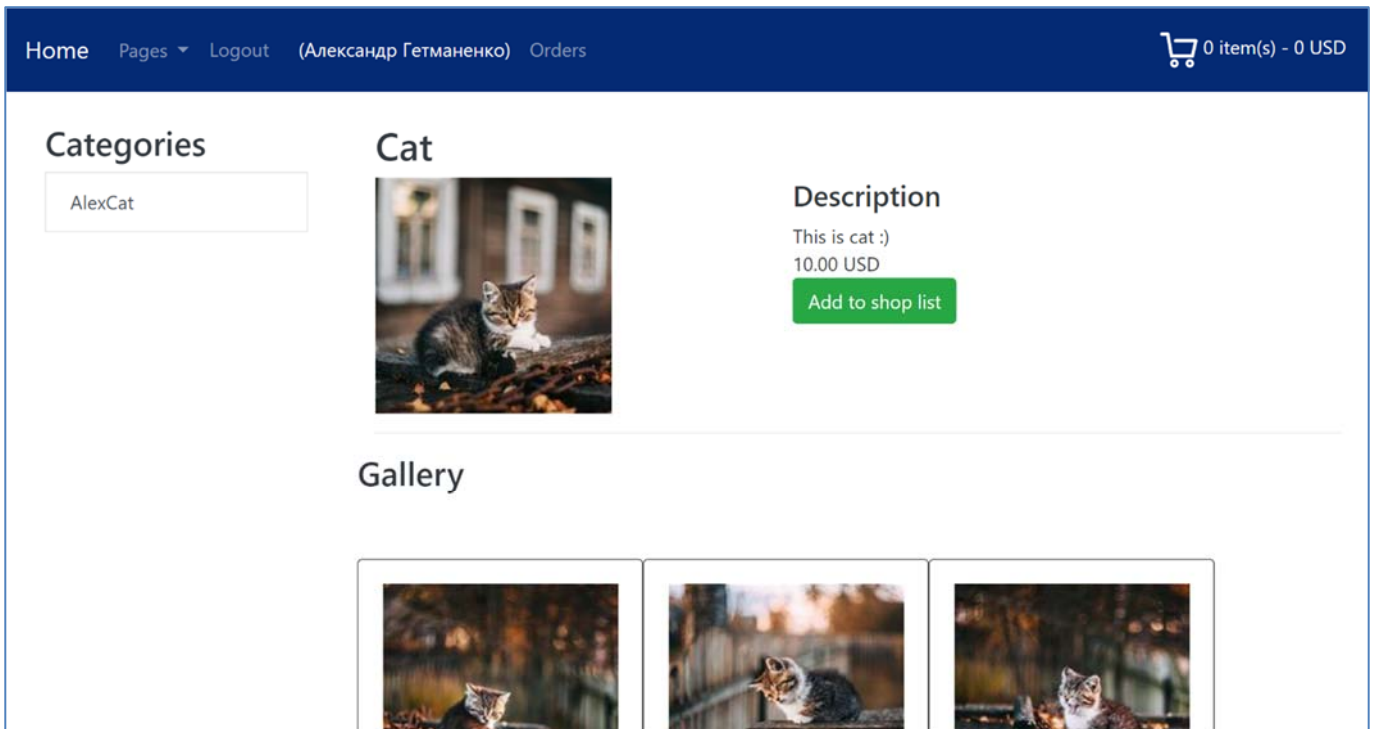


Рисунок 8.9 – Опис товару для зареєстрованих користувачів

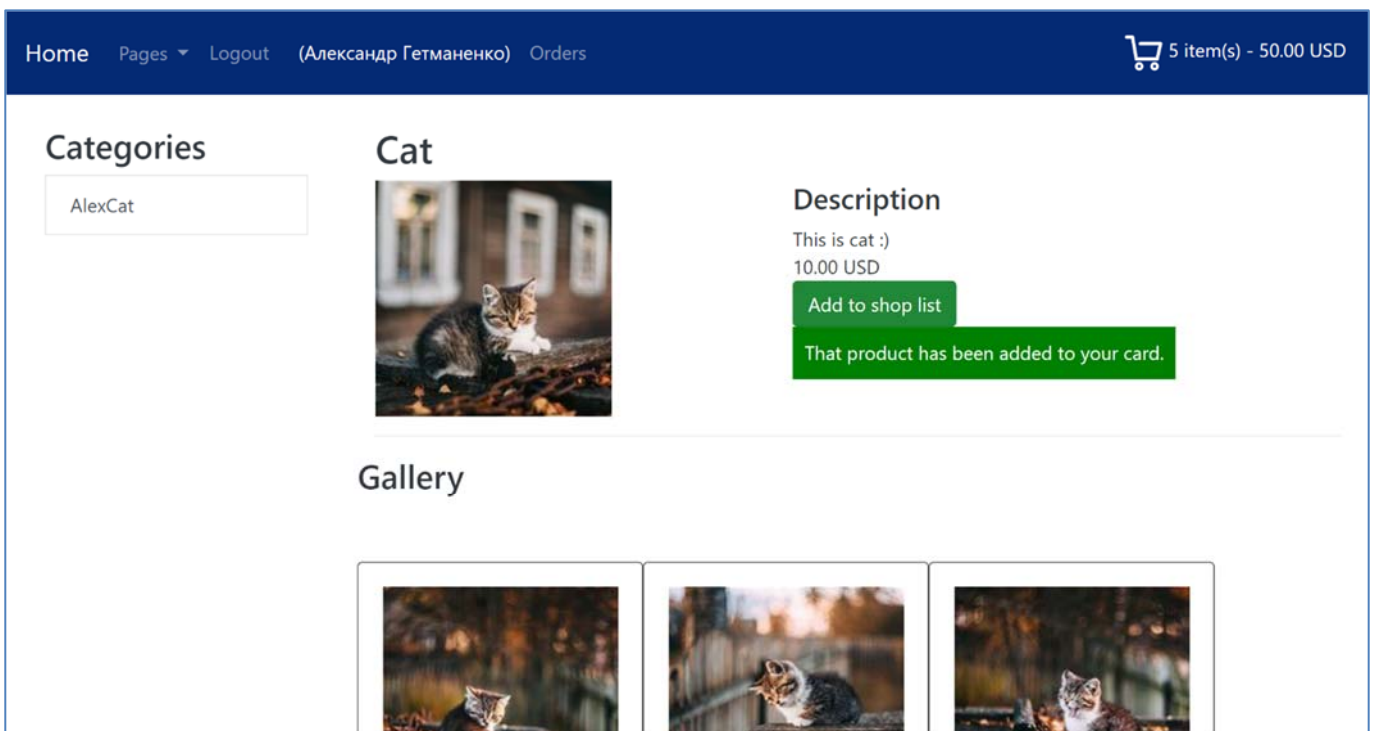



Рисунок 8.10 – Повідомлення про додання товару та зміна даних корзини

[Home](#) [Pages](#) [Logout](#) (Александр Гетманенко) [Orders](#)  5 item(s) - 50.00 USD

Categories

AlexCat

Your Profile

First name

Last name


Email

Login

Password

Confirm password

Рисунок 8.11 – Сторінка редагування профілю користувача

[Home](#) [Pages](#) [Logout](#) (Александр Гетманенко) [Orders](#)  5 item(s) - 50.00 USD


Categories

AlexCat

Orders

There are no orders

Contacts






 +38 (099) 329-57-10
 Кіровоградська область, Олександрія, вулиця Козацька 89
 agrosnabmarket@gmail.com

