Чернівецький політехнічний фаховий коледж

Відділення програмування та радіотехніки

Циклова комісія інженерія програмного забезпечення

Допускається до захисту

завідувач відділення

Андрій ШПАК

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до дипломного проєкту

молодшого спеціаліста

на тему: Інтернет-магазин зброї

Виконала: студент IV курсу, групи 541

спеціальність 121

Інженерія програмного забезпечення

спеціалізація

«Розробка програмного забезпечення»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Олексій ГЛОВАЦЬКИЙ

Керівник: \_\_\_\_\_\_\_\_\_ Антоніна БАСОВСЬКА

2023

**Анотація**

Пояснювальна записка складається з 49 сторінок, 33 рисунка, 1 додатоку, 5 джерел.

Об’єкт проєктування – програмне забезпечення «Інтернет-магазин зброї» .

Мета розробки – розробити програмне забезпечення для купівлі зброї.

Методами проєктування вибрано:

* HTML – стандартна мова розмітки гіпертексту;
* CSS – формальна мова опису зовнішнього вигляду документу, написаного з використанням мови розміток гіпертексту;
* Python – скриптова мова програмування ,була створена для генерації HTML сторінок на стороні web-сервера.

У результаті дипломного проєктування розроблене програмне забезпечення Інтернет-магазину зброї, в якому користувач має змогу зареєструватись, обирати будь-яку зброю, замовляти та оплачувати покупку.

Прогнозні припущення щодо розвитку об’єкта дослідження – вдосконалити та оптимізувати дизайн сайту інтернет-магазину, відслідковувати обсяги відвідувань.

Ключові слова: ПРОГРАМНИЙ ПРОДУКТ, БАЗА ДАНИХ, Інтернет-магазин зброї, Покупець,зброя,адміністратор.

**зміст**

[ВСТУП 4](#_Toc137807204)

[РОЗДІЛ 1. ПЕРЕДПРОЄКТНЕ ДОСЛІДЖЕННЯ 5](#_Toc137807205)

[1.1 Аналіз предметної області 5](#_Toc137807206)

[1.2 Аналіз існуючих аналогів 8](#_Toc137807207)

[1.3 Постановка задачі 13](#_Toc137807208)

[РОЗДІЛ 2. ТЕХНІЧНИЙ ПРОЄКТ 16](#_Toc137807209)

[2.1 Техні чне завдання 16](#_Toc137807210)

[2.2 Моделювання програмного продукту 18](#_Toc137807211)

[2.3 Моделювання даних 19](#_Toc137807212)

[2.4 Проєктування інтерфейсу 21](#_Toc137807213)

[РОЗДІЛ 3. Робочий проЄкт 26](#_Toc137807214)

[3.1 Засоби розробки 26](#_Toc137807215)

[3.2 Керівництво користувача 27](#_Toc137807216)

[ВИСНОВКИ 35](#_Toc137807217)

[перелік джерел 36](#_Toc137807218)

[Додатки 37](#_Toc137807219)

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

БД – база даних

ІС – інформаційна система

ІТ – інформаційні технології

ООП – об’єктно-орієнтоване програмування

ОС – операційна система

ПП – програмний продукт

СУБД – система управління базами даних

# ВСТУП

Розв'язання багатьох завдань базується на обробці інформації. Інформаційні системи створюються для полегшення обробки інформації. Автоматизованими називають інформаційні системи, в яких застосовують технічні засоби. Більшість існуючих інформаційних систем є автоматизованими.

Актуальністю дипломного проєкту, є те, що зараз високий попит на зброю

Метою дипломного проєкту є розробка програмного забезпечення для продажу та доставки зброї.

Об’єктом дослідження є інтернет-магазин, в якому здійснюється продаж зброї.

Предметом дослідження є робота з інформацією, технології створення автоматизованих систем.

Дипломний проєкт складається з програмного продукту та пояснювальної записки, що містить три частини.

У першому розділі пояснювальної записки до дипломного проєкту представлено аналіз предметної області, проаналізовано програмне забезпечення аналогічного призначення та проведено тезнологічне дослідження інновацій розробки.

У другому розділі вказано вимоги до програмного продукту, надане проєктування предметної області.

Третій розділ містить опис засобів розробки програмного продукту, документацію, що супроводжує програмний продукт.

У висновках підведено підсумки роботи над дипломним проєктом.

# РОЗДІЛ 1. ПЕРЕДПРОЄКТНЕ ДОСЛІДЖЕННЯ

**1.1 Аналіз предметної області**

На даний момент в Україні продовжується війна, спричинена державою-терористом РФ. І у громадян з’явилось велике бажання озброїтись для власної ж безпеки. Зброя, особливо напівавтоматична, завжди користувалась великим попитом у потенційних клієнтів, але раніше щоб купити собі звичайний карабін чи пістолет, потрібно було затратити дуже багато сил на пошуки спеціалізованого магазину, який може знаходитись навіть за сотні кілометрів від вашого місцезнаходження. З розвитком сучасних інформаційних технологій, клієнт може знайти зброю яка йому до вподоби, не виходячи при цьому з дому. Така політика дуже зручна як для покупців, так і для магазину тому що знижується більша кількість паперової роботи, автоматизується обробка замовлень, та є можливість аналізу товарів які є найбільш популярними серед клієнтів сервісу.

З початком російсько-української війни була прийнята остання редакція законодавства, яка передбачає певне пом'якшення питання зброї в Україні на період воєнного стану та дозволяє вбивати українських військових під час самооборони російськими військами. війська.

Щоб придбати зброю, необхідно пройти перевірку та отримати дозвіл поліції. На полюванні можна використовувати лише мисливські рушниці та гладкоствольні рушниці. Працівники судів, силових структур і журналісти можуть придбати короткоствольну травматичну зброю, що стріляє гумовими або іншими несмертельними кулями. Вогнепальною короткоствольною зброєю може володіти лише обмежена кількість осіб – переважно військові та силовики. Напівавтоматичну вогнепальну зброю з нарізними стволами можуть мати особи, які на придбання отримали дозвіл МВС України, за умови зберігання зброї в закритому місці з обмеженим доступом..[1]

Об’єкти для створення:

Об'єктом створення інтернет-магазину зброї є створення електронної торговельної платформи для продажу різноманітної зброї та відповідного спорядження, яка буде доступна клієнтам через Інтернет. Цей магазин може містити такі елементи як:

Каталог товарів: Магазин повинен мати детальний каталог товарів з описом та фотографіями кожного виду зброї та спорядження.

Кошик покупок: Клієнти мають змогу додавати товари до кошика покупок та замовляти їх в будь-який зручний для них час.

Система оплати: Магазин має мати безпечну та зручну систему оплати, яка дозволяє клієнтам здійснювати платіж за допомогою різних способів, таких як кредитні карти, електронні гроші тощо.

Система доставки: Магазин має мати систему доставки, яка забезпечує швидку та надійну доставку товару клієнтам.

Система підтримки клієнтів: Магазин повинен мати систему підтримки клієнтів, щоб допомагати їм з різними питаннями, такими як статус замовлення, повернення товарів, інформація про товар тощо.

Заходи забезпечення безпеки: Магазин повинен дотримуватися всіх правил та норм щодо продажу зброї та забезпечення безпеки клієнтів, зокрема здійснювати перевірку на здатність до придбання зброї.

В загальному, об'єктом створення інтернет-магазину зброї є створення безпечної, зручної та доступної платформи для продажу зброї та спорядження в Інтернеті.

Практичне значення розробки:

Розробка інтернет-магазину зброї має декілька практичних значень:

Зручність для клієнтів: Інтернет-магазин зброї надає зручний спосіб покупки зброї та спорядження через Інтернет, що дозволяє клієнтам з легкістю знайти необхідний товар та зробити покупку, не виходячи з дому.

Розширення ринку: Розробка інтернет-магазину дозволяє компанії збільшити свою потенційну аудиторію, оскільки товари можуть бути доступні для замовлення з будь-якого місця, де є Інтернет.

Ефективність витрат: Розробка інтернет-магазину зброї дозволяє знизити витрати на рекламу, встановлення та управління традиційним магазином.

Онлайн-аналітика: Розробка інтернет-магазину зброї надає можливість збирати дані та проводити аналітику про продажі товарів, що дозволяє компанії аналізувати свої продажі та оптимізувати свої бізнес-процеси.

Забезпечення безпеки: Розробка інтернет-магазину зброї дозволяє забезпечити безпеку продажу зброї та спорядження, оскільки він дозволяє встановити строгі контрольні процедури та обмеження на продаж певних видів зброї.

Отже, розробка інтернет-магазину зброї має практичну значимість як для клієнтів, так і для компаній, які займаються продажем зброї та спорядження.

Суб'єкти та об'єкти предметної області інтернет-магазину зброї включають наступні елементи:

1.Клієнти – люди, які відвідують інтернет-магазин з метою придбання зброї та збройних аксесуарів.

2.Адміністратори – люди, які мають доступ до адміністративної панелі магазину та відповідають за управління товаром та замовленнями.

3.Товари – зброя та збройні аксесуари, які продаються в інтернет-магазині.

4.Замовлення – запити від клієнтів на покупку товарів в інтернет-магазині.

5.Кошик – механізм, що дозволяє клієнтам додавати товари до замовлення та керувати ними перед оформленням замовлення.

6.Система платежів – механізм, який дозволяє клієнтам здійснювати оплату замовлень через інтернет.

7.Система доставки – механізм, що забезпечує доставку замовлень до клієнтів.

8.Категорії товарів – сортування товарів за категоріями, такими як ручна стрільба, мисливство, туризм тощо.

9.Країни та регіони – місця, до яких можливо доставити товари, а також місця, з яких клієнти можуть здійснювати покупки.

10.Фільтри та сортування – механізми, що дозволяють клієнтам швидко знаходити необхідні товари за допомогою певних критеріїв, таких як ціна, бренд, тип зброї тощо.

**1.2 Аналіз існуючих аналогів**

Предметну область було опрацьовано на основі інтернет-магазину «Stvol» та мілітарі-спільноти «PSDInfo»

«Stvol» – це український інтернет-магазин зброї, аксесуарів та боєкомплекту, а також спорядження, засобів індивідуального захисту. Магазин здійснює доставку по всій Україні та світу.

На стартовому екрані сайту можна побачити його логотип, та кнопки з посиланнями на інші елементи сайту (рисунок 1.1).

Верхній розділ головної сторінки веб сайту дозволяє:

* переглянути категорії доступної зброї та спорядження;
* дізнатись інформацію про тир, сервісне обслуговування зброї;
* переглянути соціальні мережі магазину та доступні контакти;
* зареєструватись на порталі сайту.

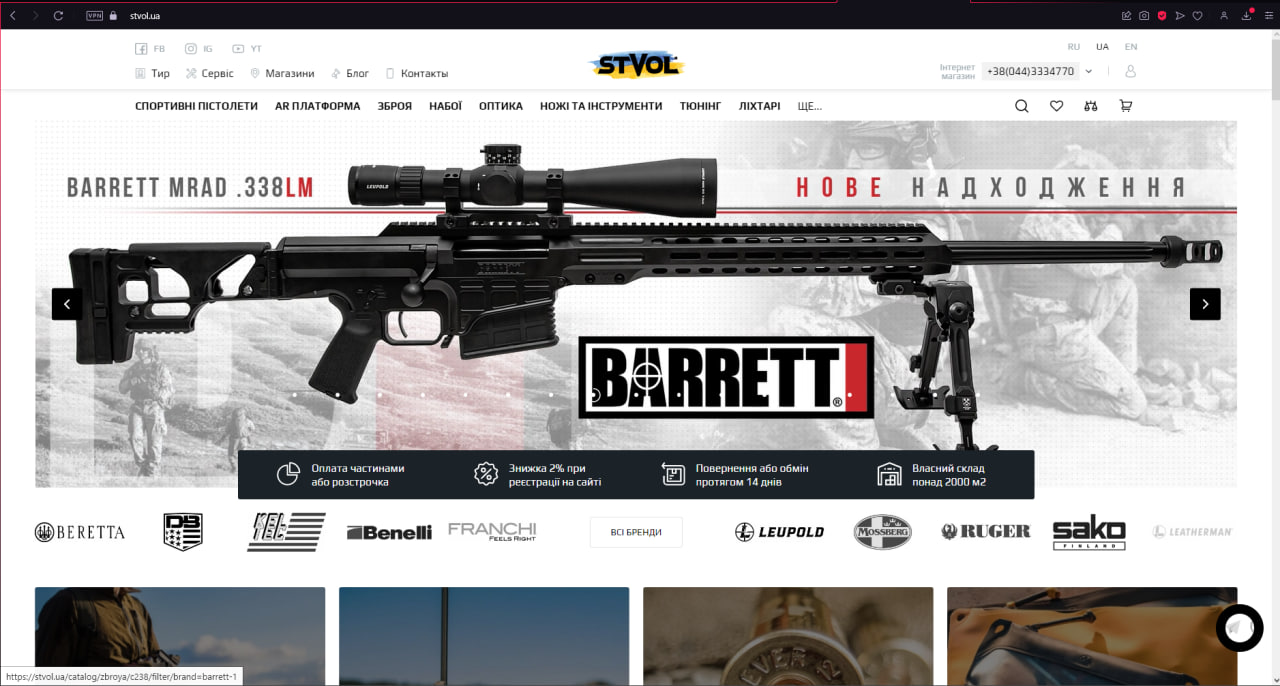


Рисунок 1.1 – Головна сторінка

На сайті реалізований зручний пошук завдяки категоріям, а також за ключовими словами. (рисунок 1.2).

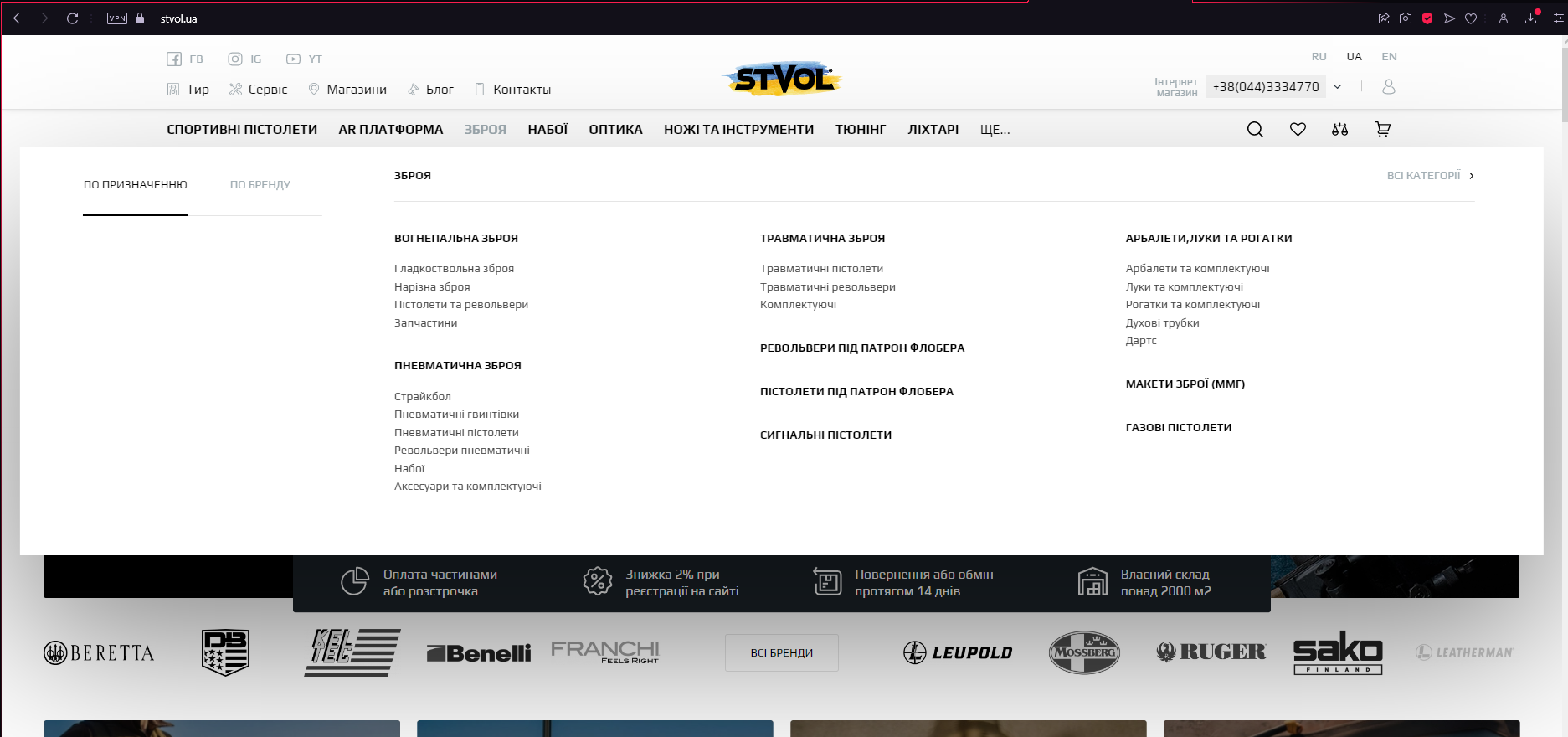


Рисунок 1.2 – Категорії зброї

При натисканні на товар, можна продивитися короткий опис, а також передивитися інші світлини (рисунок 1.3).

