

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнес-технологій
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему «АВТОМАТИЗАЦІЯ ОБРОБКИ ОНЛАЙН ЗАМОВЛЕНЬ
ІНТЕРНЕТ-МАГАЗИНУ»

Виконала студентка 4 курсу, групи ЕК-81a
(номер курсу) (шифр групи)

Спеціальності 051 «Економіка» (Економічна
кібернетика)

Чорна К.О.

(прізвище, ініціали студента)

Керівник доцент, канд. техн. наук Яценко В.В.

(посада, науковий ступінь, прізвище, ініціали)

Суми – 2022 рік

РЕФЕРАТ
кваліфікаційної роботи бакалавра на тему
«АВТОМАТИЗАЦІЯ ОБРОБКИ ОНЛАЙН ЗАМОВЛЕНЬ
ІНТЕРНЕТ-МАГАЗИНУ»

студентки Чорної Катерини Олександрівни
(прізвище, ім'я, по батькові)

Актуальність теми, обраної для дослідження, визначена тим, що автоматизація роботи підприємства за допомогою вебтехнологій призводить до збільшення доходу.

Мета кваліфікаційної роботи полягає у проектуванні та розробці веборієнтованої інформаційної системи для інтернет-магазину.

Об'єктом дослідження є підприємство «Vape Shop», яке займається роздрібною торгівлею електронних випаровувачів.

Предметом дослідження є веборієнтована система для електронної комерції.

Задачі:

- характеристика діяльності підприємства «Vape Shop»;
- аналіз стану автоматизації бізнес-процесів;
- формування вимоги до вебзастосунку;
- моделювання бізнес-процесів;
- створення архітектури інформаційної системи та визначити технології вирішення поставлених задач;
- проектування веборієнтованої інформаційної системи;
- розробка прототипу вебзастосунку та описати інструкції по використанню системи.

Для досягнення мети роботи використані методи дослідження: аналіз, індукція, дедукція, моделювання, аналогія.

Ключові слова: АВТОМАТИЗАЦІЯ, ІНТЕРНЕТ-МАГАЗИН, PYTHON, DJANGO, PUSCHARM.

Зміст кваліфікаційної роботи викладено на 1 сторінку. Список використаних джерел із 41 найменувань, розміщений на 3 сторінках. Робота

містить 24 рисунків, 1 графік, та 3 таблиці, а також 3 додатки, розміщених на 23 сторінках.

Рік виконання кваліфікаційної роботи – 2022 рік.

Рік захисту роботи – 2022 рік.

Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнес-технологій
Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.е.н., професор
_____ О.В. Кузьменко
“ ” _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
спеціальність 051 «Економіка (Економічна кібернетика)

студентки 4 курсу групи ЕК-81а
(Номер курсу) (Шифр групи)

Чорної Катерини Олександрівни

(Прізвище, ім'я, по батькові)

1. Тема роботи Автоматизація обробки онлайн замовлень інтернет-магазину

затверджена наказом по університету від «21» лютого 2022 року №

2. Термін подання студентом закінченої роботи «9» червня 2022 року

3. Мета кваліфікаційної роботи проектування та розробка веборієнтованої інформаційної системи для інтернет-магазину.

4. Об'єкт дослідження підприємство «Vape Shop», яке займається роздрібною торгівлею електронних випаровувачів.

5. Предмет дослідження веборієнтована система для електронної комерції

6. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1. Дослідження стану автоматизації бізнес-процесів –30 травня 2022 року

(назва – термін подання)

У розділі 1. Проаналізувати та надати характеристику об'єкта дослідження, проаналізувати стан автоматизації бізнес-процесів, сформулювати вимоги до вебзастосунку, змодельювати бізнес-процеси, створення архітектури інформаційної системи та визначити технології вирішення поставлених задач.

(зміст конкретних завдань до розділу, які має виконати студент)

Розділ 2. Реалізація прототипу інтернет магазину — 3 червня 2020 року

(назва – термін подання)

У розділі 2. Спроектувати веборієнтовану інформаційну систему, розробити прототип вебзастосунку, описати інструкції по використанню системи.

(зміст конкретних завдань до розділу, які повинен виконати студент)

7. Консультації з роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			

8. Дата видачі завдання: «4» квітня 2022 року

Керівник кваліфікаційної роботи

(підпис)

В.В. Яценко

(ініціали, прізвище)

Завдання до виконання одержала

(підпис)

К.О. Чорна

(ініціали, прізвище)

ЗМІСТ

ВСТУП.....	7
1 ДОСЛІДЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ.....	9
1.1.Характеристика об'єкта дослідження, та аналіз стану автоматизації.....	9
бізнес-процесів.....	9
1.2. Модель бізнес-процесів	11
1.3 Формування вимог до веборієнтованої системи	13
2 РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНТЕРНЕТ МАГАЗИНУ	18
2.1.Проектування веборієнтованої інформаційної системи	18
2.2. Розробка прототипу вебзастосунку.....	25
2.3. Інструкції по використанню системи.....	29
ВИСНОВКИ	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	40
ДОДАТОК А	43
ДОДАТОК Б	44
ДОДАТОК В.....	50

ВСТУП

Останні роки в нашому житті, дуже багато змін, ІТ – технологія все більше та більше розширюються. Майже все можна зробити в інтернеті. Особливо зараз, коли в нашій країні складна ситуація, магазини не завжди можуть працювати в офлайн – режимі, але підтримка економіки на даний момент дуже важлива.

Магазини, які реалізують продукції на ринку, стикаються з проблемами конкурентоспроможності. Це відбувається через велику кількість підприємців, які прагнуть отримати як можна більший прибуток, від ведення діяльності в даному сегменті ринку. Для можливості конкурувати в даній ситуації, однією з головних потреб, є використання сучасних технологій, та інформаційних систем. Тому, зараз дуже зручно переходити на онлайн сервіси, такі як інтернет-магазин.

Такі технології, як web додатки, є дуже зручними, як для продавців так і для покупців. Клієнт може переглянути весь асортимент, його характеристику, порівняти ціни. Вибрати те, що найкраще йому підходить сидячи дома, не витрачаючи свого часу, на те щоб поїхати, стояти в черзі, і зараз це є небезпечно. Також, це допомагає продавцям, збільшується кількість часу, за рахунок автоматизації, робітники можуть виконувати інші функції, що призведуть до оптимізації управління компанією.

Кожен день технології йдуть вперед, за для автоматизації всіх процесів. Автоматизація – це велика можливість для маленьких компаній, які прагнуть розширитись.

Тому, метою кваліфікаційної роботи є, проектування та розробка веборієнтованої інформаційної системи для інтернет-магазину.

Задачі дослідження:

- аналіз діяльності підприємства «Vape Shop», проаналізувати танадати характеристику об'єкта дослідження;
- проаналізувати стан автоматизації бізнес-процесів;
- сформуванати вимоги до вебзастосунку;

- змоделювати бізнес-процеси;
- створення архітектури інформаційної системи та визначити технології вирішення поставлених задач;
- спроектувати веборієнтовану інформаційну систему, розробити прототип вебзастосунку;
- описати інструкції по використанню системи.