Рисунок 8.12 – Сторінка історії замовлень

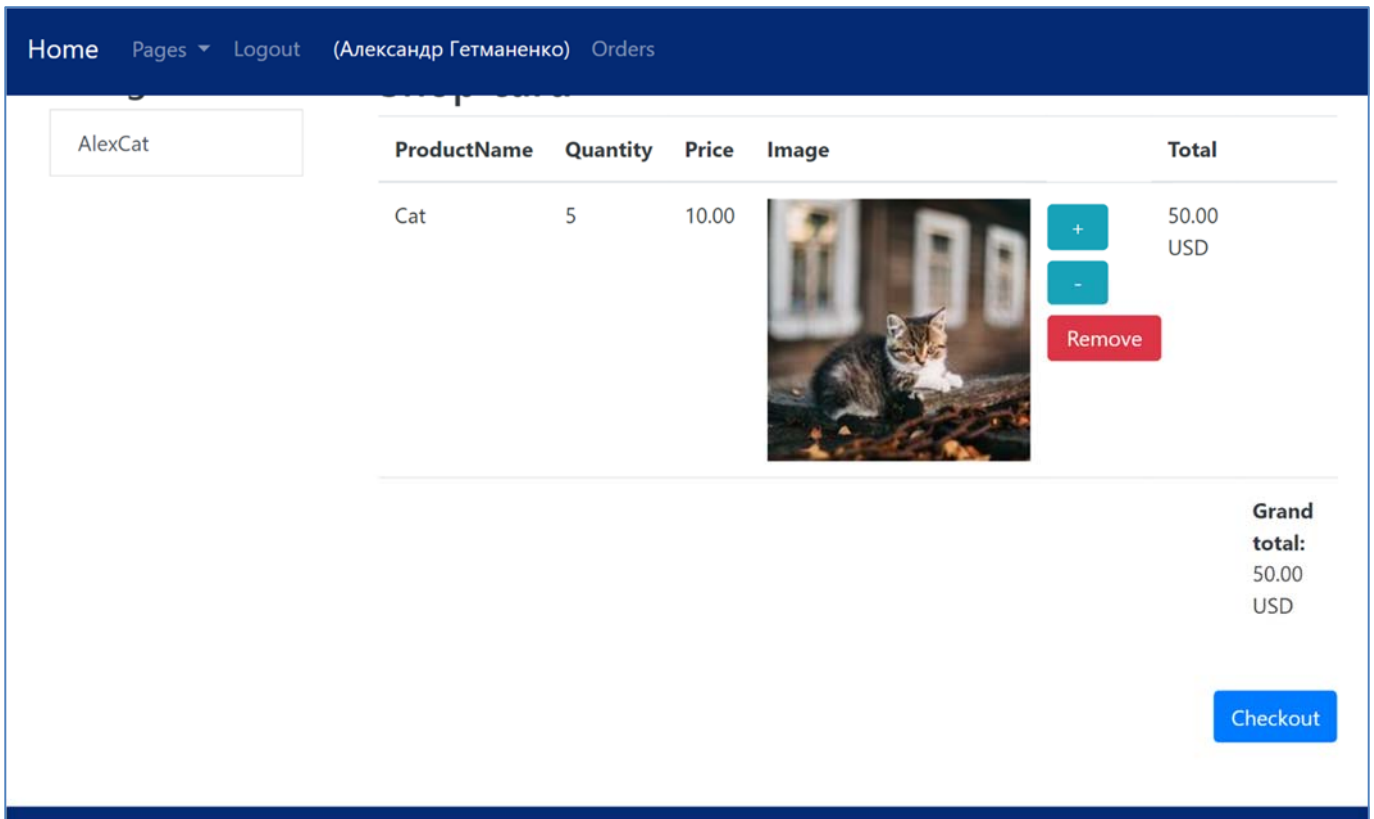


Рисунок 8.13 – Сторінка корзини товарів

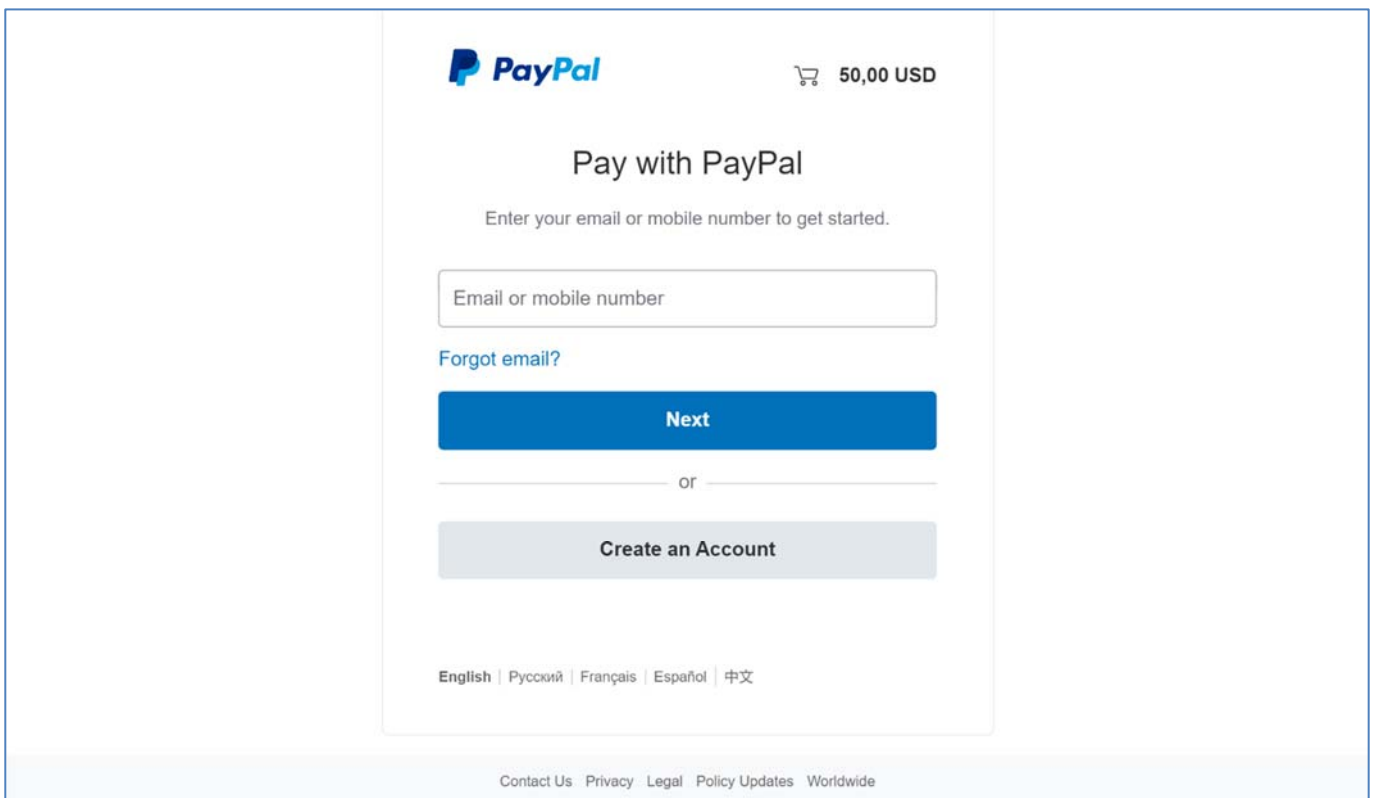


Рисунок 8.14 – Сторінка оплати товару

[Home](#)
[Pages](#)
[Logout](#)
(Александр Гетманенко)
[Orders](#)

5 item(s) - 50.00 USD

Categories

AlexCat

Orders

Order Number	Order details	Created At	Total
2	Cat x 5	5/30/2022 7:19:09 PM	50.00

Рисунок 8.15 – Сторінка історії замовлень

[Dashboard](#)
[Menu](#)
[Shopping card](#)

Dashboard

NO IMAGE AVAILABLE

It is a long established fact that a reader will be distracted by the readable content of a page when ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like). Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem

NO IMAGE AVAILABLE

It is a long established fact that a reader will be distracted by the readable content of a page when ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like). Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem

NO IMAGE AVAILABLE

It is a long established fact that a reader will be distracted by the readable content of a page when ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like). Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem

Contacts

+38 (099) 329-57-10

Kropyvnytskyi region, Olexandria, street Cossack Way 89

agrosnabmarket@gmail.com

Advice

Рисунок 8.16 – Головна сторінка адміністративної зони



Рисунок 8.17 – Вигляд головної сторінки покупця в режимі адміністратора

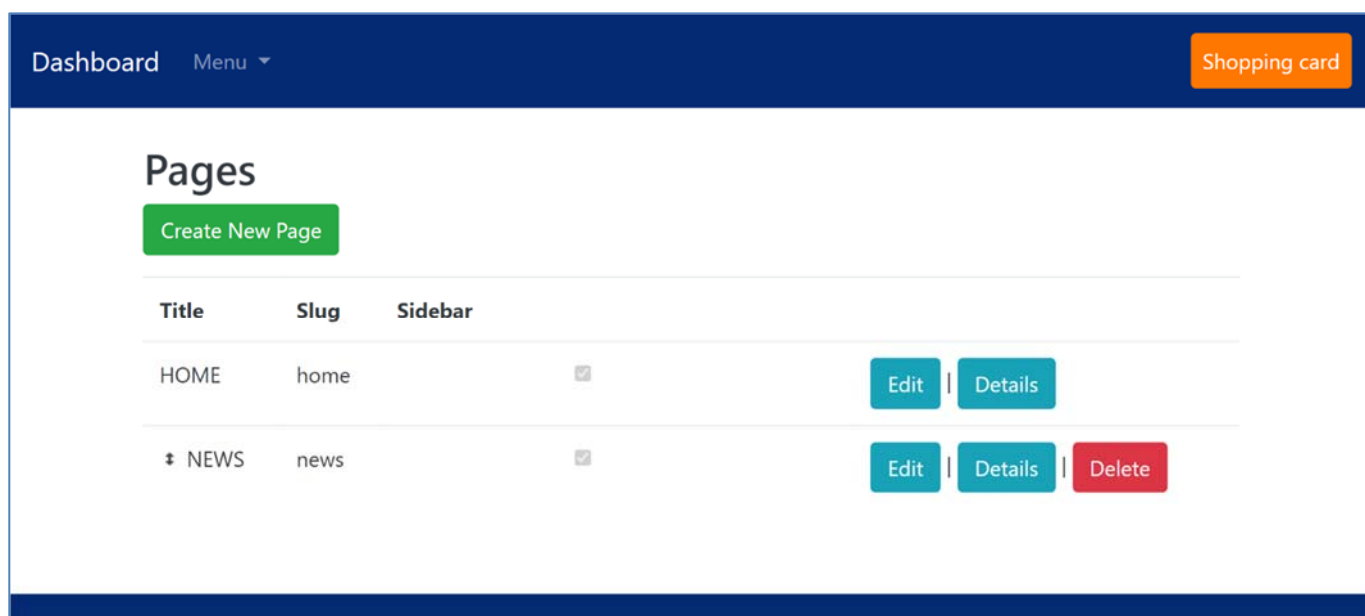


Рисунок 8.18 – Сторінка роботи з інформативними сторінками


[Dashboard](#) [Menu](#) [Shopping card](#)

PageDetails

Title
HOME

Slug
home

Body



News 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras pellentesque justo vitae ipsum faucibus, sed sagittis sem viverra. Vestibulum iaculis, felis a pulvinar imperdiet, magna metus hendrerit magna, in fringilla arcu magna a mi. Pellentesque finibus cursus orci, ut fermentum ligula aliquam vel. Quisque eu sollicitudin nibh. Sed a augue ipsum. Maecenas et turpis neque. Suspendisse varius, ipsum nec euismod accumsan, ante sem porta ex, vel interdum dolor risus blandit nibh. Sed ac felis nisl. Ut at ipsum eu metus pharetra mattis. Sed aliquam lectus eu rhoncus porta. Phasellus consequat posuere suscipit. Integer tincidunt feugiat tincidunt. Nam scelerisque lorem id volutpat pretium. Aliquam magna libero, faucibus sit amet volutpat vitae, accumsan volutpat elit.