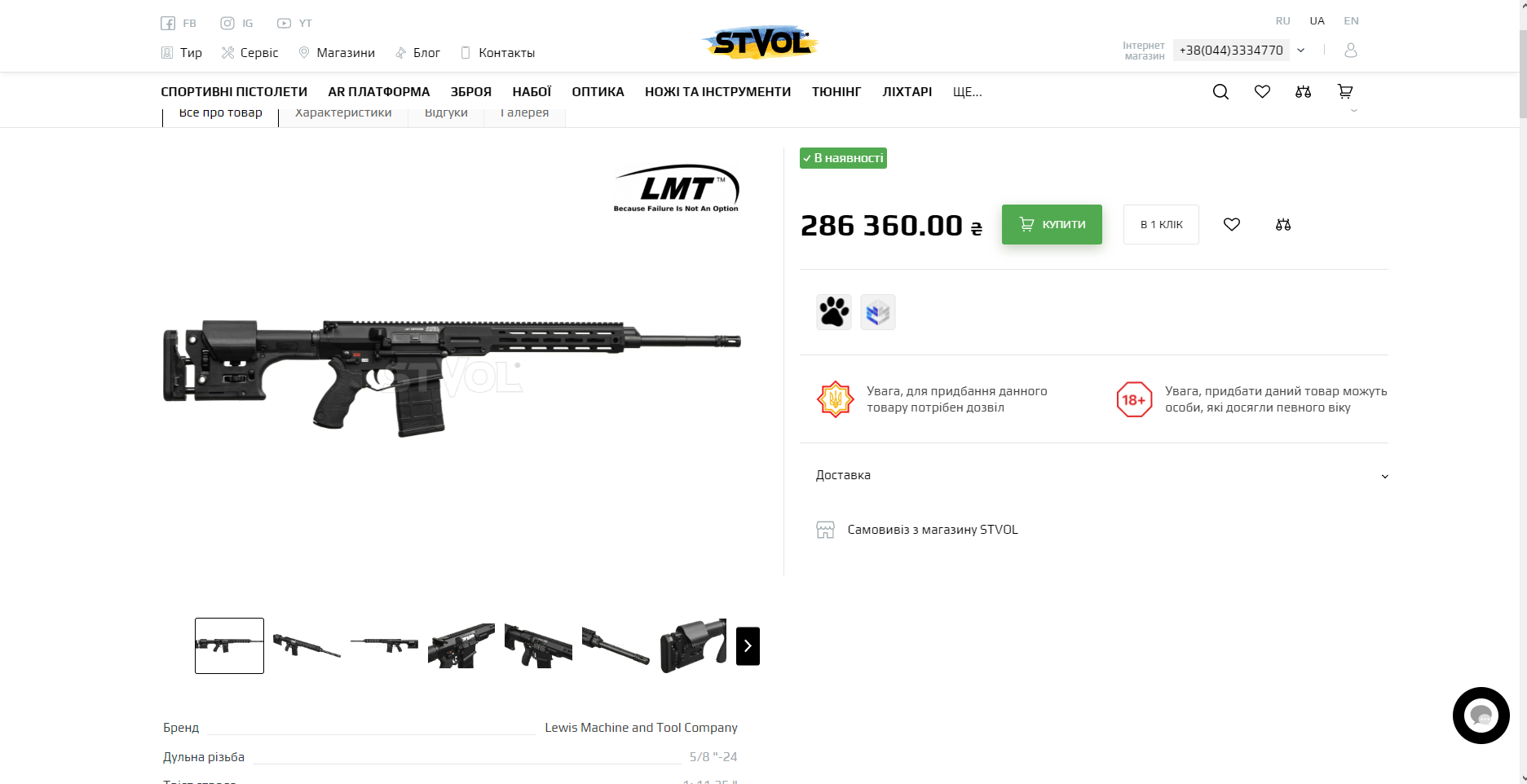


Рисунок 1.3 – Детальна інформація про зброю

При додаванні товару в кошик, ми попадаємо на сторінку оформлення замовлення, що дає нам можливість одразу замовити товар (рисунок 1.4).

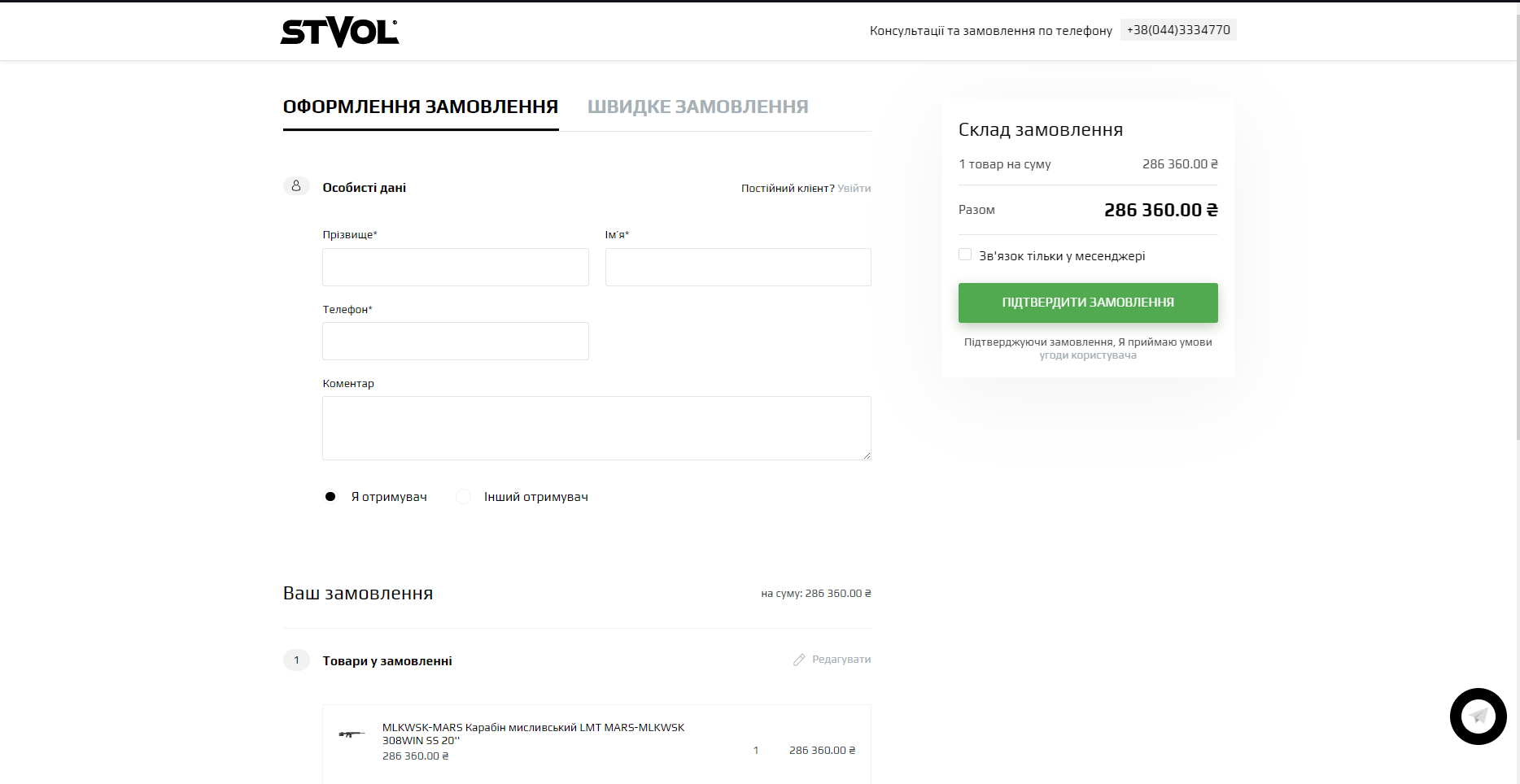


Рисунок 1.4 – Кошик і оформлення замовлення

«PSDInfo» – це мілітарі-спільнота, яка орієнтована на продаж спорядження, аксесуарів для зброї, засобів індивідуального захисту, шевронів та засобів догляду за зброєю, з можливістю доставки по всій Україні.

Верхній розділ головної сторінки веб сайту дозволяє:

* увійти або зареєструватись.
* переглянути каталог квітів;
* пошук;
* переглянути детальну інформацію про товар;
* переглянути товари в кошику і оформити замовлення(рисунок 1.5)

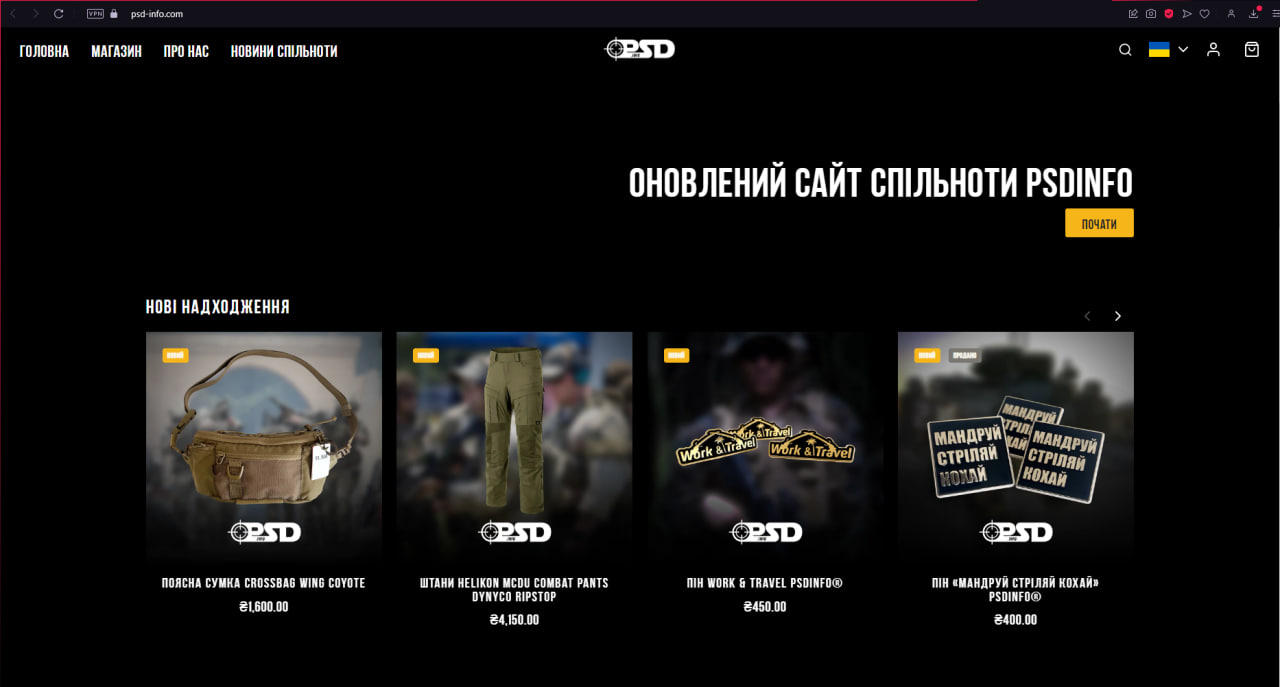


Рисунок 1.5 – Головна сторінка сайту

На сторінці авторизації ми маємо можливість авторизуватися в існуючий аккаунт, або створити новий (рисунок 1.6)

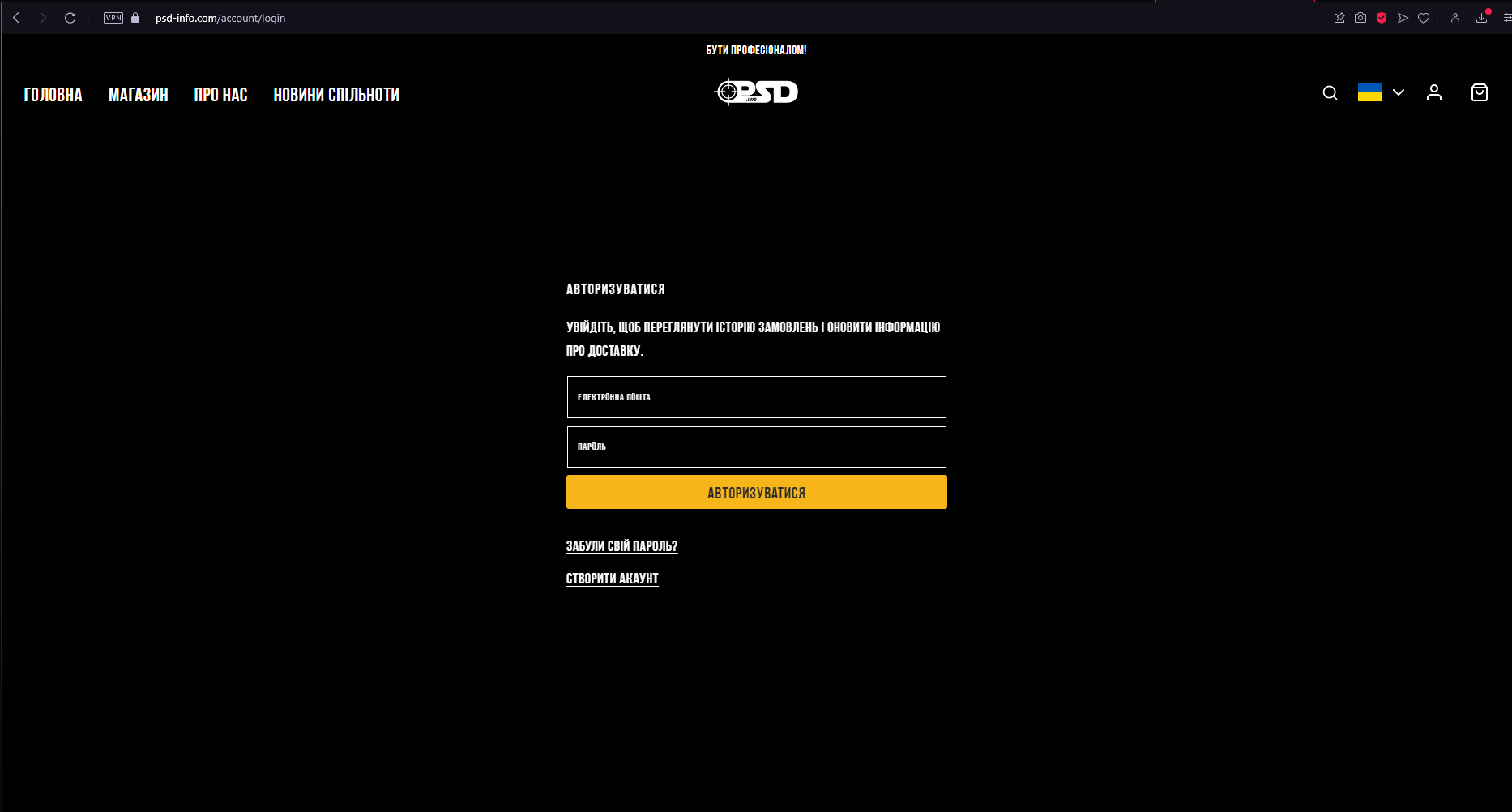


Рисунок 1.6 – Авторизація

На сторінці реєстрації ми маємо можливість створити аккаунт з повного нуля (рисунок 1.7)

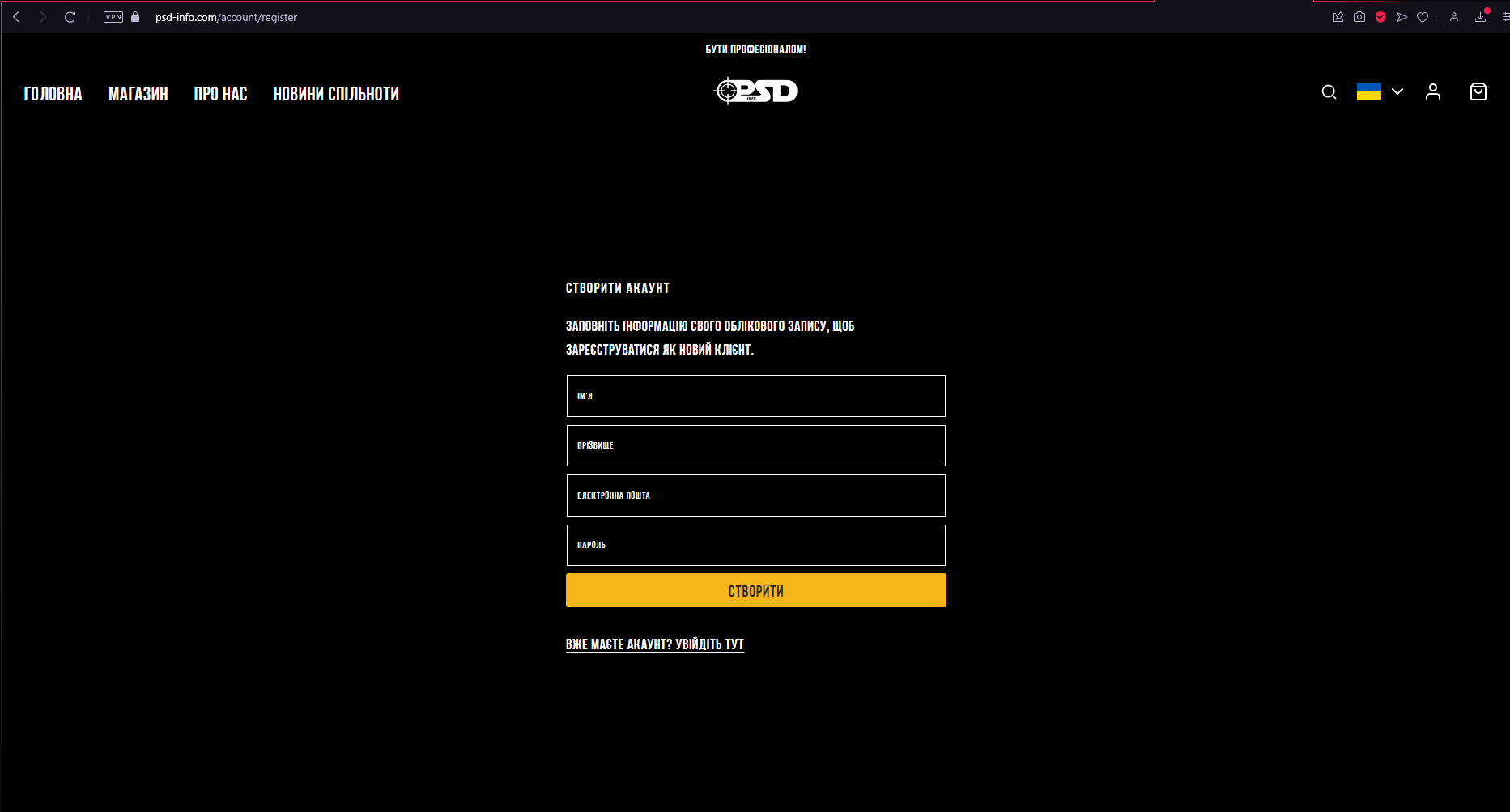


Рисунок 1.7 – Реєстрація

На сайті реалізований каталог, та пошук за ключовими словами, та фільтри, завдяки чому, можна легко здійснювати пошук (рисунок 1.8).