Для досягнення мети роботи використані методи дослідження: аналіз, індукція, дедукція, моделювання, аналогія.

Об'єктом дослідження є підприємство «Vape Shop», яке займається роздрібною торгівлею електронних випаровувачів.

Предметом дослідження є веборієнтована система для електронної комерції.

1 ДОСЛІДЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

1.1. Характеристика об'єкта дослідження, та аналіз стану автоматизації бізнес-процесів

Об'єктом кваліфікаційної роботи є підприємство «Vape Shop» міста Суми. Підприємство знаходить на ринку вже п'ять років, та займається торгівлею електронних випаровувачів. Це був першим магазин, в такому напрямку, що відкрився в місті Суми.

Головним елементом здійснення торгівельної діяльності, є задоволення потреб споживачів. Головними операціями такої діяльності є:

- прогнозування та дослідження споживчого попиту;
- аналіз попиту на товари;
- визначення та аналіз джерел закупівлі;
- вибір постачальників;
- встановлення ділових відносин з постачальниками товарів;
- знаходження та аналіз каналів просування товару.

Для правильної та вигідної закупівлі, є певна структура закупівельних робіт. Спочатку виконуються підготовчі роботи, робиться прогнозування попиту на товарну продукцію, досліджують можливість постачальників, та складаються проекти контракти. Прямі закупівлі ведуть за собою: складання контрактів, створення заявок на доставку, прийом товару. Кінцевою операцією є: контроль та встановлення на облік товару.

Головна мета підприємства, є отримання прибутку від реалізації продукції, в свою чергу діяльність спрямована на забезпечення населення товарами для електронних випаровувачів. Внаслідок своєї фінансово-господарської діяльності, підприємство отримує дохід, з якого регулярно виплачує різні платежі та податки, а також частина прибутку на даний момент, спрямована на підтримку нашої армії.

Внутрішньою складовою підприємства є власник, персонал, що стоїть за реалізацією продукції та бухгалтер.

До зовнішнього складового відносяться: покупці, постачальники, державні органи та конкуренти.

Підприємство знаходиться за адресою: м. Суми, вулиця Горького 23/1. Підприємство розташоване майже в центрі, тому до нього зручно дістатись.

Основним конкурентом в м. Суми є магазин Vape Express на вулиці Кооперативній.

Автоматизація на даний момент являється необхідною частиною бізнесу. Документообіг, це вже застаріла технологія, яка не є зручною, ефективною та прогресивною. Автоматизовані підприємства, що мають сучасне програмне забезпечення, мають сильнішу конкурентоспроможність більший прибуток, можливість до розширення бізнесу [2].

При надмірному навантаженні на роботі, навіть кваліфіковані працівники можуть допускати велику кількість помилок. І зазвичай таких ситуацій в торговій сфері дуже багато. Ручні процеси можуть дуже сильно впливати на прибутковість підприємства. І в не найкращу сторону. Тому, автоматизація процесів є одним з найкращих рішень для компанії, яка хоче покращити своє становище. В не залежності від того яких розмірів організація, всі будуть в вигоді від скорочення деяких процесів. Власники також звільняються від деяких процесів, і починають більше уваги приділяти саме розвитку бізнеса.

Бізнес-процес – логічна послідовність дій, процедур, для перетворення вхідних ресурсів, в кінцевий продукт, за допомогою працівників організації, за для задоволення потреб споживача. Відповідно, якщо такі процеси будуть автоматизовані, то споживач буде задоволений потребою швидше, що призведе до збільшення клієнтів, і виходячи з цього до покращення прибутку.

На даний момент на підприємстві «Vape Shop», автоматизація не на дуже високому рівні. Це, дуже уповільнюють весь процес роботи. Для обліку

товару, використовується програма «УКРСКЛАД». Є термінал, який прив'язано до цієї програми, також до «УКРСЛАДУ», прив'язана технологія CheckVox яка автоматично передає всі дані до податкової служби. Так ведеться облік товару.

Компанія має свій Телеграм-канал, на якому, кожен день виставляють товари, які є в наявності. Цим займається власник магазину. Тобто, замість того, щоб займатись розширенням та покращенням бізнесу, він весь день відповідає на повідомлення типу, «Що є в наявності?», «Скільки залишилося?», «Можна забронювати?».

Тому, проаналізувавши роботу підприємства, вирішено автоматизувати процеси обробки замовлень, для покращення прибутку, та стану бізнеса.

1.2. Модель бізнес-процесів

Для того, щоб розробити веборієнтовану систему, для початку необхідно проаналізувати всі бізнес-процеси на підприємстві. Після такого аналізу, буде зрозуміло, які етапи в бізнес-процесах можуть бути покращені за допомогою автоматизації системи [12].

Щоб проаналізувати бізнес-процеси, та наглядно їх зобразити, необхідно використовувати моделювання бізнес-процесів – це один з методів покращення ефективності роботи організації. В основі такого методу лежить процесний підхід до управління компанією [8-11].

Моделювання бізнес-процесів дозволяє сконцентруватися на конкретних аспектах, визначити властивості та зв'язки компонентів і продемонструвати їх як графічно так і текстом.

Кількість методів досить велика, але можна виділити найосновніші, та найпопулярніші, це UML, ARIS, IDEF0 [31-33].

Використовуємо саме методологію IDEF0, вона передбачає ієрархічну систему діаграм. Спочатку описуємо систему та зв'язки з навколишнім середовищем, далі робимо функціональну декомпозицію. На рисунку 1.2.1

зображена контекстна діаграма функціонування інформаційної системи підприємства [17-20].

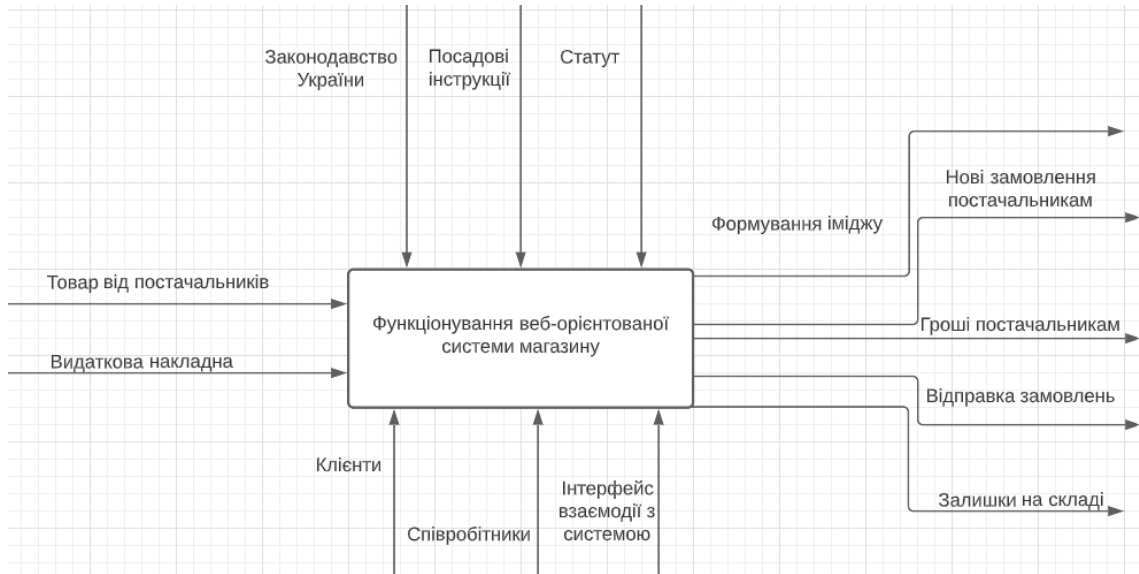


Рисунок 1.2.1 – Контекстна діаграма функціонування системи

Більш конкретною діаграмою є декомпозиція контекстної діаграми, зображена на рисунку 1.2.2.