News 2

Mauris id sem in nibh lacinia egestas non ut risus. Cras dignissim semper tellus, eget volutpat sapien efficitur sed. Sed consectetur eros quis dictum imperdiet. Donec quis facilisis enim. Duis lorem erat, bibendum vel nibh a, vulputate bibendum nisl. Integer in ipsum enim. Etiam nunc est, iaculis sit amet volutpat at, cursus id ipsum. Suspendisse eget urna et neque malesuada aliquam ut sed quam. Nam diam orci, euismod sit amet nisl vitae, dignissim cursus diam. Duis ultricies cursus gravida. Pellentesque pharetra tellus neque. Maecenas congue metus a nunc tincidunt, id efficitur ipsum porttitor. Duis tincidunt et nisl ut placerat. Nullam quis viverra risus, vitae pellentesque velit. Nulla pretium, dolor at pulvinar rhoncus, nisi neque tincidunt ante, vitae ultricies lacus nibh ut ante.

News 3

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Duis a iaculis nibh, non dignissim lacus. Donec at congue ligula, eget tempor massa. Sed facilisis odio ac tortor laoreet molestie. Curabitur sollicitudin auctor erat eget efficitur. Ut commodo aliquet magna, et lobortis velit vestibulum eget. Nulla in condimentum tellus. Nulla sed ex luctus, convallis metus nec, dictum dui. Suspendisse faucibus sit amet dolor vitae accumsan. Duis semper tellus ullamcorper, dignissim dolor congue, finibus lorem. Nam vulputate nisi sed enim dapibus, nec aliquam tellus maximus. Nullam et massa urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In commodo mauris

[Edit](#) | [Back to Pages](#)

Рисунок 8.19 – Сторінка перегляд даних інформативної сторінки

Dashboard Menu ▾ Shopping card

Categories

Category Name

Add a new category

Name

That title is taken Delete

AlexCat

Рисунок 8.20 – Спроба змінити категорію на вже існуючу

Dashboard Menu ▾ Shopping card

Filters

Product name

Minimal price


Maximal price

Search

Products

All ▾

Add new product



Cat

AVAILABLE
This is cat :)

10.00

Edit DISABLE

1

Рисунок 8.21 – Сторінка роботи з товарами в адміністративній зоні

Dashboard

Menu

Shopping card

Edit Product

Name

Cat

Description

This is cat :)

Price


10.00


Category

Alex Cat

Image

W1jYZkWaHqA.jpg





NO IMAGE AVAILABLE

Выберите файл


Файл не выбран

Deleted


☐

Edit products


Drop files here to upload




Delete




Delete



Delete



Delete



Delete


[Back to products](#)

Рисунок 8.22 – Редагування інформації про товар

Dashboard
Menu
Shopping card

Product name
Minimal price
Maximal price
Search

All
Add new product



Cat

UNAVAILABLE

This is cat :)

10.00


Edit DISABLE

Рисунок 8.23 – Зміна стану товару на відсутній

Dashboard
Menu
Shopping card

Orders

Order Number	User name	Order details	Created At	Total
1	Alex	Cat x 2	5/29/2022 10:52:57 AM	20.00
2	New User	Cat x 5	5/30/2022 7:19:09 PM	50.00



Contacts

+38 (099) 329-57-10

Kropyvnytskyi region, Olexandria, street Cossack Way 89

agrosnabmarket@gmail.com

Advice

Рисунок 8.24 – Сторінка історії замовлень користувачів

9 ПОРЯДОК РОБОТИ З ПРОГРАМОЮ

У разі виникнення питань, пов'язаних з продуктом «Інтернет-магазин», можна подати запит за електронною адресою: alexandergetmanenko1@gmail.com. Підтримка з 10:00 до 16:00 у понеділок та четвер.