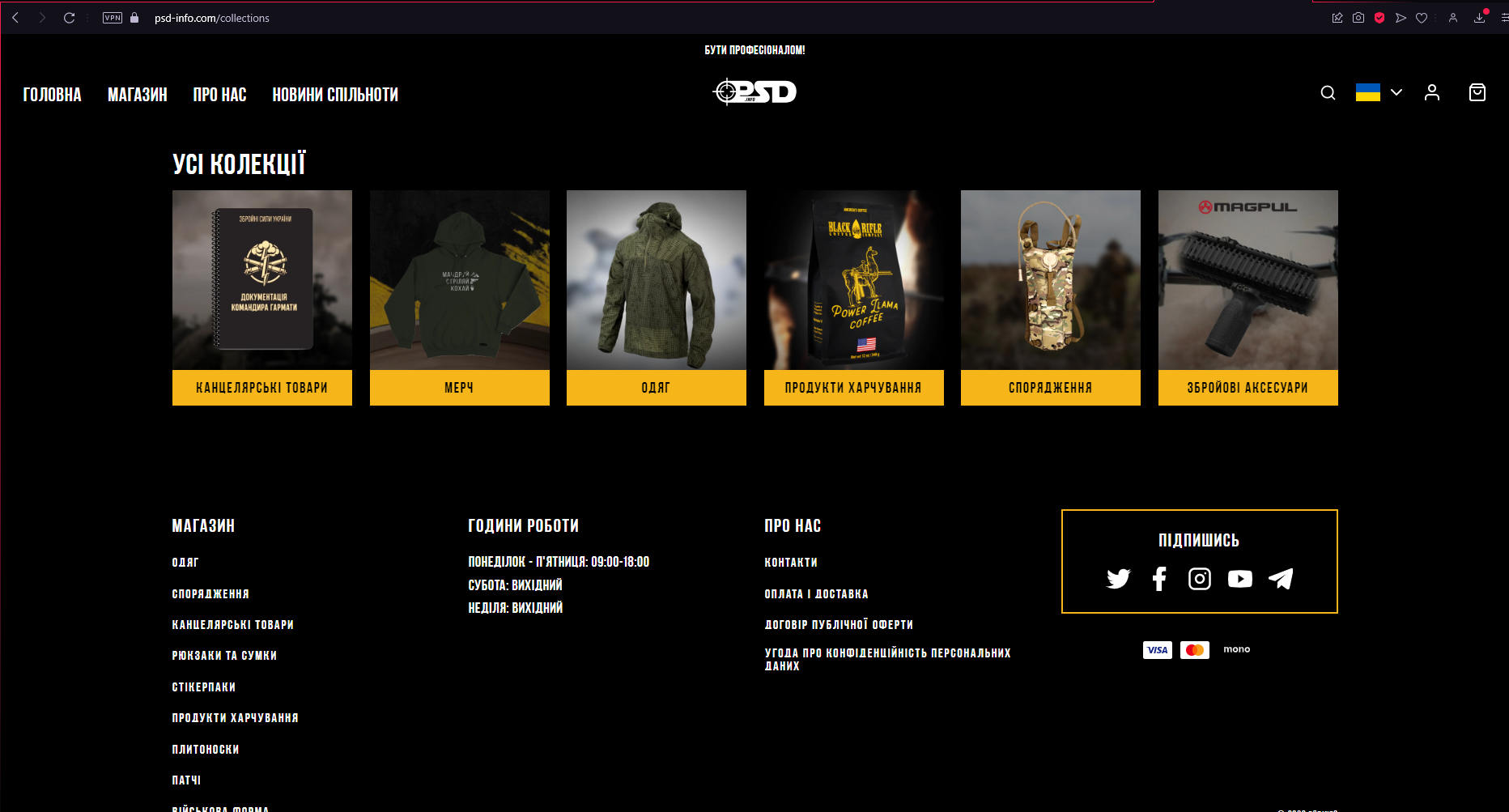


Рисунок 1.8 – Каталог товарів

При натисканні на товар, можна продивитися короткий опис, а також передивитися інші світлини (рисунок 1.9).

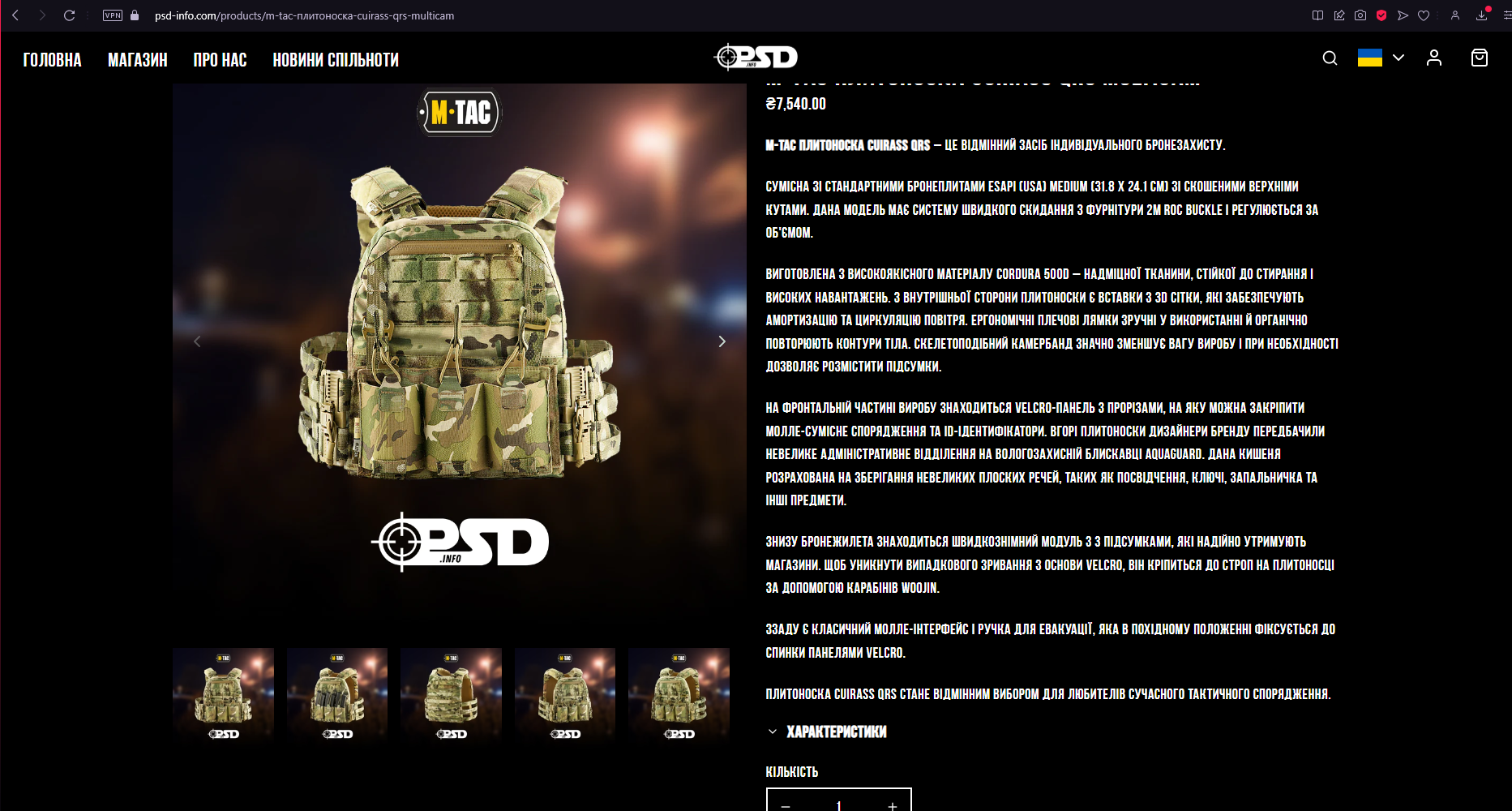


Рисунок 1.9 – Детальна інформація про товар

При додаванні товару в кошик, ми попадаємо на сторінку оформлення замовлення, що дає нам можливість одразу замовити товар (рисунок 1.10).

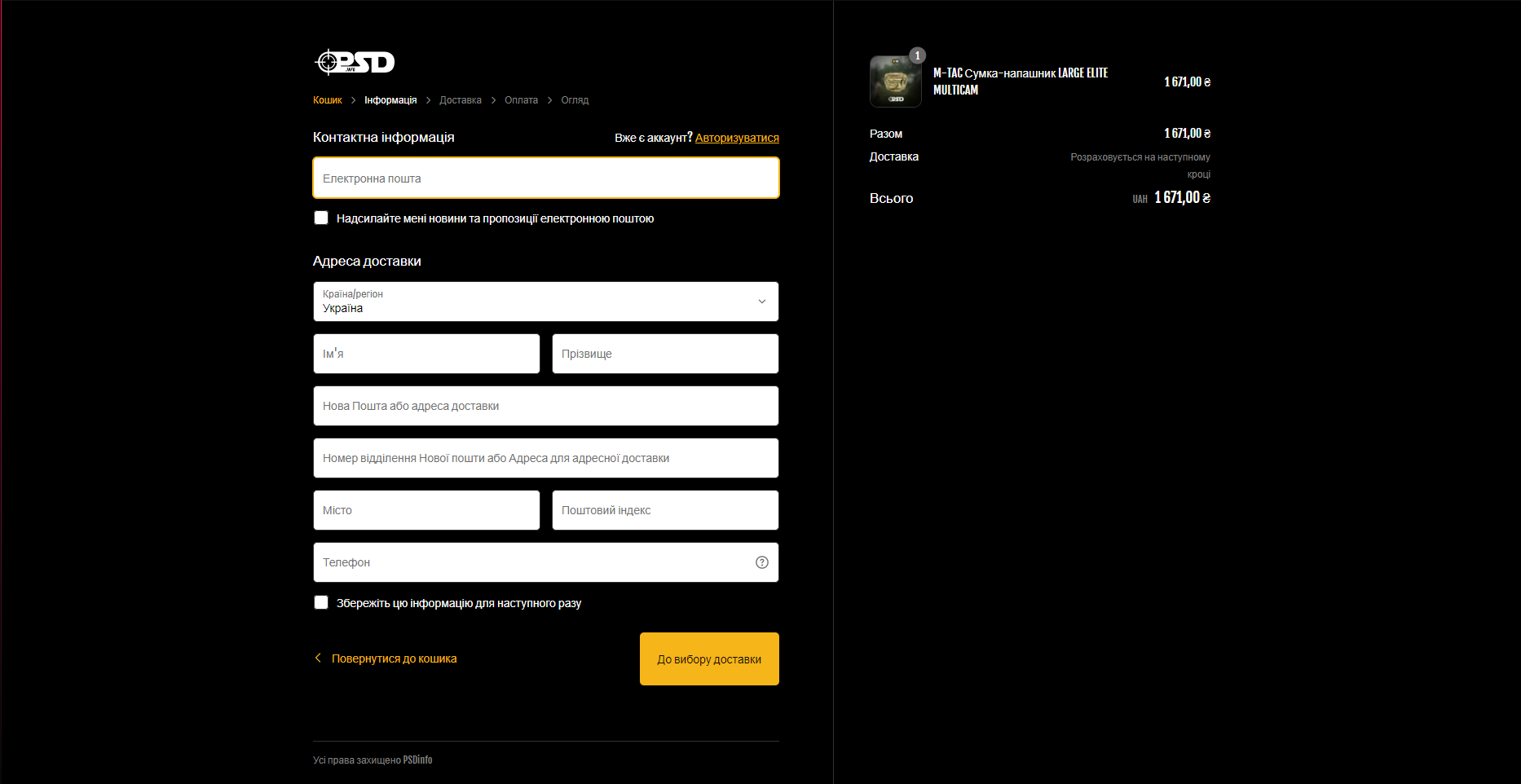


Рисунок 1.10 – Оформлення замовлення

**1.3 Постановка задачі**

Постановка задач для інтернет-магазину зброї включає наступні пункти:

Розробка зручного та інтуїтивно зрозумілого інтерфейсу для клієнтів, щоб забезпечити зручність покупок та збільшити ймовірність замовлення.

Забезпечення безпеки продажу зброї та спорядження, зокрема шляхом встановлення строгих контрольних процедур та обмежень на продаж певних видів зброї.

Розробка ефективної системи управління запасами, щоб забезпечити наявність необхідних товарів та уникнути надлишків, які можуть призвести до збитків.

Розвиток маркетингових стратегій для просування магазину та залучення нових клієнтів, включаючи використання соціальних мереж, контекстної реклами та пошукової оптимізації.

Забезпечення швидкої та надійної доставки товарів клієнтам, включаючи розробку оптимальної системи логістики та співпрацю з провідними транспортними компаніями.

Проведення регулярних аудитів та вивчення потреб ринку, щоб забезпечити вчасну реакцію на зміни та вдосконалювати продуктову лінійку.

Забезпечення високого рівня обслуговування клієнтів, включаючи швидку відповідь на запити та вирішення проблем, які можуть виникнути під час процесу замовлення.

Завданням дипломного проєктування було розробити інтернет-магазин зброї, за допомогою якого можна вибирати зброю, докуповувати аксесуари для неї та боєкомплект. Також розробити адміністративну частину для керування даними, доступ до якої має тільки адміністратор-розробник.

Основні дані для додавання товарів:

* Назва зброї
* Модель
* Калібр

Функціональні можливості:

1. Перегляд наявної зброї
2. Пошук
3. Додавання нових видів озброєння
4. Видалення
5. Авторизація адміністратора
6. Перегляд списку доступних товарів
7. Реєстрація користувача
8. Авторизація користувача
9. Додавання зброї в закладки
10. Замовлення товару
11. Система оплати
12. Простота в користуванні

Технічні характеристики апаратного забезпечення:

– операційна система – Windows 7 і вище;

– браузери Google Chrome, Opera, Mozilla;

– об’єм оперативної пам’яті – 128 Мб;

– місце на диску – 20 Мб.

Отже, постановка задачі для інтернет-магазину зброї має на меті забезпечення ефективного та безпечного продажу зброї та спорядження, забезпечення задоволення потреб клієнтів та збільшення прибутковості. І сформульовано та описано весь функціонал інтернет-магазину зброї.

# РОЗДІЛ 2. ТЕХНІЧНИЙ ПРОЄКТ

## 2.1 Технічне завдання

Призначення розробки: інтернет-магазин для купівлі зброї в режимі онлайн. Також реалізувати адміністративну панель, за допомогою якої можна додавати нові види зброї та видаляти їх. Забезпечити виконання функцій купівлі зброї. Користувачі матимуть вільний доступ тільки для переглядів продукції.

Програмне забезпечення що розробляється повинно вирішувати наступні задачі:

Розробити зручний та інтуїтивно зрозумілий інтерфейс, щоб гарантувати зручність покупок для клієнтів та збільшити кількість замовлень.

Забезпечити безпеку продажу озброєння та техніки, в тому числі шляхом встановлення суворих процедур контролю та обмежень на продаж певних видів зброї.

Розробити ефективні системи управління запасами для забезпечення адекватних поставок товарів та уникнення затоварювання, що призводить до збитків.

Розробити маркетингову стратегію, включаючи використання соціальних мереж, контекстної реклами та пошукової оптимізації, для просування магазину та залучення нових клієнтів.

Забезпечити швидку та надійну доставку товарів покупцям, в тому числі налагодити оптимальну логістичну систему та співпрацювати з великими транспортними компаніями.

Проводити регулярні аудити та дослідження ринку для забезпечення своєчасного реагування на зміни та вдосконалення асортименту продукції.

Забезпечення високого рівня обслуговування клієнтів, включаючи оперативне реагування на запити та вирішення будь-яких проблем, що можуть виникнути при виконанні замовлень.

Під час розробки програмного продукту можна використовувати стандартні програмні засоби, які застосовуються для програмування веб-сайтів. Щодо оцінки результатів розробки і визначення доцільності продовження робіт, потрібно надати такі матеріали:

Перелік файлів, що знаходяться на резервному носії.

Короткий опис роботи додатка та опис всіх файлів, необхідних для його роботи.

Список документів, включаючи технічне завдання, пояснювальну записку, додаток та методику тестування, керівництво для користувача та програміста.

Передпроєктні дослідження, включаючи аналіз предметної області та існуючих аналогів, технологічне дослідження інновацій у розробці програмного забезпечення.

Технічний проєкт, що включає технічне завдання, концептуальну, інформаційну та функціональну моделі, проєктування інтерфейсу.