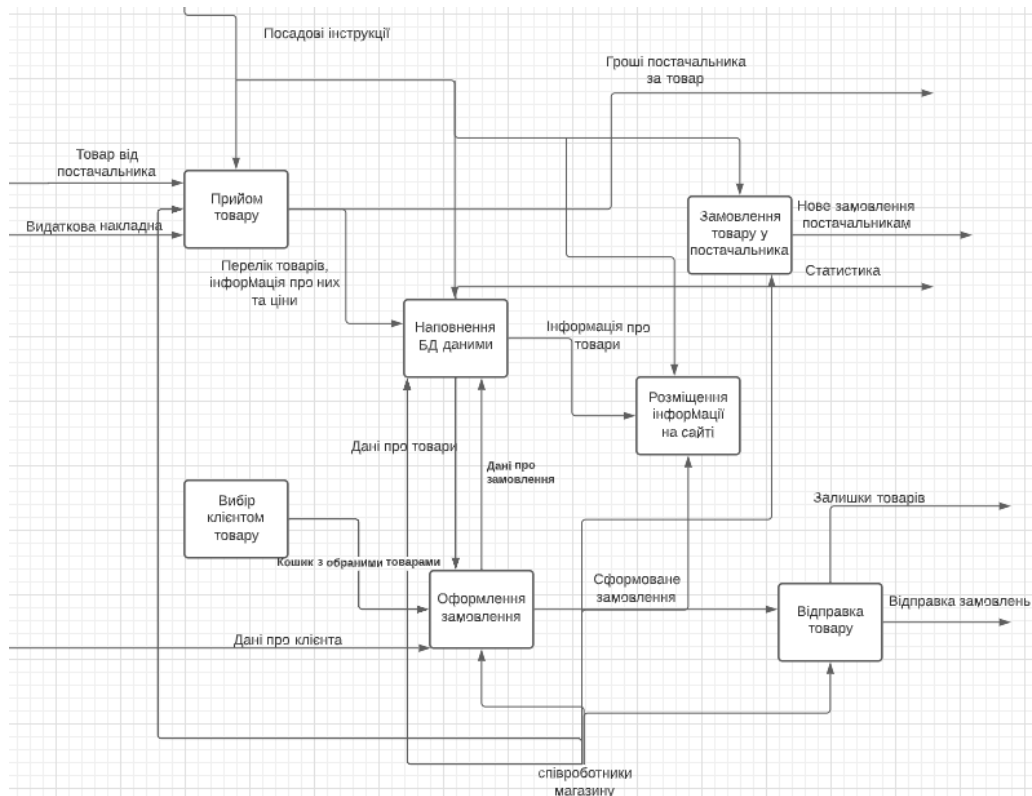


Рисунок 1.2.2 – Декомпозиція контекстної діаграми функціонування системи

1.3 Формування вимог до веборієнтованої системи

Веборієнтована система це спеціальний вебсайт, який призначений для збільшення обсягів продажу, просування товару, залучення більшої кількості клієнтів [21-26].

Особливостями веборієнтованих систем є те, що в них користувач може отримати велику кількість інформації про товари, і це допоможе прийняти краще рішення для покупки, також такі сайти можуть пропонувати товари, с огляду на те, що було переглянуто. Це дозволяє використовувати індивідуальний підхід до кожного користувача інтернет-магазинів [1, 8, 35].

Превагою торгівлі через веборієнтовану системи є те, що для них не потрібно витрат на оренду будівель, приміщень. Якщо немає можливості витратити велику кількість грошей на матеріально-технічну базу, то веборієнтована система саме для таких ситуацій.

Web орієнтована система, характеризується повною автоматизацією, це в свою чергу дає можливість індивідуально опрацьовувати кожне замовлення, з кожним клієнтом [3-7].

Для автоматизації процесів роботи магазину, прийнято рішення, створити Web-орієнтовану систему, з такими вимогами.

1. Головна сторінка, шапка Меню, якої містить перелік додаткових сторінок, кнопку Входу для реєстрації користувачів, та Кошик, в головній частині присутня назва магазину, та каталог товарів, в підвалі сторінки знаходиться інформація Про нас та Контакти (рис.1.3.1).

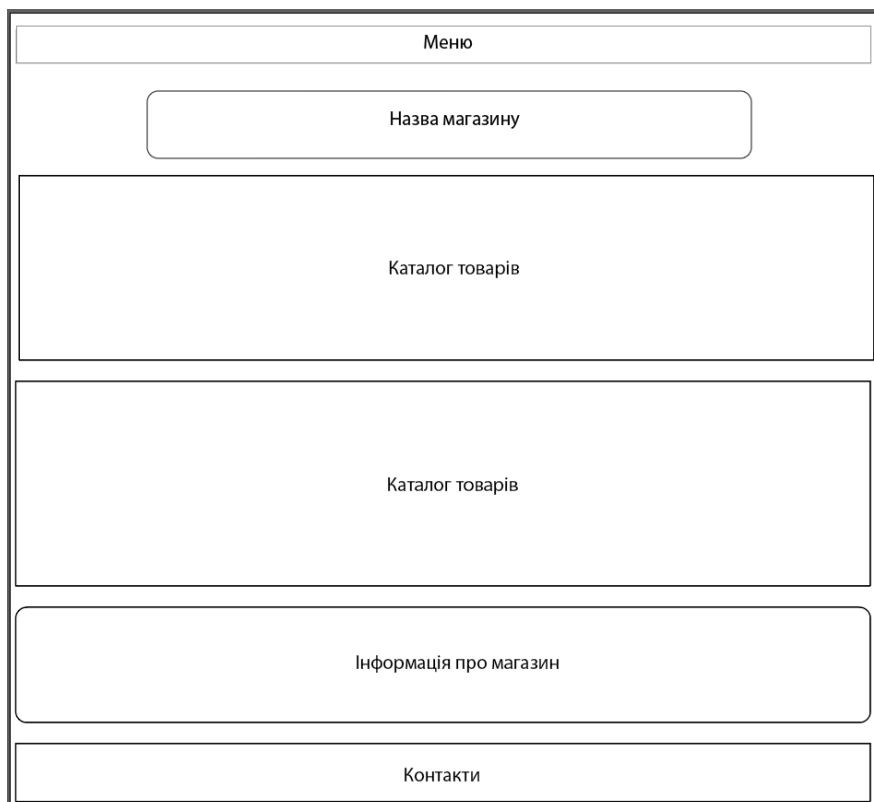


Рисунок 1.3.1 – Головна сторінка сайту

2. Сторінка каталогу товарів, містить товари, можливість фільтру (рис.1.3.2).