10 ПОВІДОМЛЕННЯ

В табл. 10.1 приведені повідомлення користувачу в залежності від ситуацій, що виникли, а також рекомендовані дії.

Таблиця 10.1 – Повідомлення системи

Текст	Адресат	Ситуація	Рекомендовані дії
1	2	3	4
Obligatory field	Користувач	Заповнення обов'язкових полів даних таких як пароль логін та інше	Правильно заповнити відповідне поле
Mail does not match	Користувач	Невірно введено електронну пошту	Правильно заповнити відповідне поле
That title already exist.	Адміністратор	Заповнення даних про сторінку в адміністративній зоні	Ввести назву якої не існує в системі
That slug already exist.	Адміністратор	Заповнення даних про сторінку в адміністративній зоні	Ввести назву якої не існує в системі
The page doesn't exist.	Адміністратор	Редагування неіснуючої сторінки	Повернутись до обрання сторінок
You have deleted a page	Адміністратор	Видалення сторінки в адміністративній зоні	Переглянути
You have edited the sidebar	Адміністратор	Успішне редагування бічної панелі	Переглянути
Title taken	Адміністратор	Спроба зберегти категорію з ім'ям, що вже існує	Ввести назву якої не існує в системі
You have deleted a category	Адміністратор	Успішне видалення категорії	Переглянути
Category has been renamed	Адміністратор	Успішне перейменування категорії	Переглянути
That product name is taken	Адміністратор	Спроба зберегти продукт з ім'ям, що вже існує	Ввести назву якої не існує в системі
Wrong image extension	Адміністратор	При завантаженні файлу було обрано файл, що не є зображенням	Повторити спробу додати файли з одним із розширень .jpg, .jpeg, .gif, .png
That product doesn't exist	Адміністратор	Редагування неіснуючого продукту	Повернутись до обрання продукту

1	2	3	4
You have edited the product	Адміністратор	Успішне збереження продукту	Переглянути
You have deleted a product from gallery	Адміністратор	Успішне видалення продукту з галереї продуктів	Переглянути
Password don't match	Користувач / Адміністратор	Поля пароль та підтвердження паролю містять різні дані	Заповнити поля паролів однаковими значеннями
Login is taken	Користувач	Користувач намагається створити або зберегти акаунт з вже наявним логіном	Повторювати спробу з новим логіном поки не буде введений такий, що не існує в системі
Successfully registered	Користувач	Користувач правильно заповнив всі поля форми реєстрації та підтвердив реєстрацію	Переглянути
Not correct login or password	Користувач / Адміністратор	Неправильно заповнено поля форми авторизації та була спроба увійти у систему	Ввести правильні дані для авторизації або зателефонувати за посиланням в розділі контактів внизу сайту для отримання консультації чи відновлення паролю
Password is taken	Користувач	Користувач намагається змінити пароль на пароль що зараз є в системі	Ввести пароль, що відрізняється від поточного
You have edited your information	Користувач	Користувач правильно заповнив поля форми профілю та підтвердив збереження	Переглянути
Your shop card is empty	Користувач	Користувач перейшов у порожню корзину товарів	Переглянути або додати товар до корзини

Продовження таблиці 10.1

1	2	3	4
Category name must be at least 3 characters long	Адміністратор	Адміністратор намагається створити категорію, назва якої менше 3-х символів	Ввести назву кількість символів якої більше 3-х
You have added a product	Адміністратор	Успішне додання товару	Переглянути
The category has been added	Адміністратор	Успішне додання категорії	Переглянути
Confirm product deletion	Адміністратор	Підтвердження видалення продукту	Підтвердити видалення або скасувати
You will be redirected to paypal now	Користувач	Оплата товару, перехід до сторінки оплати	Переглянути повідомлення, очікувати завантаження відповідної форми авторизації, авторизуватись та сплатити
That product has been added to your card.		Успішне додання товару до кошика	Продовжити покупки або перейти до кошика для редагування кількості товару