Робочий проєкт, включаючи засоби розробки та документи для супроводження програмного забезпечення (інструкції для програміста та користувача), план тестування (розробка тестів та аналіз результатів тестування), висновки, список джерел та додатки.

Все програмне забезпечення та супроводжуюча технічна документація повинні відповідати чинному законодавству. Тестування проводиться відповідно до «Програми та методики тестування», яку розробляє та затверджує виконавець. Під час розробки програмного забезпечення виконуються наступні види випробувань:

* тестування елементів системи;
* тестування системи в цілому;
* дослідна експлуатація.

Після закінчення відповідного етапу робіт формується відповідний комплект документації. Завершення етапу розробки фіксується відповідним протоколом захисту дипломних проєктів.

## 2.2 Моделювання програмного продукту

Для проєктування інтернет-магазину була обрана клієнт-серверна архітектура, оскільки вона є найбільш підходящою для реалізації програмного забезпечення.

Музичний портал був розроблений за допомогою бази даних, яка складається з таблиць, які забезпечують функціонування інтернет-магазину. Таблиці сформовані за критеріями, які реалізовують постановку задачі.

Суб'єкти та об'єкти предметної області інтернет-магазину зброї включають наступні елементи:

1.Клієнти – люди, які відвідують інтернет-магазин з метою придбання зброї та збройних аксесуарів.

2.Адміністратори – люди, які мають доступ до адміністративної панелі магазину та відповідають за управління товаром та замовленнями.

3.Товари – зброя та збройні аксесуари, які продаються в інтернет-магазині.

4.Замовлення – запити від клієнтів на покупку товарів в інтернет-магазині.

5.Кошик – механізм, що дозволяє клієнтам додавати товари до замовлення та керувати ними перед оформленням замовлення.

6.Система платежів – механізм, який дозволяє клієнтам здійснювати оплату замовлень через інтернет.

7.Система доставки – механізм, що забезпечує доставку замовлень до клієнтів.

8.Категорії товарів – сортування товарів за категоріями, такими як ручна стрільба, мисливство, туризм тощо.

9.Країни та регіони – місця, до яких можливо доставити товари, а також місця, з яких клієнти можуть здійснювати покупки.

10.Фільтри та сортування – механізми, що дозволяють клієнтам швидко знаходити необхідні товари за допомогою певних критеріїв, таких як ціна, бренд, тип зброї тощо.

Перш ніж приступити до виконання завдань, ми опрацювали найважливіші аспекти роботи сайту. Тому було створено математичну модель сайту за допомогою інструментів UML, щоб визначити пріоритетність завданб щодо створення сайту.

Концептуальна модель – це абстрактна модель, яка визначає структуру системи, що моделюється, властивості її елементів і специфічні для системи причинно-наслідкові зв’язки, які є істотними для досягнення цілей моделювання. Попередні малюнки використовуються для представлення концептуальної моделі.

Діаграма прецедентів – це діаграма, яка демонструє нам взаємозалежність між користувачем та адміністратором з предметами в системі (рисунок 2.1)

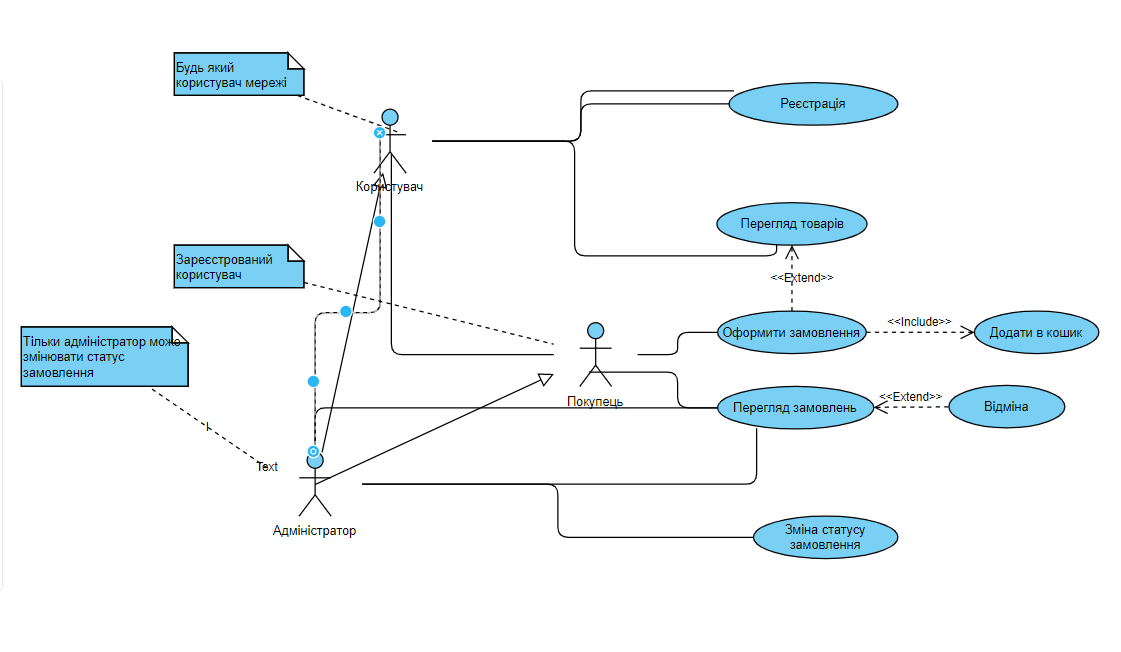
****

Рисунок 2.1 – Діаграма прецедентів

## 2.3 Моделювання даних

Моделювання даних – це представлення структур даних у таблиці для бази даних і є дуже потужним вираженням бізнес-вимог. Ця модель даних є посібником, що використовується функціональними та технічними аналітиками при розробці та впровадженні бази даних.

Моделі даних використовуються для багатьох цілей, від концептуальних моделей високого рівня до фізичних моделей даних.

Для збереження даних використовується база даних на локальному сервері з базою даних SQLite.

Зв’язки між таблицями зображено на рисунку 2.2.

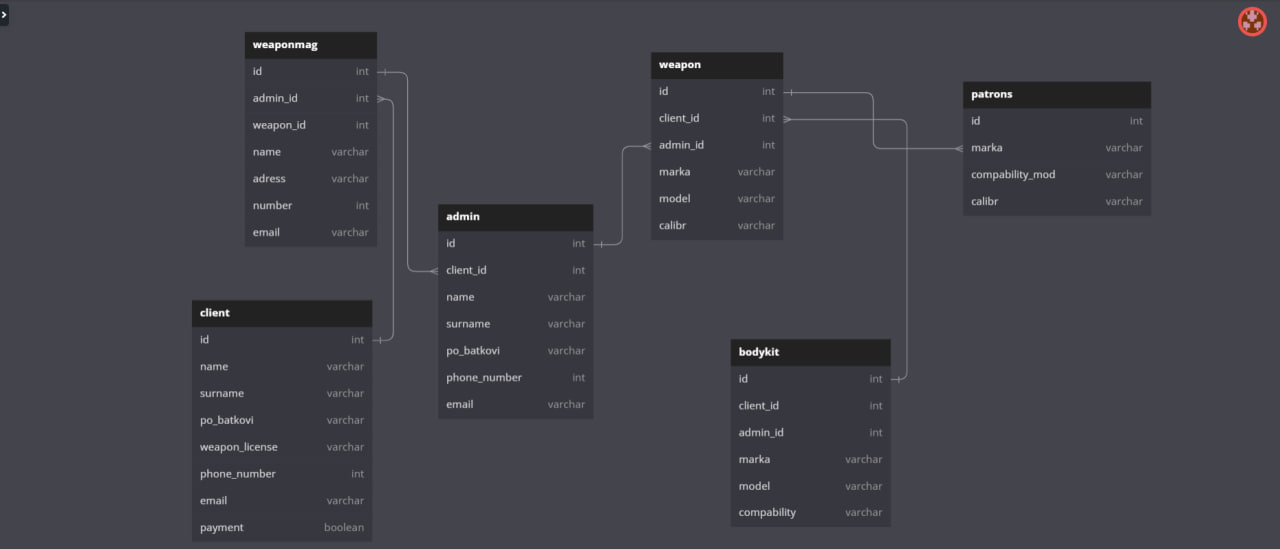
****

Рисунок 2.2 – ER-діаграма

Діаграма класів – це статичний декларативний набір елементів моделі. Дана діаграма дає нам найповніший і детальний огляд зв’язків, функцій та інформації про окремі класи інтернет-магазину. (рисунок 2.3)

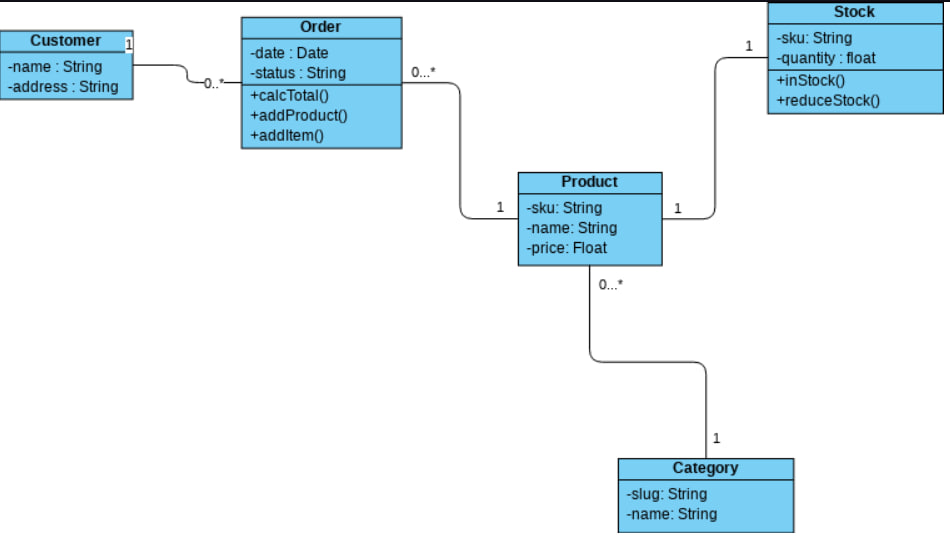
****

Рисунок 2.3 – Діаграма класів

Моделювання даних та програмного забезпечення є запорукою ефективності функціоналу та навігації по сайту майбутньої розробки, що призведе до простоти у використанні.

## 2.4 Проєктування інтерфейсу

Інтерфейс вікна стартової сторінки інтернет-магазину складається з меню. Інтерфейс користувача (UI) – є однією з основних частин у реалізації програмного продукту, оскільки він повинен бути простим у використанні та відповідати вимогам місії.

Інтерфейсні можливості повинні повністю відображати функціональність програми. Час відгуку інтерфейсу інтернет-магазину має бути мінімальним, щоб користувачеві не довелося чекати довше, ніж необхідно, щоб виконати задану дію.

Майже будь-який сайт – це набір сторінок, які містять текст, зображення, мультимедіа чи інші об’єкти. Щоб відвідувачам було комфортно працювати на сайті та легко шукати потрібну інформацію, сайт повинен мати чітку та продуману структуру.

Структура сайту – це внутрішня організація сайту, його «скелет», розміщення сторінок, розділів, підрозділів, додаткових матеріалів. І перше завдання – створити акуратний порядок із хаотичного накопичення інформації.

Інтерфейс головної сторінки сайту складається з меню, логотипу та головної інформації (рисунок 2.4).



Рисунок 2.4 – Головна сторінка

Головна сторінка містить 6 основних сторінок (рисунок 2.5 – 2.10).

На сторінці «Реєстрація» виводиться інформація, яку новому користувачу потрібно заповнити для подальшого користування інтернет-магазином (рисунок 2.5).

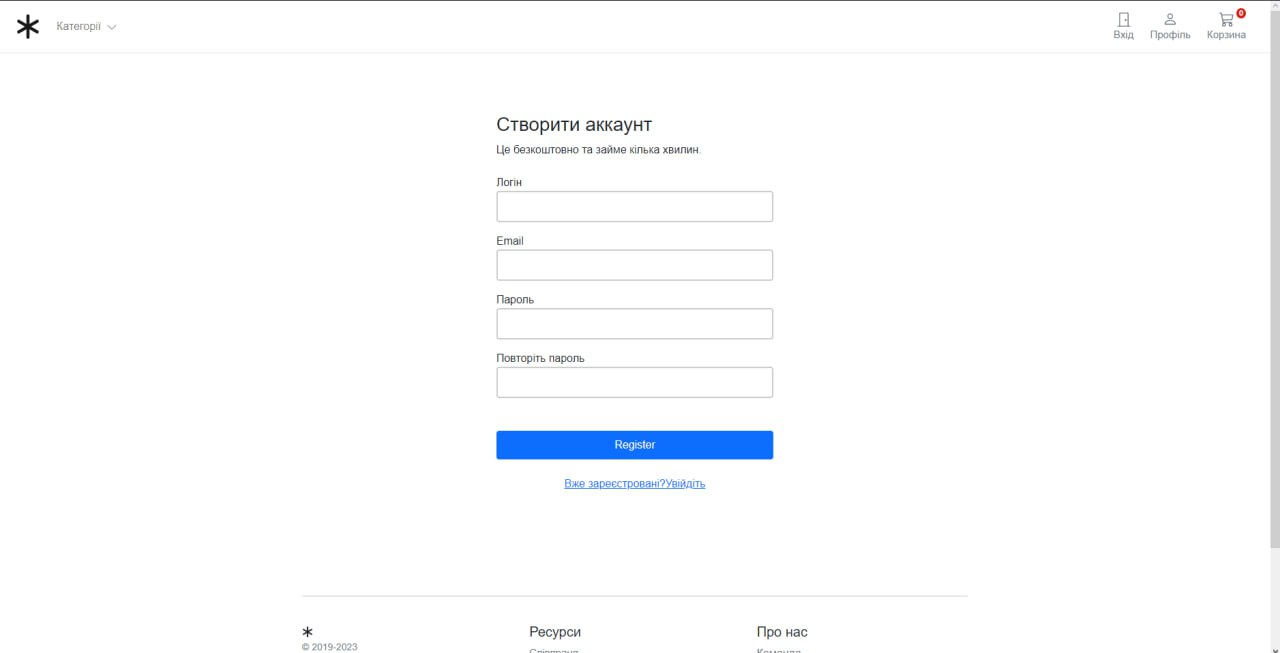


Рисунок 2.5 – Сторінка «Реєстрація»

На сторінці «Авторизація» користувач може авторизуватись в інтернет- магазині (рисунок 2.6).

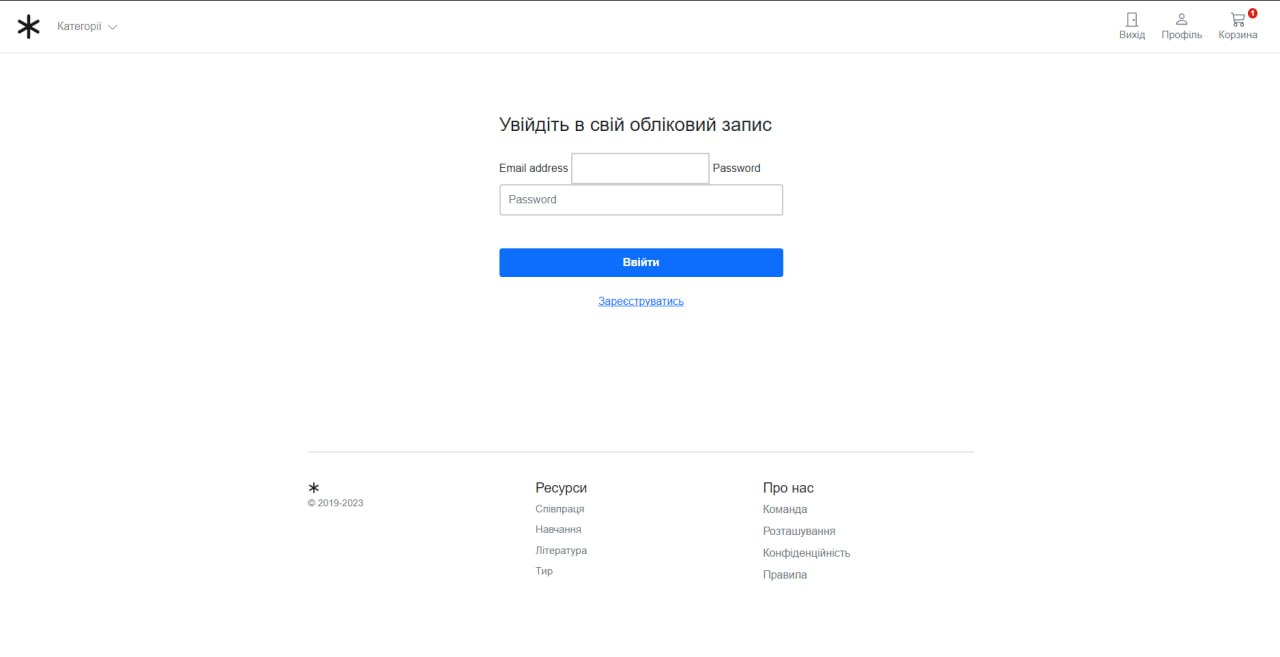


Рисунок 2.6 – Сторінка «Авторизація»

Для висвітлення інформації по певних характеристиках розроблені сторінки по категоріях продукції в інтерне-магазині. На рисунку 2.7 відображається сторінка по категорії «Гвинтівка», де міститься список даної категорії.