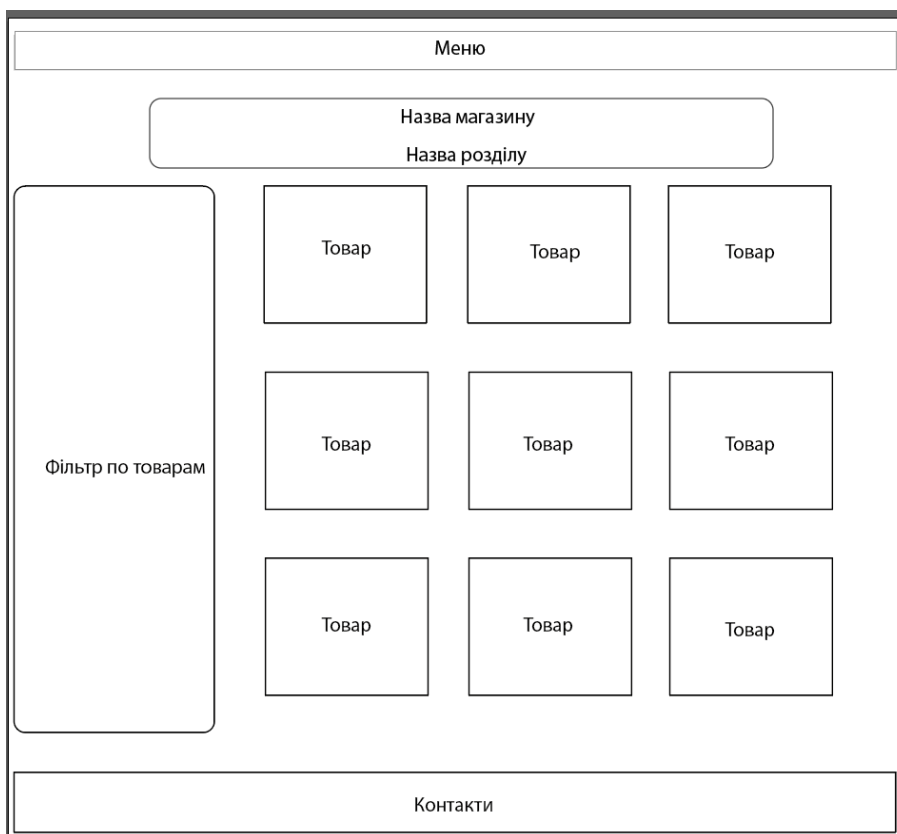


Рисунок 1.3.2 – Сторінка каталогу товарів

3. Сторінка для реєстрації покупця, зі вказанням номеру або пошти, та імені.

4. Картка товару, в якій знаходиться фото товару, повна характеристика, ціна, доступна кількість (рис. 1.3.3).

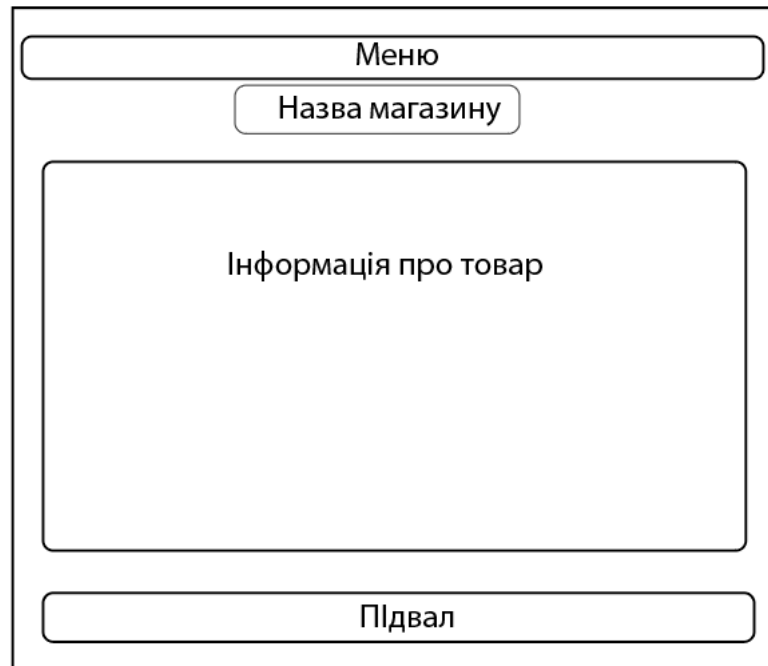


Рисунок 1.3.3 – Сторінка картки товару

5. Форма кошика, має містити: товар, ціну, коротку характеристику, можливість зміни кількості, загальну суму замовлення

6. Вибір методу оплати та доставки товару.

7. Додаткові інформативні сторінки, новини, огляд товарів (рис. 1.3.4).

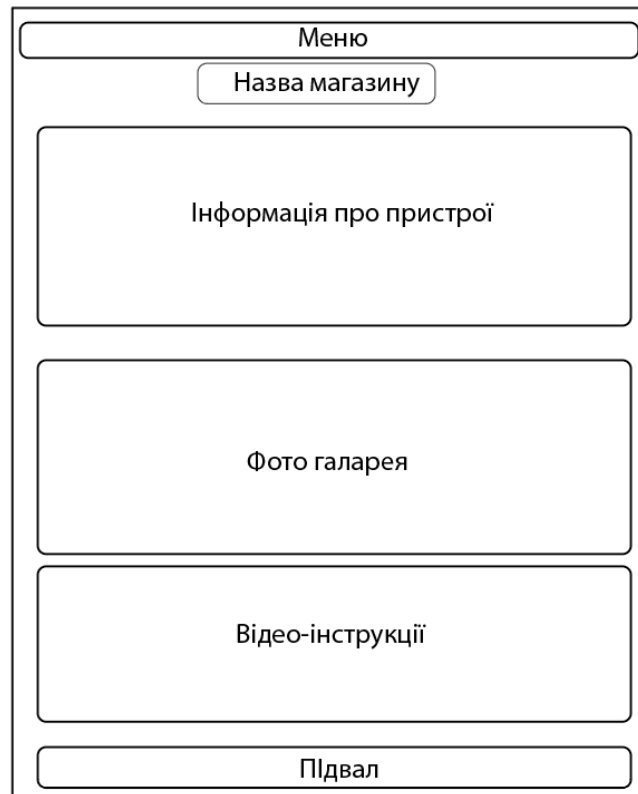


Рисунок 1.3.4 – Сторінка додаткової інформації

Пошук товарів в web орієнтованій системі здійснюється за допомогою каталогу . Тут користувач повністю може переглянути повну характеристику товару, переглянути фото або відео, дізнатись ціну та наявність на даний момент.

Під час перегляду товарів покупець формує власний кошик, в якому потім можна переглянути відібраний товар з ціною на нього, кількість, та загальну суму. Є можливість користувачеві зареєструватися, для того, щоб в подальшому переглядати історію своїх замовлень [1].

Для реєстрації покупцеві необхідно заповнити спеціальну форму, в якій буде знаходитись інформація про покупця, його ім'я, електронна адреса, номер телефону, поштова адреса, пароль. За допомогою спеціальних методів захисту інформації, такими засобами виступають протоколи SET, SSL, особиста інформація покупця при передачі даних є захищеною [35, 37].

Форми оплати присутні для вибору на сайті:

1. Оплата готівкою при доставці ним товару на адресу.

2. Через платіжні системи мережі Internet.

3. Накладений платіж на пошті.

4. При отриманні в магазині

Дотримуючись таких вимог, необхідно створити інтернет-магазин, для того, щоб робота була автоматизованою та ефективнішою .

2 РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНТЕРНЕТ МАГАЗИНУ

2.1.Проектування веборієнтованої інформаційної системи

Створення веборієнтованої системи для магазину надає такі переваги як, зручність, інформативність, індивідуальний підхід до кожного користувача сайту, інтерактивність. Аби ці переваги були доступні для кожного відвідувача сайту, треба відбудувати чітку архітектуру інформаційної системи, яка буде відповідати структурі інтернет-магазину.

Для того, щоб побудувати архітектуру, необхідно оцінити весь контент, який буде розміщений на вебсторінках інтернет-магазину.

На рисунку 2.1.1 відображена архітектура веборієнтованої системи.

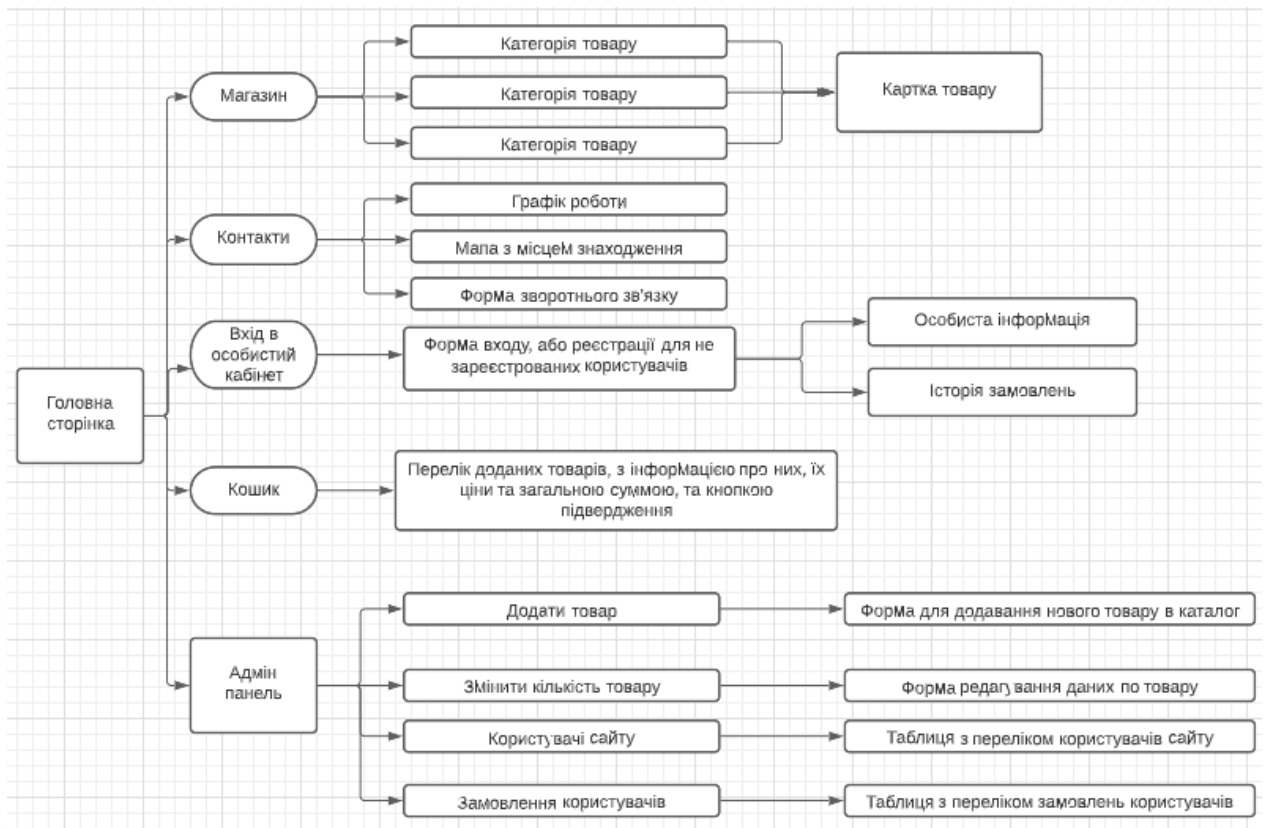


Рисунок 2.1.1 – Архітектура інтернет магазину

Вебсайт – це сукупність вебсторінок, доступних в інтернеті через протоколи https/http. Для створення сайтів на даний момент є велика

кількість платформ, фреймворків, всі вони досить різноманітні. Розробка веборієнтованої системи буде виконуватись не за допомогою шаблонізаторів, а за допомогою мови програмування Python [4-8].

Чому саме Python? За рахунок того що, в цій мові можливе використання стандартних бібліотек і сторонніх, його функціональні можливості розширюються. Python використовується для різних завдань, таких як: машинне навчання, створенні вебдодатків, створенні баз даних, аналіз даних, розробка ігор. Навіть така компанія, як Google використовує в свої розробці цю мову [7-9].

Синтаксис мови є достатньо зрозумілим, простим та лаконічним, що забезпечує доступність коду для читання, підтримки, виправлення та оновлення. Якщо порівнювати з іншими мовами, то код Python досить компактний у розмірі. Під час розробки компіляція не потрібна.

Під час розробки різних інтернет-проектів можна підключати зовнішні бібліотеки, що дадуть можливість створити сайти різного формату, та з широкою функціональністю.

На мові Python є велика кількість якісних фреймворків. Під час вибору фреймворка для розробки вебсервісу, треба розглянути всі їх можливості, та вибрати той, який найкраще буде підходити для типу розробки. Оскільки від цього вибору залежить кінцевий результат роботи [5, 13].

Якщо подивитись на графік (рис.2.1.2), то на даний момент можна виділити чотири з найпоширеніших фреймворків це Django, Flask, Tornado, web2py [5].