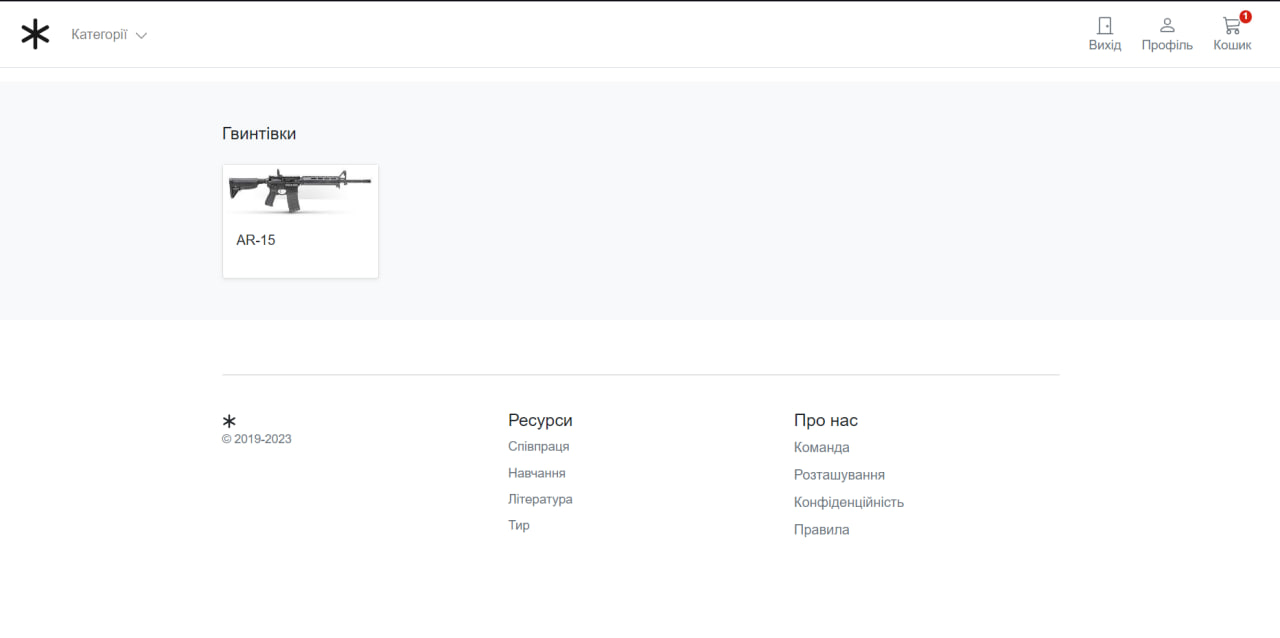


Рисунок 2.7 – Сторінка категорія «Гвинтівки»

Сторінка «Корзина» розроблена для оформлення замовлення. Вона проста у користуванні і інтуїтивно зрозуміла користувачу (рисунок 2.8)

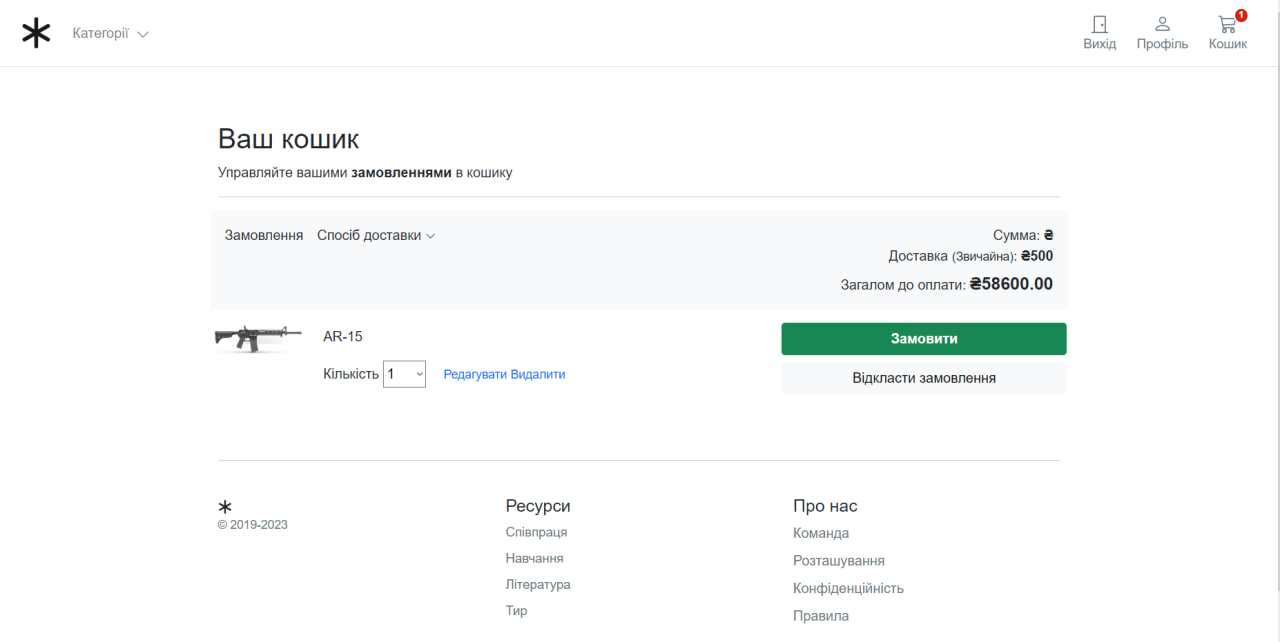


Рисунок 2.8 – Сторінка «Корзина»

Після оформлення замовлення користувачу відкривається сторінка платіжної системи для онлайн-оплати (рисунок 2.9). Після введені необхідної інформації, здійснюється процес оплати.

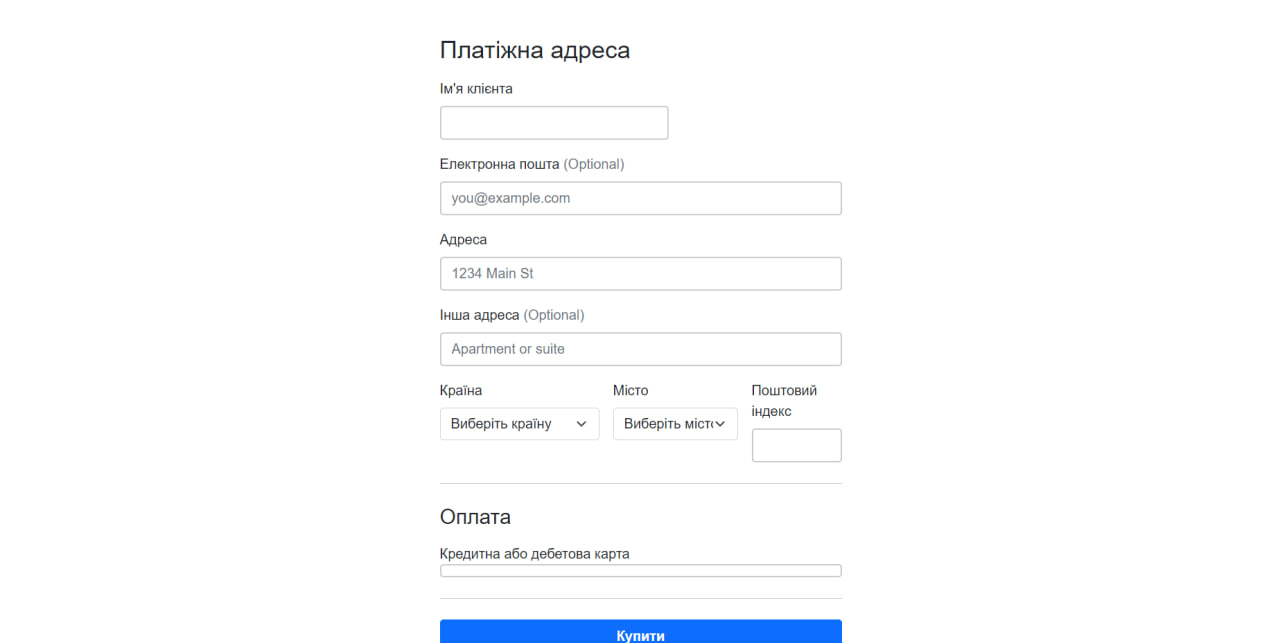
****

Рисунок 2.9 – Сторінка оплати

Дана сторінка розроблена на основі опрацьованих аналогових інтернет- магазинів. Вона зрозуміла та містить підказки для користувача.

Для подальшого користування сайту передбачена адміністративна панель. Вона спроєктована для інформаційного наповнення інтернет-магазину та управлінської діяльності в цілому. Сторінка, яка відповідає за функціонування адміністративної панелі, дуже проста, інтуїтивно зрозуміла та розрахована на звичного користувача, який має бажання керувати своїм інтернет-магазином.

Вхід в адміністраторську панель здійснюється через посилання на вхід в систему в ролі адміністратора. Після успішного входу в систему відкривається сторінка адміністратора (рисунок 2.10).

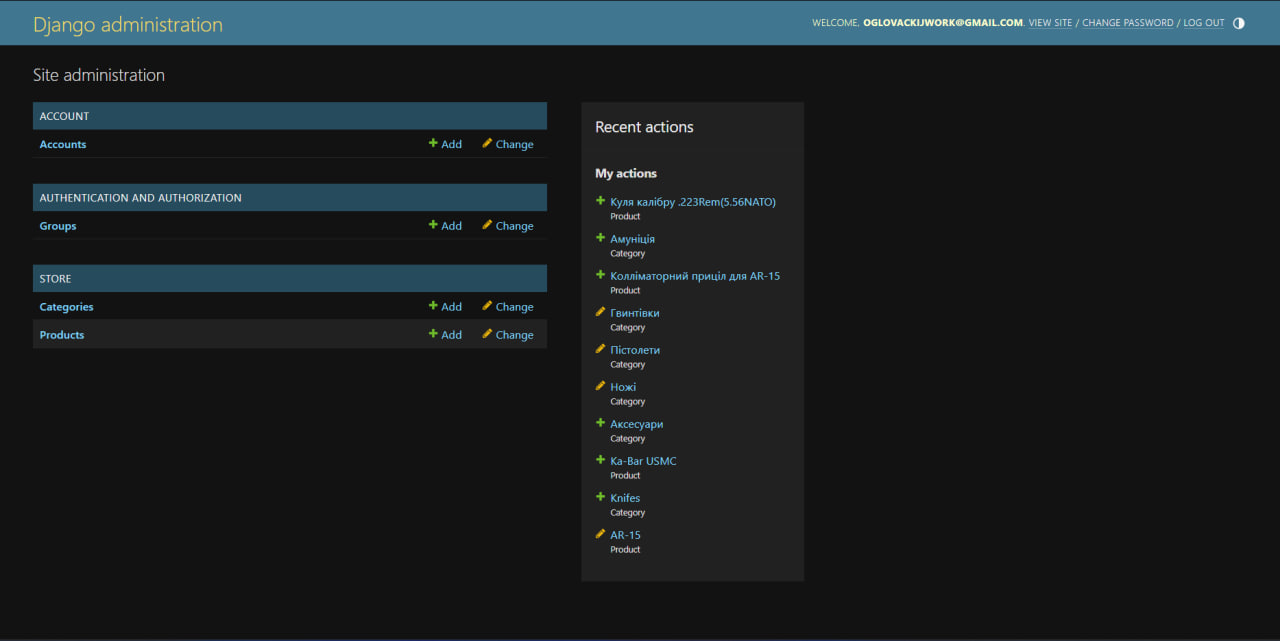


Рисунок 2.10 – Панель адміністратора

Основні функції інтернет-магазину реалізовано в таких файлах:

* account/models.py, forms.py– функціонал користувачами;
* store/admin.py, models.py – функціонал адміністратора.

В даних файлах описано основні функції інтернет-магазину.

Однією з основних функцій є додавання товарів в базу даних (БД). Дана функція реалізована у файлі store/admin.py.

Для авторизації та подальшої реєстрації користувачів розроблено код, який знаходиться у файлі forms.py.

Після вибору товару, він додається у кошик. Даний функціонал реалізовано в файлі store/models.py.

Після вибору товару, він додається у кошик та відбувається процес оплати. Дану функцію реалізовано в файлі payment/views.py.

Повний лістинг програмного коду представлено в додатку А.

# РОЗДІЛ 3. Робочий проЄкт

## 3.1 Засоби розробки

Для розробки даного інтернет-магазину було використано такі технології:

Front-End частина:

1.HTML/CSS(Bootstrap 5) – HTML (HyperText Markup Language) є стандартною мовою розмітки для створення та представлення веб-сторінок. Вона використовується для структурування контенту на веб-сторінках, таких як текст, зображення, посилання, форми та інші елементи. HTML визначає структуру та семантику веб-сторінок за допомогою тегів і атрибутів.

2.JavaScript(JQuery) – jQuery – це широко використовувана бібліотека JavaScript, яка спрощує взаємодію з документом HTML, обробку подій, анімацію, роботу з AJAX і багато іншого. Вона надає потужні і зручні функції та методи для вирішення типових задач, пов'язаних з розробкою веб-сторінок і додатків.

Для розробки серверної частини (Back-End) було використано такі технології:

1.Мова програмування Python в парі з фреймворком Django-Python – це високорівнева інтерпретована мова програмування загального призначення. Вона була створена в 1991 році Гвідо ван Россумом і має простий і зрозумілий синтаксис, що робить її дуже популярною серед початківців у програмуванні, а також серед досвідчених розробників. Django – це високорівневий веб-фреймворк, написаний на мові Python, який дозволяє швидко і ефективно будувати складні веб-додатки. Він пропонує набір готових інструментів, бібліотек і шаблонів, які допомагають розробникам зосередитися на логіці додатку, а не на низькорівневих деталях.

2.База даних SQLite – це компактна вбудовувана реляційна база даних, яка працює як бібліотека, доступна на багатьох платформах, включаючи Windows, macOS, Linux та інші. Вона надає спосіб зберігання та доступу до даних без необхідності встановлення окремого сервера бази даних.

SQLite підтримує стандартний набір SQL-операцій, таких як SELECT, INSERT, UPDATE та DELETE, що дозволяє створювати, змінювати та видаляти дані у базі даних.

## 3.2 Керівництво користувача

При відкритті інтернет-магазину в браузері з’являється головна сторінка (рисунок 3.1). На головній сторінці відображається перелік найпопулярніших товарів за запитами користувачів, меню категорії, корзина, профіль та посилання на вхід та вихід. На даній сторінці користувач може обрати цікавий йому жанр.

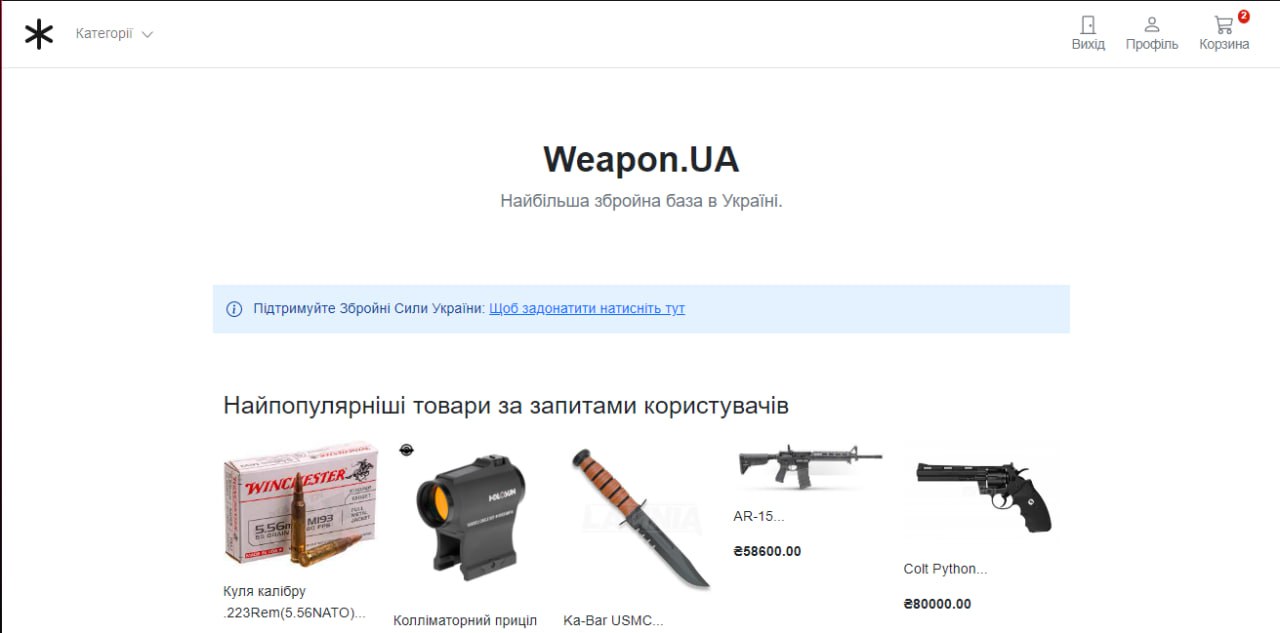


Рисунок 3.1 – Головна сторінка

Вхід в адміністраторську панель здійснюється через посилання на вхід в систему в ролі адміністратора. Після успішного входу в систему відкривається сторінка адміністратора, де міститься перелік музичних композицій (рисунок 3.2).

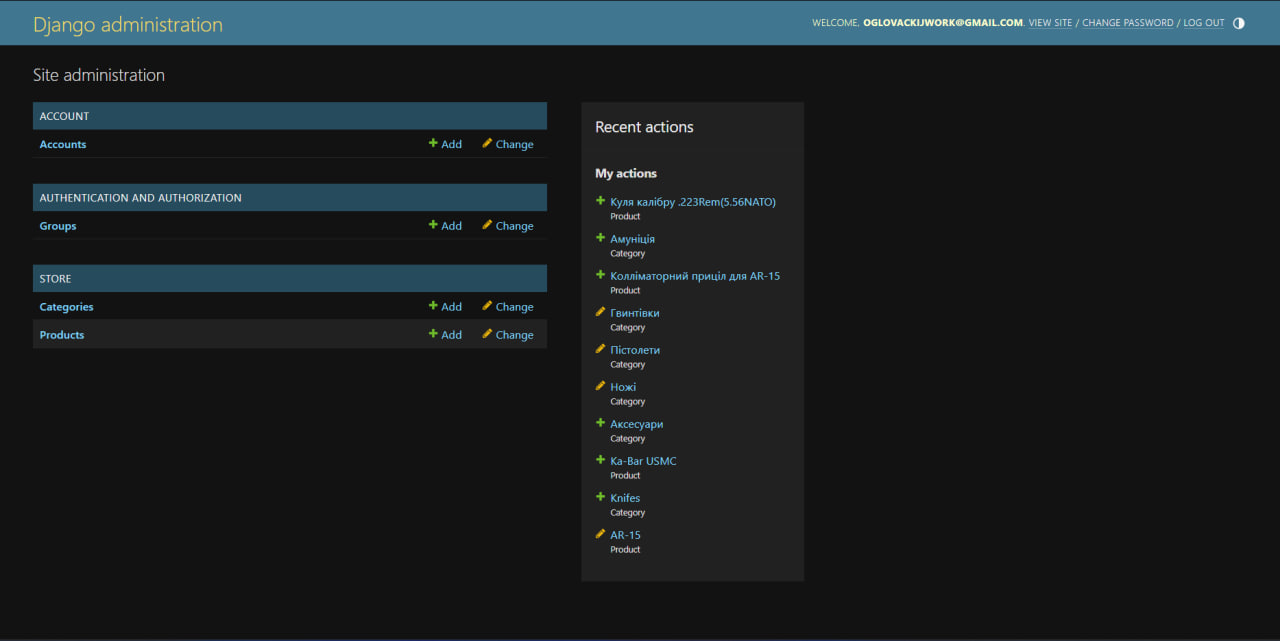


Рисунок 3.2 – Панель адміністратора

В інтернет-магазині користувачі мають можливість авторизуватися для того, щоб здійснювати купівлю. При натисканні на посилання «Профіль» з’являється відповідна форма, де користувач повинен заповнити всього одне поле «Email address». Після чого акаунт користувача створюється автоматично і здійснюється перехід у панель користувача (рисунок 3.3)

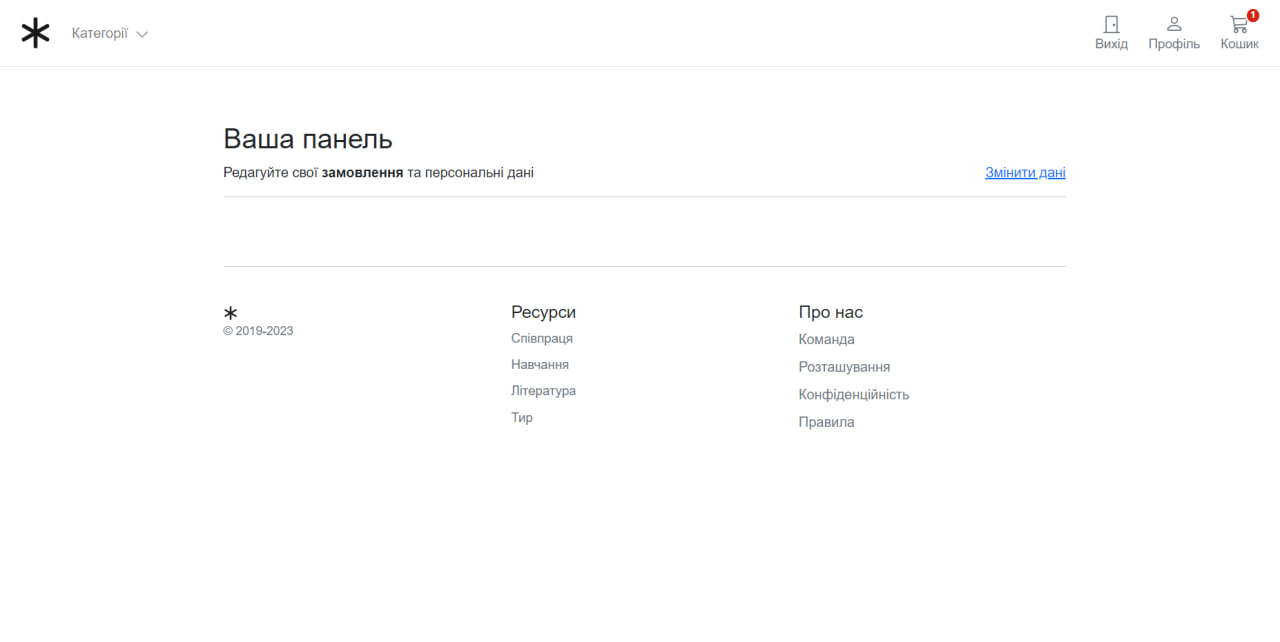


Рисунок 3.2 – Панель користувача

Задля забезпечення хоча б мінімального захисту даних майбутніх клієнтів передбачена реєстрація тобто створення акаунту інтернет-магазину зброї та відслідковування проданої продукції. Для реєстрації передбачено мінімальні дані, які необхідно ввести у форму зображену на рисунку 3.3.

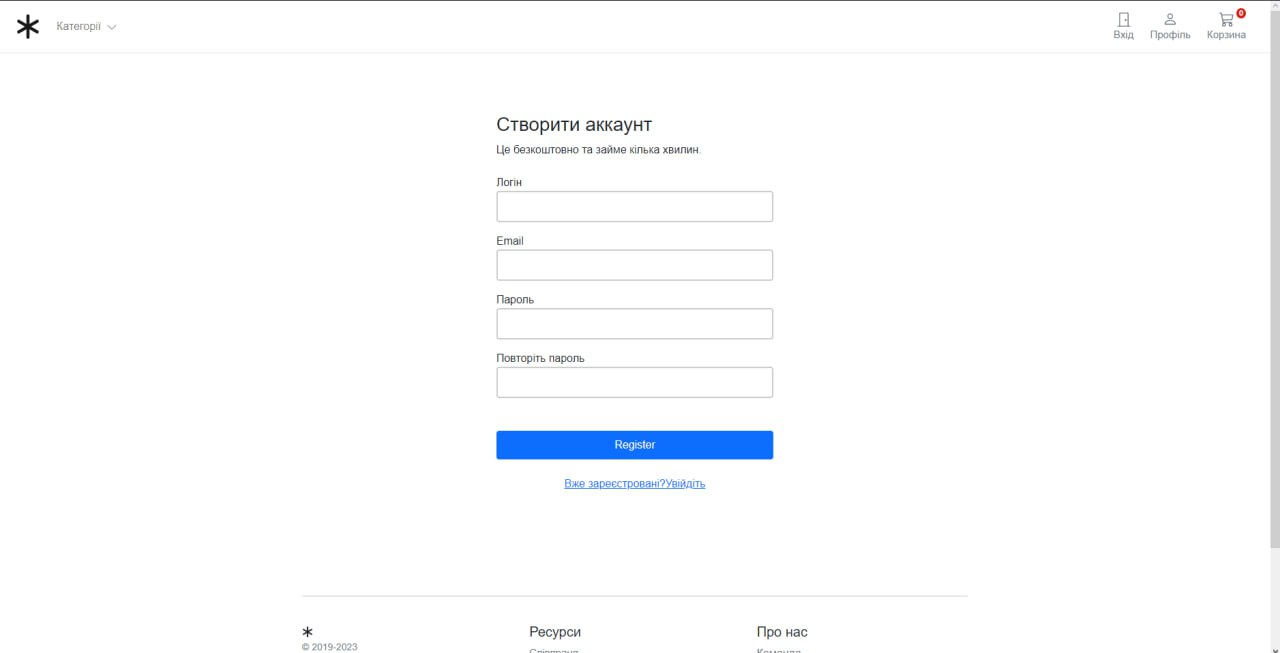


Рисунок 3.3 – Сторінка реєстрації користувачів

Після реєстрації в інтернет-магазині для подальшого користування ним, необхідно буде авторизуватися, а саме ввести логін і пароль (рисунок 3.4).

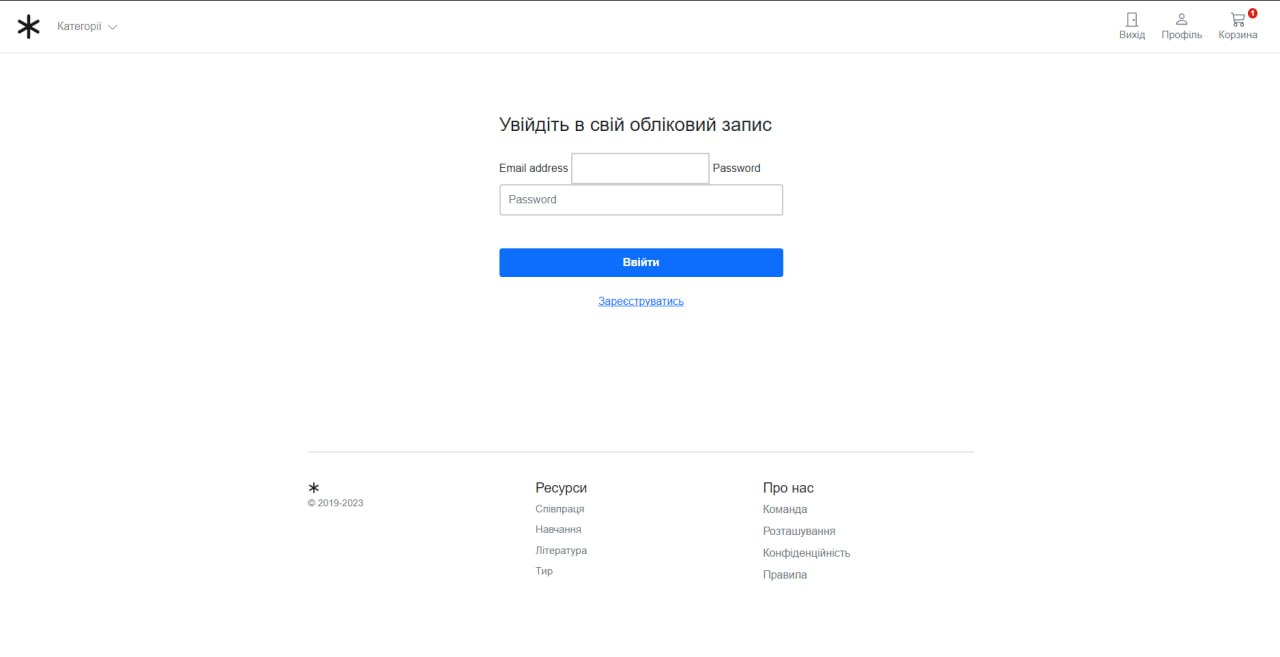


Рисунок 3.4 – Сторінка авторизації

При розробці інтернет-магазину було передбачено форму зміни даних в акаунті користувача або видалення акаунту повністю (рисунок 3.5)

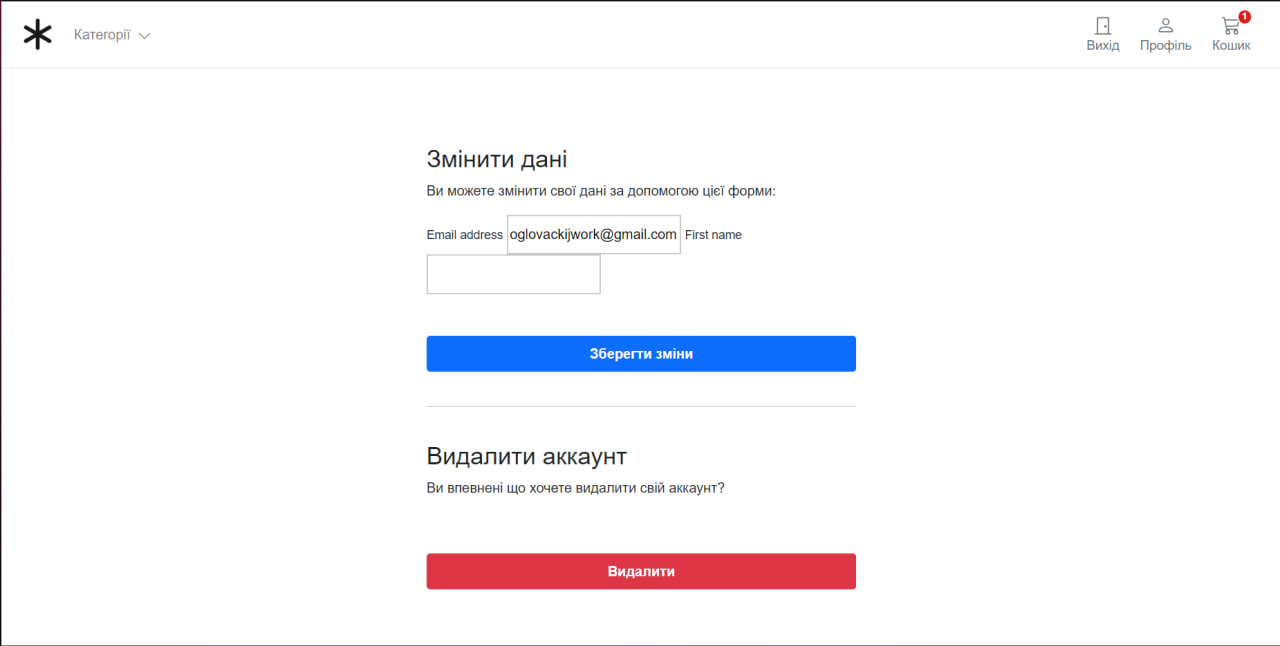


Рисунок 3.5 – Форма зміни інформації та видалення аккаунту

Для полегшення мобільності сайту наявна продукція об’єднана по категоріях. При виборі категорії «Гвинтівка» відкривається сторінка, де міститься список даної категорії (рисунок 3.6)

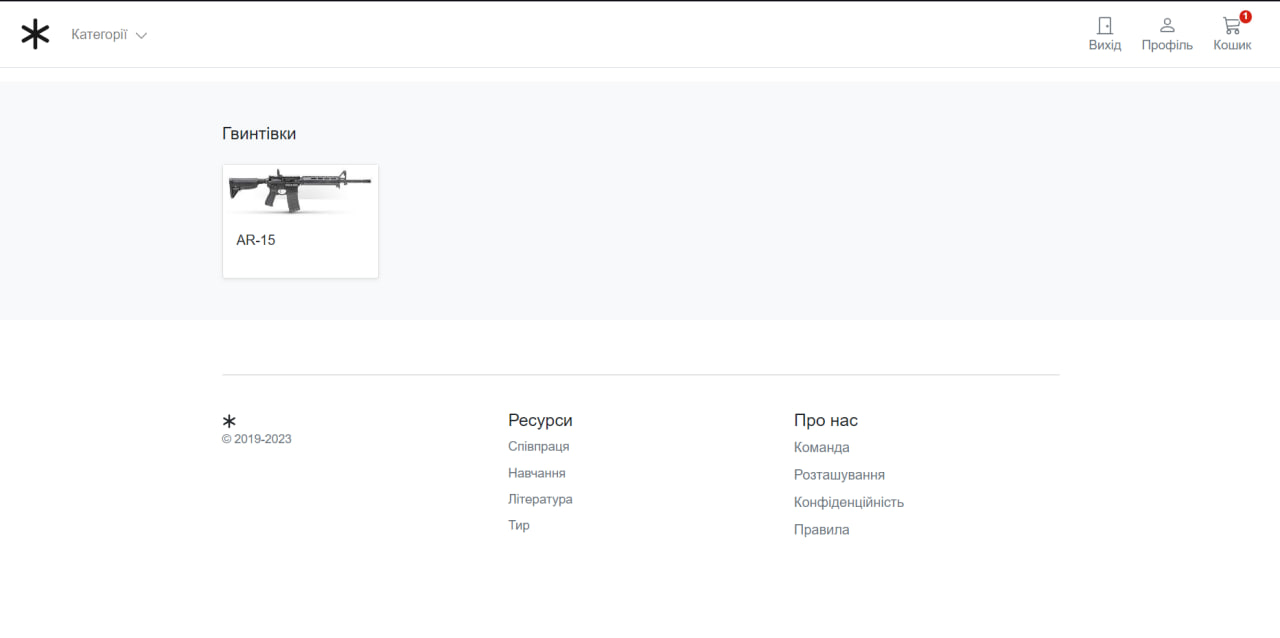


Рисунок 3.6 – Категорія «Гвинтівки»

При виборі категорії «Пістолети» відкривається сторінка, де міститься список даної категорії (рисунок 3.7)