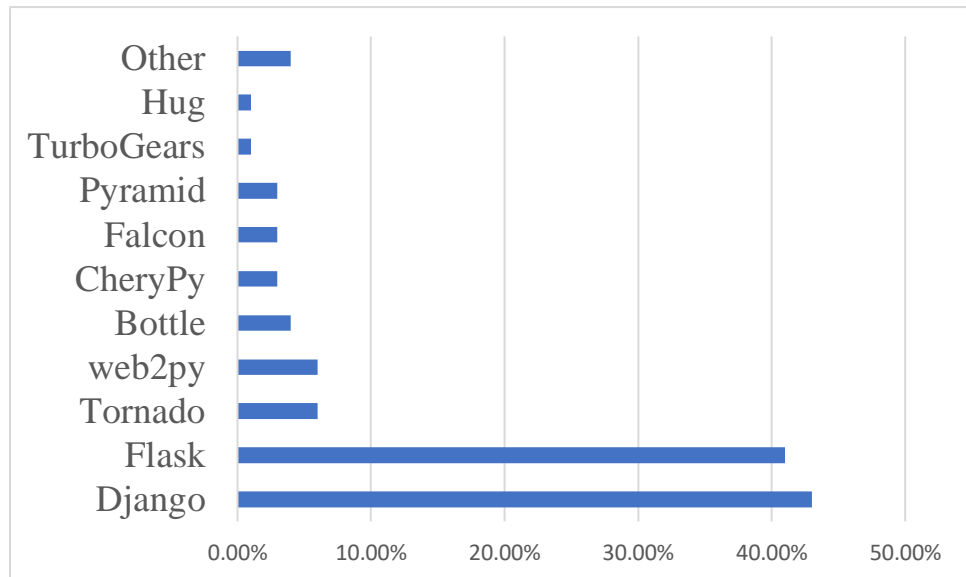


Рисунок 2.1.2 – Графік найпопулярніших фреймворків

Потрібно вибрати один фреймворк який найбільше підходить для створення інтерне-магазину. Переваги фреймворків:

Flask:

- висока продуктивність;
- мінімалістичний;
- велика можливість реалізацій;
- велика кількість пакетів;
- високий рівень захисту від витоку інформації та вебатак;
- влаштований сервер розробки.

Tornado:

- вбудована підтримка аутентифікації користувача;
- підтримка перекладу та локалізації;
- гарна продуктивність;
- якісна структура;
- неблокуючий HTTP-клієнт;
- реалізація сторонніх схем аутентифікації.

Web2py:

- відкритий код зі стеком Python;

- навчальні інструменти;
- зручний у використанні;
- відсутність вимог щодо налаштувань.

Django:

– Швидкість: розроблений для того, щоб створювати додатки не витрачаючи велику кількість часу. При створенні проекту, автоматично створюється повна структура, в яку залишається додати шаблони та дизайн. Тому, це є ідеальний фреймворк для програмістів [10-13].

– Комплектація: Django має додаткові функції, які допомагають з аутентифікацією, мапами сайту, адмініструванням, RSS і багато іншого.

– Безпека: використовуючи цей фреймворк, ви отримуєте захист від помилок, які можуть загрожувати вашому проекту.

– Різноманітність: Django досить ефективно справляється з менеджментом контенту, науковими обчислювальними платформами [6-8].

Таким чином, можна зробити висновок, що Django являється ідеальним інструментом для створення веборієнтованих систем .

В якості мов програмування, для розробки графічного інтерфейсу веборієнтованої системи обрано HTML та CSS, а також відкритий фреймворк Bootstrap.

Одним з найважливіших компонентів для створення вебдодатку є система управління БД. Вона буде забезпечувати отримання всіх даних, необхідних для інформаційного забезпечення сайту. В якості СУБД було обрано SQLite, влаштовану базу даних в Django.

Рисунок 2.1.3 показує функціональну схему інтернет-магазину, із зображення всіх функціонуючих компонентів

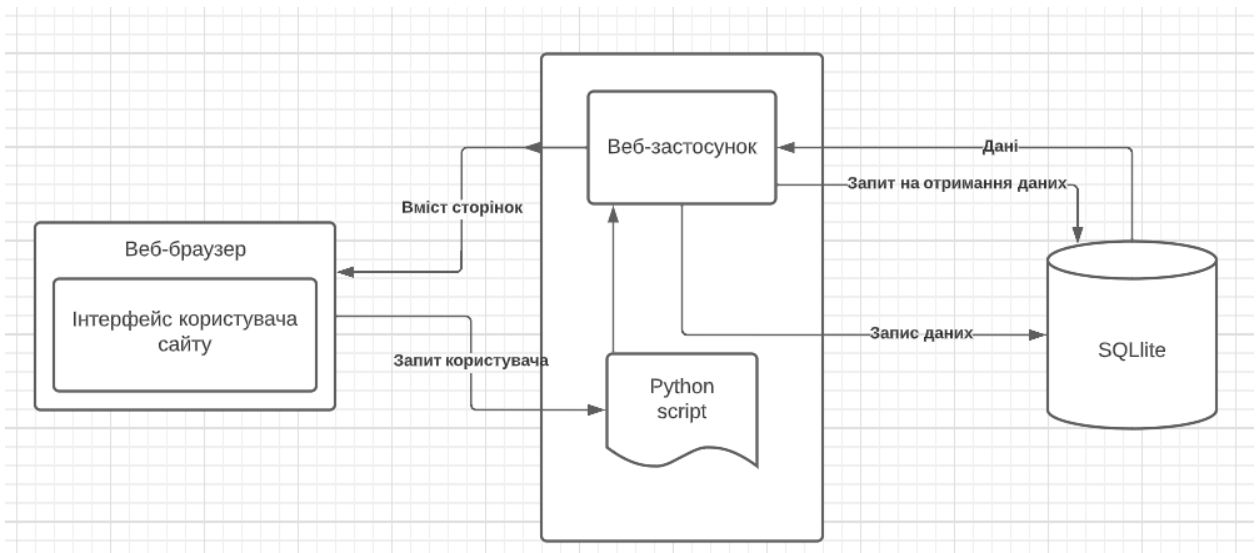


Рисунок 2.1.3 – Функціональна схема інтернет-магазину

Інформаційне забезпечення – це сукупність правил систем класифікації та кодування інформаційних даних, систем документів, структури потокової інформації, які присутні в організаціях. Можна сказати, що це сукупність всіх графічних, текстових матеріалів, посилань, за допомогою яких ми наповнюємо веборієнтовану систему.

Для того щоб користуватися вище-переліченими даними, необхідно створення реляційної бази даних, тобто електронних таблиць, які будуть зберігати всі дані, і до яких можна звернутися запитом.

База даних буде містити, всі дані, щодо товарів, їх кількості, наявності, та характеристики, всю інформацію по зареєстрованим користувачам, та по усіх замовленням на товари.

Виходячи з цього, нам треба створити три таблиці бази даних, вони матимуть такі назви:

- products(таблиця товарів);
- orders(таблиця замовлень);
- users(таблиця користувачів).

Всі таблиці матимуть зв'язок одна з одною, за реквізитом – унікальним ід в кожній таблиці.

На рисунку 2.1.4 зображена ER-діаграма, яка ілюструє зв'язки таблиць між собою всередині бази даних SQLite.