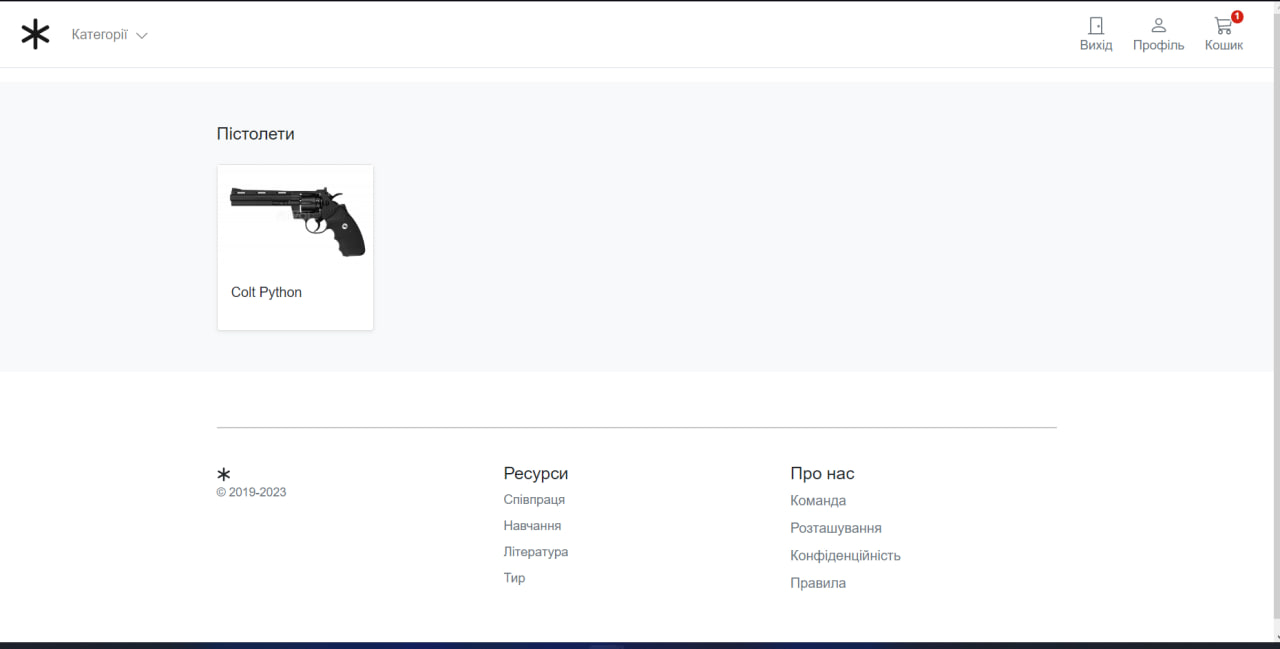


Рисунок 3.7 – Категорія «Пістолети»

При виборі категорії «Ножі» відкривається сторінка, де міститься список даної категорії (рисунок 38)

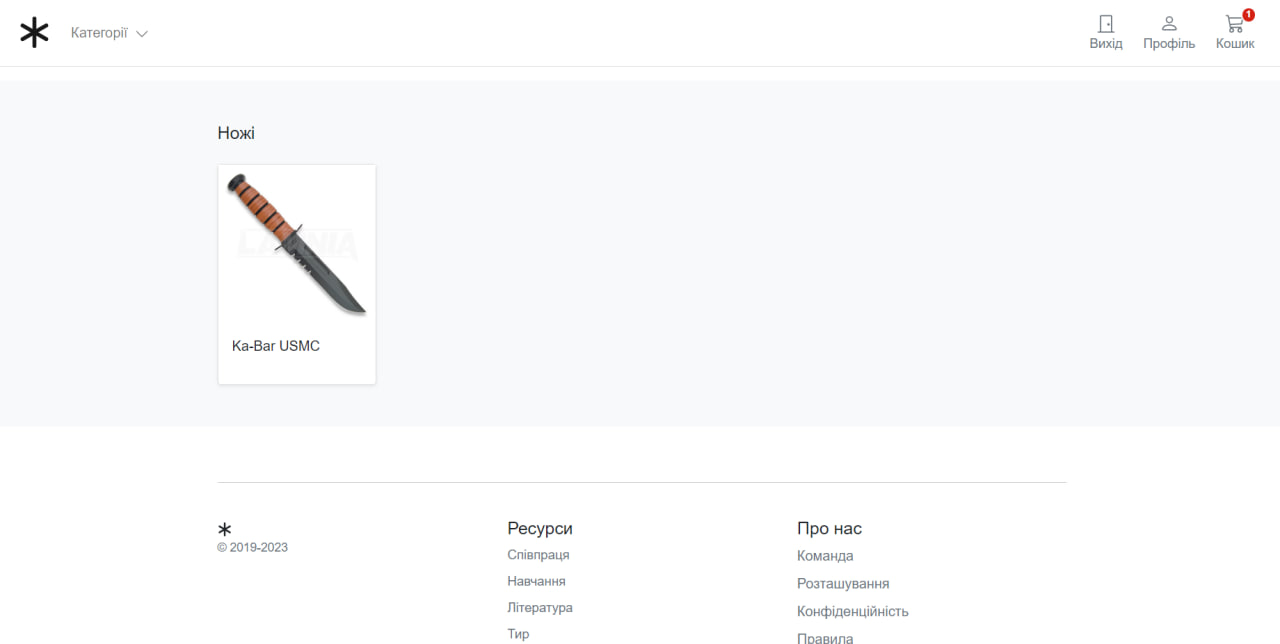


Рисунок 3.8 – Категорія «Ножі»

При виборі категорії «Аксесуари» відкривається сторінка, де міститься список даної категорії (рисунок 3.9)

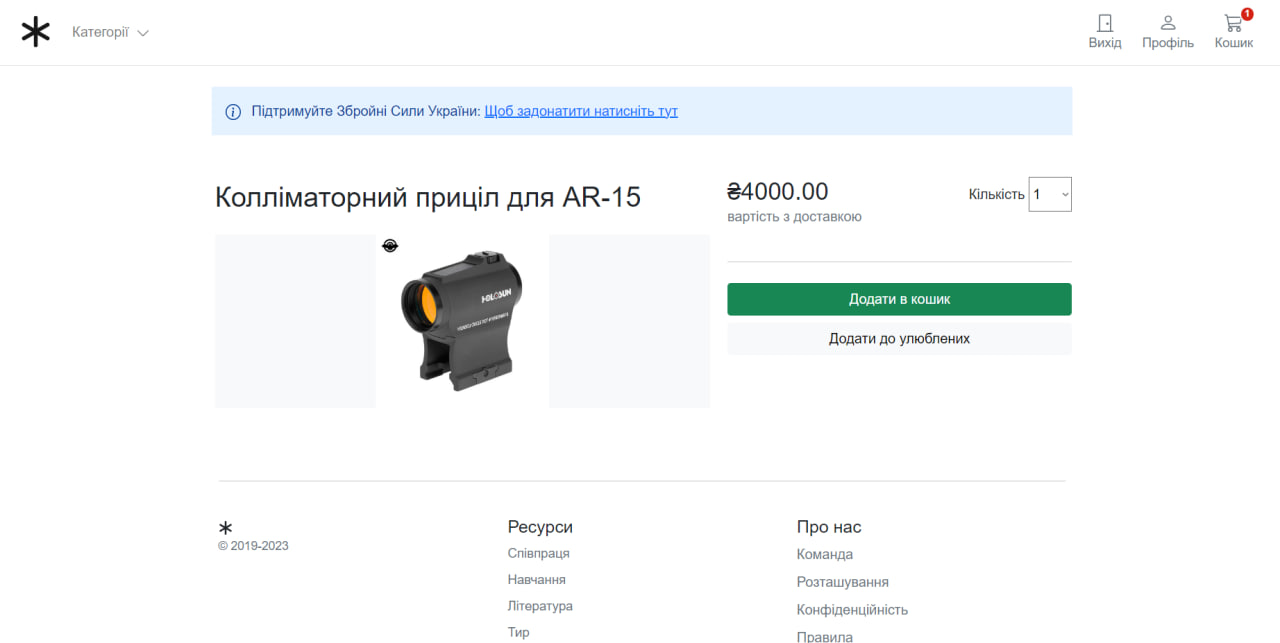


Рисунок 3.9 – Категорія «Аксесуари»

Категорія «Амуніція» має вигляд (рисунок 3.10)

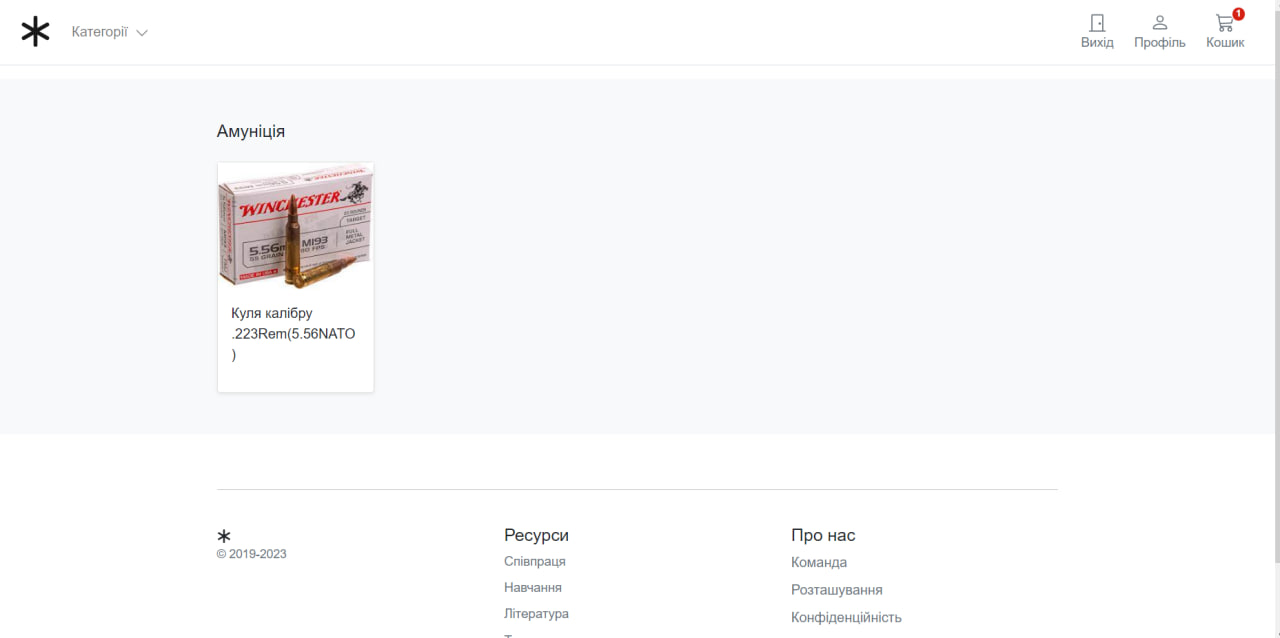


Рисунок 3.10 – Категорія «Амуніція»

При натисканні на обрані з каталогу товари, користувач переходить на нову сторінку, на якій йому доступна повна інформація про даний товар. Також на цій же сторінці, користувач має можливість додати цей товар в кошик (рисунок 3.11)

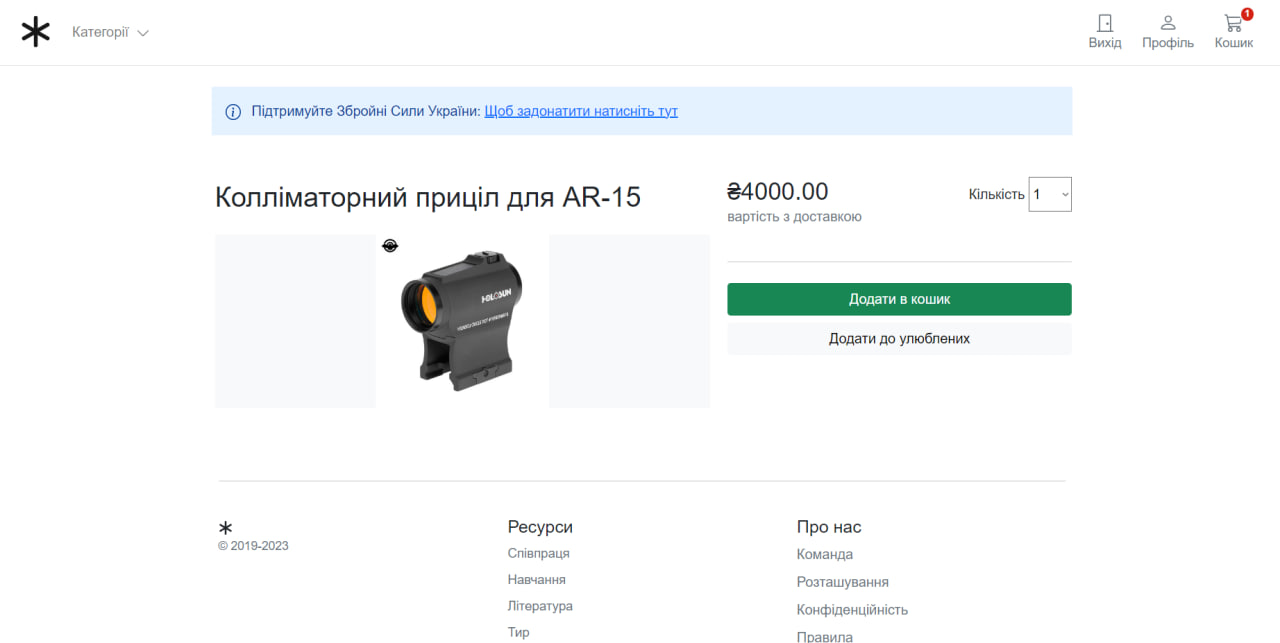


Рисунок 3.11 – Сторінка наявного товару

Після додавання продукції в корзину, вона з’являється в корзині користувача. На сторінці «Корзина» користувач має можливість оформити загальне замовлення, а також додавати та видаляти товари із кошика (рисунок 3.12)

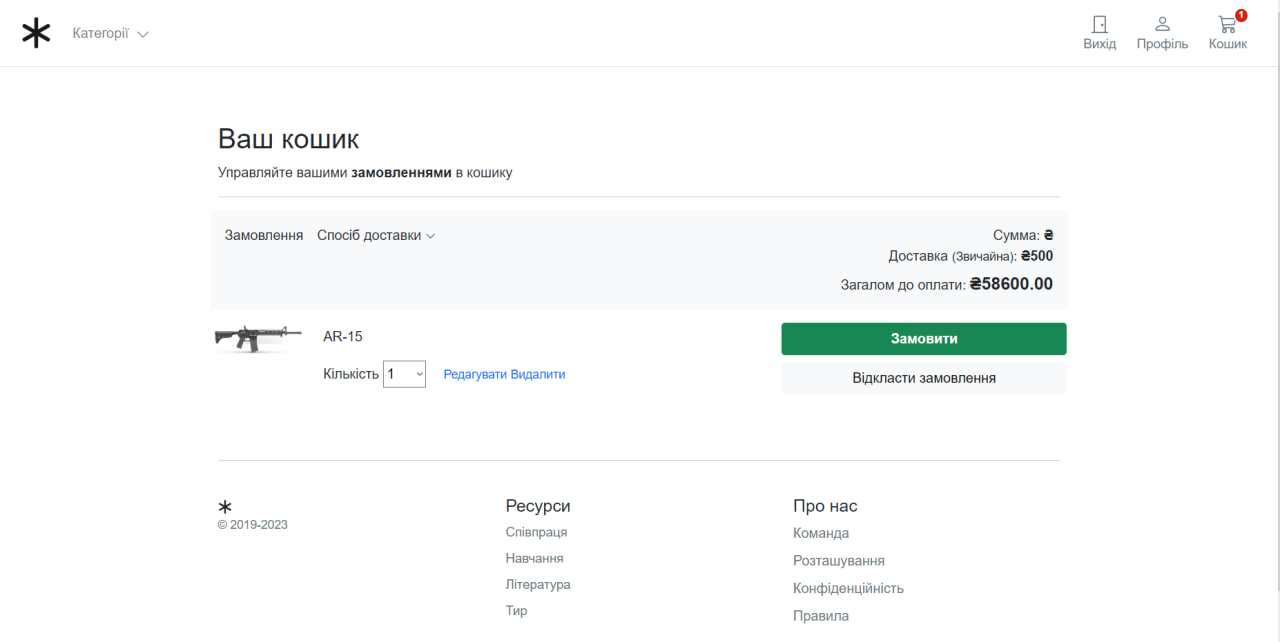


Рисунок 3.12 – Сторінка «Корзина»

Після оформлення замовлення користувачу відкривається сторінка платіжної системи для онлайн-оплати (рисунок 3.13). Після введення необхідної інформації, здійснюється процес оплати.

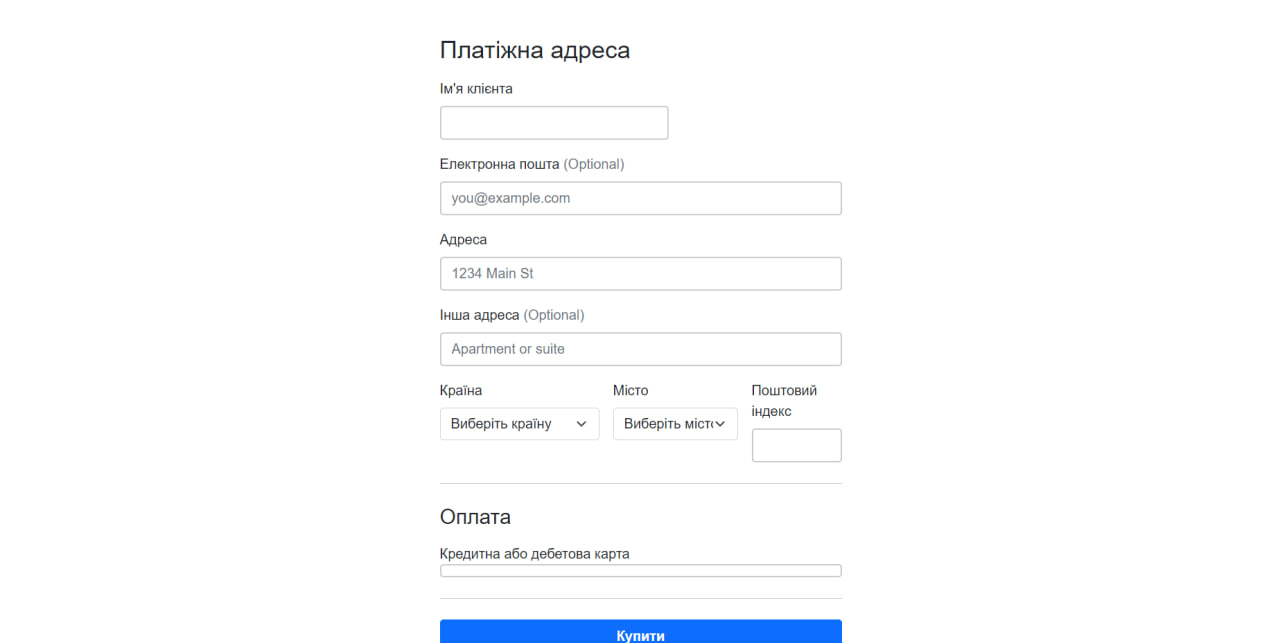
****

Рисунок 3.13 – Сторінка оплати

Для перевірки коректної роботи було проведене тестування інтернет-магазину зброї. Оскільки тестування – це процес технічного дослідження, який виконується на вимогу замовників, і призначений для вияву інформації про якість продукту відносно контексту, в якому він має використовуватись. До цього процесу входить виконання програми з метою знайдення помилок.