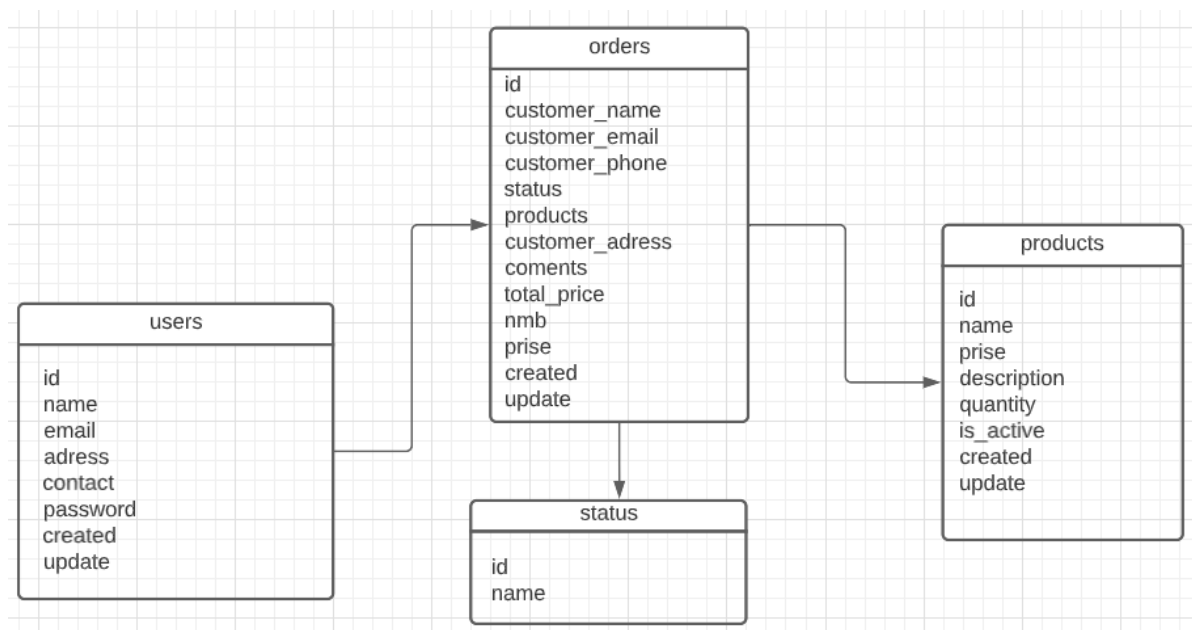


Рисунок 2.1.4 – Зв'язки таблиць в базі даних

Розглянемо структури таблиць детальніше.

В таблиці 2.1.5 зображена структура products, що містить в собі всі дані по товарам.

Таблиця 2.1.5 – Структура таблиці products

Номер атрибуту	Назва атрибуту	Тип даних	Порожнє значення	Призначення
1	name	CharField	None	Назва товару
2	prise	DecimalField	0	Ціна товару
3	description	TextField	None	Характеристика
4	Is_active	BooleanField	True	Наявність товару
5	quantity	DecimalField	0	Кількість

Структура orders, що зберігає дані по замовлення зображена на таблиці 2.1.6

Таблиця 2.1.6- Структура таблиці orders

Номер атрибуту	Назва атрибуту	Тип даних	Порожнє значення	Призначення
1	customer_name	CharField	0	Ім'я клієнта
2	customer_email	DecimalField	None	Емейл клієнта
3	customer_phone	CharField	None	Номер клієнта
4	status	class	None	Статус замовлення
5	product	TextField	None	Товар
6	customer_adress	CharField	None	Адреса доставки
7	coments	CharField	None	Коментарії
8	total_price	DecimalField	0	Загальна сума
9	nmb	IntegerField	1	Кількість товару
10	prise	DecimalField	0	Ціна товару

Структура user, що зберігає дані по користувачам сайту зображена в таблиці 2.1.7.

Таблиця 2.1.7 – Структура таблиці products

Номер атрибуту	Назва атрибуту	Тип даних	Порожнє значення	Призначення
1	name	CharField	None	Ім'я користувача
2	email	EmailField		Email користувача
3	phone	CharField		Номер користувача

За допомогою типового файлу `admin.py` всі ці моделі пов'язуються, з використанням допоміжних класів, їх можна переглянути в додатку Б.

2.2. Розробка прототипу вебзастосунку

PyCharm – це платформа розроблена для професійної розробки на мові Python. В ньому присутня велика кількість інструментів.

Переваги PyCharm:

- підтримка великої кількості пакетів;
- інтелектуальна реорганізація коду;
- в ньому присутня перевірка коду на помилки, що допомагає в розробці;

- досить простий в використанні;

- присутній інтегрований SSH термінал, та інтеграція с Docker.

Починаючи розробку сайту спочатку потрібно створити Django Project. Щоб була можливість створювати проект з вже встановленим Django, треба використовувати PyCharm Pro. Створюємо проект Django (рис.2.2.1).

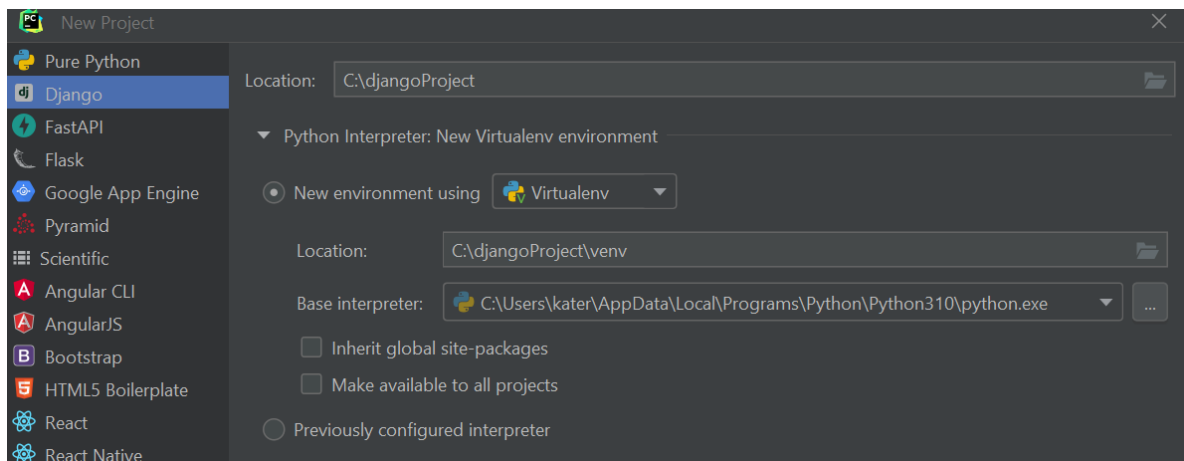


Рисунок 2.2.1 – Створення проекту Django

Як тільки проект буде створений, директорія проекту Django матиме такий вигляд:

- `djangoProject / settings.py` – конфігурація проекту;

– *djangoProject urls.py* – оголошуються всі URL використані в проекті;
– *manage.py*- за допомогою цього файлу, ми будемо робити запити до проекту .

Після створення такої структури, можемо запустити проект, прописуємо в терміналі такий код:

```
Python manage.py runserver
```

Якщо все вірно, то в результаті з'явиться посилання на сервер з проектом в такому вигляді – <http://127.0.0.1:8000/> . Переходимо по ній.