Тестування інтернет-магазину зброї виконувалося для перевірки коректності роботи сайту в різних браузерах та для виявлення виняткових ситуацій на сайті.

Інтернет-магазин зброї відображається адекватно в різних браузерах та на даному етапі на виявлено виняткових ситуацій.

# ВИСНОВКИ

Розвиток електронної комерції в сфері зброї є актуальним і перспективним напрямком. Збільшення кількості покупців, які шукають зброю онлайн, свідчить про популярність цього ринку.

В результаті виконання дипломного проєкту було спроектовано та реалізовано інтернет-магазин зброї. Даний сервіс має зручний та інтуїтивно-зрозумілий інтерфейс. Призначенням даного порталу є придбання зброї, яка є у вільному доступі.

Перед створенням інтернет-магазину було проведено аналіз та дослідження даної предметної області в даній галузі, розглянуто аналоги.

При виконанні дипломного проєкту було створено архітектуру, структуру та інтерфейс даного інтернет-магазину. Після тестування веб-сайту можна зробити висновки, що інтернет-магазин зброї працює коректно і відповідає усім вимогам.

Для розробки даного проєкту були використані функції та алгоритми які були описані вище.

Програму було розроблено мовою програмування Python з використанням додаткових специфікацій HTML, CSS та Java Script за допомогою локального сервера, СУБД SQLite та фреймворку для веб-розробки Python-Django.

Під час розробки програмного продукту отримано навички проєктування та розробки програмного забезпечення, вдосконалено навички для створення власних функцій.

Важливими перевагами даної розробки є простота і зручність в користуванні У майбутньому даний програмний продукт можна перепрофілювати в повноцінний воєнторг, тобто здійснювати продаж військового одягу та засобів індивідуального захисту, покращити інтерфейс та додати прямий зв’язок між адміністрацією та потенційними покупцями.. Потенційний попит на створене програмне забезпечення розраховано на специфічну цільову аудиторію.

Розроблений програмний продукт пройшов тестування та відповідає основним вимогам надійності.

# перелік джерел

1. Інтернет-енциклопедія [Електронний ресурс]. – <https://uk.wikipedia.org/wiki/>
2. Технічна документація Django. [Електронний ресурс]. URL: https://docs.djangoproject.com/en/3.2/
3. Бегун А.В., Камінський О.Є. Web-програмування Навч. посіб. – КНЕУ, 2011. – 436с.
4. Шлосснейгл Дж. Професійне програмування на PHP. – ВД «Вільямс», 2006. – 367с.
5. Форум з зразками sql та php функцій [Електронний ресурс]. –http://www.cyberforum.uа

# Додатки

**Додаток А**

**Лістинг коду**

A1. Models.py

from django.conf import settings

from django.db import models

from django.urls import reverse

class ProductManager(models.Manager):

def get\_queryset(self):

return super(ProductManager, self).get\_queryset().filter(is\_active=True)

class Category(models.Model):

name = models.CharField(max\_length=255, db\_index=True)

slug = models.SlugField(max\_length=255, unique=True)

class Meta:

verbose\_name\_plural = 'categories'

def get\_absolute\_url(self):

return reverse('store:category\_list', args=[self.slug])

def str(self):

return self.name

class Product(models.Model):

category = models.ForeignKey(Category, related\_name='product', on\_delete=models.CASCADE)

created\_by = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE, related\_name='product\_creator')

title = models.CharField(max\_length=255)

author = models.CharField(max\_length=255, default='admin')

description = models.TextField(blank=True)

image = models.ImageField(upload\_to='images/', default='images/default.png')

slug = models.SlugField(max\_length=255)

price = models.DecimalField(max\_digits=150, decimal\_places=2)

in\_stock = models.BooleanField(default=True)

is\_active = models.BooleanField(default=True)

created = models.DateTimeField(auto\_now\_add=True)

updated = models.DateTimeField(auto\_now=True)

objects = models.Manager()

products = ProductManager()

class Meta:

verbose\_name\_plural = 'Products'

ordering = ('-created',)

def get\_absolute\_url(self):

return reverse('store:product\_detail', args=[self.slug])

def str(self):

return self.title

A2. admin.py

from django.contrib import admin

from .models import Category, Product

@admin.register(Category)

class CategoryAdmin(admin.ModelAdmin):

list\_display = ['name', 'slug']

prepopulated\_fields = {'slug': ('name',)}

@admin.register(Product)

class ProductAdmin(admin.ModelAdmin):

list\_display = ['title', 'author', 'slug', 'price',

'in\_stock', 'created', 'updated']

list\_filter = ['in\_stock', 'is\_active']

list\_editable = ['price', 'in\_stock']

prepopulated\_fields = {'slug': ('title',)}

views.py

from django.shortcuts import get\_object\_or\_404, render

from .models import Category, Product

def product\_all(request):

products = Product.products.all()

return render(request, 'store/home.html', {'products': products})

def category\_list(request, category\_slug=None):

category = get\_object\_or\_404(Category, slug=category\_slug)

products = Product.objects.filter(category=category)

return render(request, 'store/products/category.html', {'category': category, 'products': products})

def product\_detail(request, slug):

product = get\_object\_or\_404(Product, slug=slug, in\_stock=True)

return render(request, 'store/products/single.html', {'product': product})

basket.py

from decimal import Decimal

from store.models import Product

class Basket():

"""

A base Basket class, providing some default behaviors that

can be inherited or overrided, as necessary.

"""

def init(self, request):

self.session = request.session

basket = self.session.get('skey')

if 'skey' not in request.session:

basket = self.session['skey'] = {}

self.basket = basket

def add(self, product, qty):

"""

Adding and updating the users basket session data

"""

product\_id = str(product.id)

if product\_id in self.basket:

self.basket[product\_id]['qty'] = qty

else:

self.basket[product\_id] = {'price': str(product.price), 'qty': qty}

self.save()

def iter(self):

"""

Collect the product\_id in the session data to query the database

and return products

"""

product\_ids = self.basket.keys()

products = Product.products.filter(id\_\_in=product\_ids)

basket = self.basket.copy()

for product in products:

basket[str(product.id)]['product'] = product

for item in basket.values():

item['price'] = Decimal(item['price'])

item['total\_price'] = item['price'] \* item['qty']

yield item

def len(self):

"""

Get the basket data and count the qty of items

"""

return sum(item['qty'] for item in self.basket.values())

def update(self, product, qty):

"""

Update values in session data

"""

product\_id = str(product)

if product\_id in self.basket:

self.basket[product\_id]['qty'] = qty

self.save()

def get\_total\_price(self):

return sum(Decimal(item['price']) \* item['qty'] for item in self.basket.values())

def delete(self, product):

"""

Delete item from session data

"""

product\_id = str(product)

if product\_id in self.basket:

del self.basket[product\_id]

print(product\_id)

self.save()

def save(self):

self.session.modified = True

account/models.py

from django.contrib.auth.models import (AbstractBaseUser, BaseUserManager,

PermissionsMixin)

from django.core.mail import send\_mail

from django.db import models

from django.utils.translation import gettext\_lazy as \_

from django\_countries.fields import CountryField

class CustomAccountManager(BaseUserManager):

def create\_superuser(self, email, user\_name, password, \*\*other\_fields):

other\_fields.setdefault('is\_staff', True)

other\_fields.setdefault('is\_superuser', True)

other\_fields.setdefault('is\_active', True)

if other\_fields.get('is\_staff') is not True:

raise ValueError(

'Superuser must be assigned to is\_staff=True.')

if other\_fields.get('is\_superuser') is not True:

raise ValueError(

'Superuser must be assigned to is\_superuser=True.')

return self.create\_user(email, user\_name, password, \*\*other\_fields)

def create\_user(self, email, user\_name, password, \*\*other\_fields):

if not email:

raise ValueError(\_('You must provide an email address'))

email = self.normalize\_email(email)

user = self.model(email=email, user\_name=user\_name,

\*\*other\_fields)

user.set\_password(password)

user.save()

return user

class UserBase(AbstractBaseUser, PermissionsMixin):

email = models.EmailField(\_('email address'), unique=True)

user\_name = models.CharField(max\_length=150, unique=True)

first\_name = models.CharField(max\_length=150, blank=True)

about = models.TextField(\_(

'about'), max\_length=500, blank=True)

# Delivery details

country = CountryField()

phone\_number = models.CharField(max\_length=15, blank=True)

postcode = models.CharField(max\_length=12, blank=True)

address\_line\_1 = models.CharField(max\_length=150, blank=True)

address\_line\_2 = models.CharField(max\_length=150, blank=True)

town\_city = models.CharField(max\_length=150, blank=True)

# User Status

is\_active = models.BooleanField(default=False)

is\_staff = models.BooleanField(default=False)

created = models.DateTimeField(auto\_now\_add=True)

updated = models.DateTimeField(auto\_now=True)

objects = CustomAccountManager()

USERNAME\_FIELD = 'email'

REQUIRED\_FIELDS = ['user\_name']

class Meta:

verbose\_name = "Accounts"

verbose\_name\_plural = "Accounts"

def email\_user(self, subject, message):

send\_mail(

subject,

message,

'l@1.com',

[self.email],

fail\_silently=False,

)

def str(self):

return self.user\_name

A3. account/forms.py

from django import forms

from .models import UserBase

from django.contrib.auth.forms import(AuthenticationForm,UserChangeForm,PasswordResetForm,PasswordChangeForm,SetPasswordForm)

from django.contrib.auth.views import LoginView

class UserLoginForm(AuthenticationForm):

username=forms.CharField(widget=forms.TextInput)

attrs={'class':'form-control mb-3','placeholder':'Username','id':'login-username'}

password=forms.CharField(widget=forms.PasswordInput(

attrs={

'class':'form-control',

'placeholder':'Password',

'id':'login-pwd',

}

))

class UserChangeForm(forms.ModelForm):

email = forms.EmailField(

label='Account email (can not be changed)', max\_length=200, widget=forms.TextInput(

attrs={'class': 'form-control mb-3', 'placeholder': 'email', 'id': 'form-email', 'readonly': 'readonly'}))

user\_name = forms.CharField(

label='Firstname', min\_length=4, max\_length=50, widget=forms.TextInput(

attrs={'class': 'form-control mb-3', 'placeholder': 'Username', 'id': 'form-firstname', 'readonly': 'readonly'}))

first\_name = forms.CharField(

label='Username', min\_length=4, max\_length=50, widget=forms.TextInput(

attrs={'class': 'form-control mb-3', 'placeholder': 'Firstname', 'id': 'form-lastname'}))

class Meta:

model = UserBase

fields = ('email', 'user\_name', 'first\_name',)

def init(self, \*args, \*\*kwargs):

super().init(\*args, \*\*kwargs)

self.fields['user\_name'].required = True

#self.fields['email'].required = True

class RegistrationForm(forms.ModelForm):

user\_name=forms.CharField(label='Логін', min\_length=4,max\_length=50,help\_text='Required')

email=forms.EmailField(max\_length=100,help\_text='Required',error\_messages={'required':'Sorry,you will need an email'})

password=forms.CharField(label='Пароль',widget=forms.PasswordInput)

password2=forms.CharField(label='Повторіть пароль',widget=forms.PasswordInput)

class Meta:

model=UserBase

fields=('user\_name','email',)

def clean\_username(self):

user\_name=self.cleaned\_data['user\_name'].lower()

r=UserBase.objects.filter(user\_name=user\_name)

if r.count():

raise forms.ValidationError("Користувач з таким іменем вже існує")

return user\_name

def clean\_password2(self):

cd=self.cleaned\_data

if cd['password']!=cd['password2']:

raise forms.ValidationError('Паролі не збігаються')

return cd['password2']

def clean\_email(self):

email=self.cleaned\_data['email']

if UserBase.objects.filter(email=email).exists():

raise forms.ValidationError('Цей email вже використовується')

return email

def init(self, \*args, \*\*kwargs):

super().init(\*args, \*\*kwargs)

self.fields['user\_name'].widget.attrs.update(

{'class': 'form-control mb-3', 'placeholder': ''})

self.fields['email'].widget.attrs.update(

{'class': 'form-control mb-3', 'placeholder': '', 'name': 'email', 'id': 'id\_email'})

self.fields['password'].widget.attrs.update(

{'class': 'form-control mb-3', 'placeholder': ''})

self.fields['password2'].widget.attrs.update(

{'class': 'form-control', 'placeholder': ''})

class PwdResetForm(PasswordResetForm):

email = forms.EmailField(max\_length=254, widget=forms.TextInput(

attrs={'class': 'form-control mb-3', 'placeholder': 'Email', 'id': 'form-email'}))

def clean\_email(self):

email = self.cleaned\_data['email']

u = UserBase.objects.filter(email=email)

if not u:

raise forms.ValidationError(

'Нажаль ми не можемо знайти дану електронну пошту')

return email

class PwdResetConfirmForm(SetPasswordForm):

new\_password1 = forms.CharField(

label='New password', widget=forms.PasswordInput(

attrs={'class': 'form-control mb-3', 'placeholder': 'New Password', 'id': 'form-newpass'}))

new\_password2 = forms.CharField(

label='Repeat password', widget=forms.PasswordInput(

attrs={'class': 'form-control mb-3', 'placeholder': 'New Password', 'id': 'form-new-pass2'}))

A4. payment/views.py

import stripe

from django.shortcuts import render

from django.contrib.auth.decorators import login\_required

from basket.basket import Basket

@login\_required

def BasketView(request):

basket=Basket(request)

total=str(basket.get\_total\_price())

total=total.replace('.','')

total=int(total)

stripe.api\_key = 'sk\_test\_51NC3oGCtDtbyYHLCDeXUvercuytY1cY1nPIaqnO00wfg3hups91VAl0nqxi3fRMIn2LtVReHzZNO8rEra9UHcHNp00YOefs5ML'

intent = stripe.PaymentIntent.create(

amount=total,

currency='gbp',

metadata={'userid': request.user.id}

)

return render(request,'payment/home.html',{'client\_secret':intent.client\_secret})

settings.py

import os

from pathlib import Path

BASE\_DIR = Path(file).resolve().parent.parent

SECRET\_KEY = '3xk\*)i0x#k$btl=(6q)te!19=mp6d)lm1+zl#ts4ewxi3-!vm\_'

DEBUG = True

ALLOWED\_HOSTS = ['yourdomain.com', '127.0.0.1']

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'store',

'basket',

'account',

'payment',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'core.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

'store.context\_processors.categories',

'basket.context\_processors.basket',

],

},

},

]

WSGI\_APPLICATION = 'core.wsgi.application'

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

STATIC\_URL = '/static/'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, "static")

]

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media/')

AUTH\_USER\_MODEL='account.UserBase'

LOGIN\_REDIRECT\_URL='/account/dashboard'

LOGIN\_URL='/account/login'

#Email

EMAIL\_BACKEND='django.core.mail.backends.console.EmailBackend'

**ЗАЯВА-ЗАСВІДЧЕННЯ АВТОРА ДИПЛОМНОГО ПРОЄКТУ**

Я, Гловацький Олексій Олександрович, студент IV курсу спеціальності 121 Інженерія програмного забезпечення групи 541, заявляю: моя письмова робота на тему «Iнтернет-магазин зброї», представлена екзаменаційній комісії для публічного захисту, виконана самостійно i без порушення норм законодавства України про авторське право. Матеріал інших авторів, отриманий з друкованих та електронних видань, а також з  кваліфікаційних робіт здобувачів освіти, оформлені належним чином з посиланням на джерело.

Я ознайомлений з чинними «Положенням про дотримання академічної доброчесності педагогічними працівниками та здобувачами освіти Чернівецького політехнічного фахового коледжу», згідно з якими виявлення плагіату та/або критично низький рівень унікальності роботи є підставою для відмови в допуску її до захисту або відмови у присудженні відповідного ступеня фахової передвищої освіти автору такої роботи.

Я даю дозвіл на розміщення електронної копії моєї роботи у фонді дипломних проєктів Чернівецького політехнічного фахового коледжу.

Я даю дозвіл Чернівецькому політехнічному фаховому коледжу на розміщення електронної копії моєї роботи в Базі даних Системи виявлення збігів/ідентичності/схожості академічних текстів.

Робота для перевірки на наявність текстових збігів надається в електронному вигляді. Електронна версія моєї роботи відповідає (ідентична) друкованій.

19 червня 2023 р. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Олексій ГЛОВАЦЬКИЙ

**БІБЛІОГРАФІЧНА ДОВІДКА**

Тема дипломного проєкту: Інтернет-магазин зброї

Обсяг пояснювальної записки: 49 аркуші

Дата закінчення проєкту 19 червня 2023 р.

Підпис студента-дипломника \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_