Нам буде відкрита типова сторінка від Django. Далі переходимо до створення додатку. В терміналі прописуємо такий код, в кінці вказуємо назву додатку:

```
Python manage.py startapp shop
```

Після того як є створений пустий додаток, треба розробити структуру, яка зображена на рисунку 2.2.2.

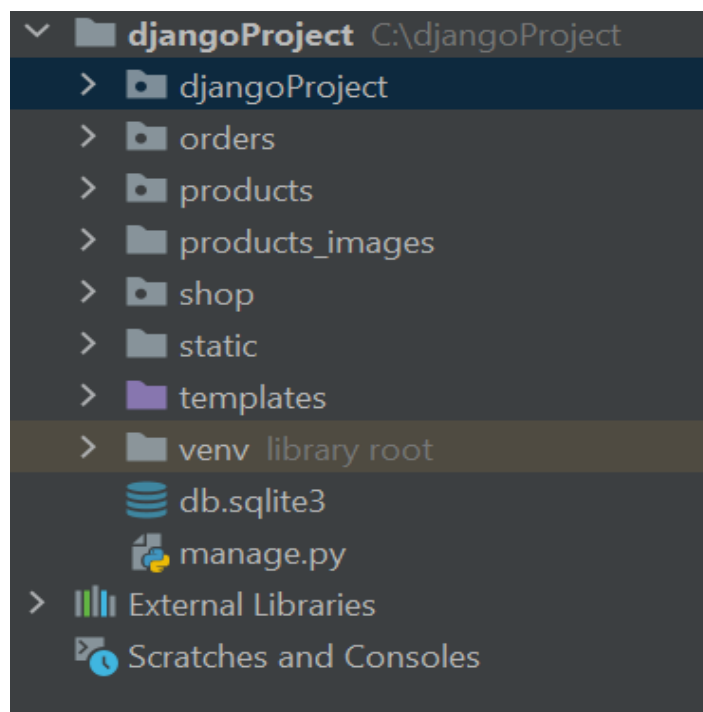


Рисунок 2.2.2 – Структура проекту Django

Опис структури файлів:

orders – в папці знаходяться файли в який описується робота с замовленнями

products – в папці знаходяться всі файли в яких описана структура товарів

static – в папці розташовуються файли стилів сайту, скрипти зображення тощо.

templates – в папці знаходяться шаблони сторінок сайту

db.sqlite3 – це база даних проекту

Після того як створено всю структуру, переходимо до наповнення всіх файлів кодом. Спочатку в головних файлах проекту, таких як *settings.py*, *urls.py*, треба прописати головні налаштування. Для того, щоб була можливість використання файлів CSS та зображень, в файлі *settings.py* треба прописати такий код:

```
MEDIA_URL = '/image/'

MEDIA_ROOT = os.path.join(
    BASE_DIR, 'static/image'
)

STATICFILES_DIRS = [
    os.path.join(BASE_DIR,
        "../..../djangoProject/static"), ]
```

Далі, треба в файлі *urls.py* прописати всі додатки які будуть використані на сайті:

```
from django.contrib import admin
from django.urls import path, include
admin.autodiscover()

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('shop.urls')),
    path('', include('products.urls')),
    path('', include('orders.urls')),
]
```

Тепер переходимо до заповнення коду в інших файлах. Для початку створюємо структуру бази даних. Оскільки нам треба, щоб користувач міг зареєструватися, переглянути товар, та замовити, необхідно створити форми та моделі бази даних (Додаток Б). Після створення форм та моделей треба в консолі прописати код для міграції в базу даних, в такому вигляді (після кожного натискаємо Enter):

```
Python manage.py makemigrations shop
```

```
Python manage.py makemigrations products
```

```
Python manage.py makemigrations orders
```

```
Python manage.py migration
```

Запускаємо сервер, та в результаті будемо мати такий вигляд на адмін панелі (рис. 2.2.3).

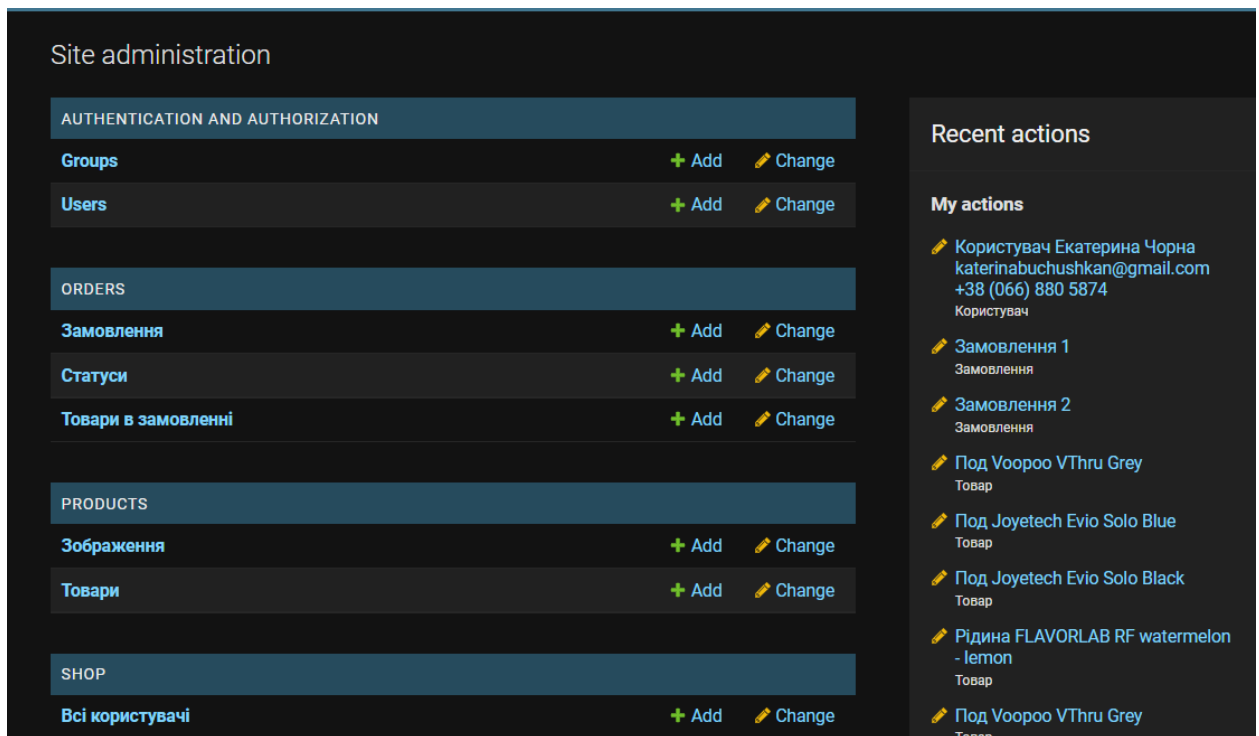


Рисунок 2.2.3 – Структура бази даних магазину на Django

Далі необхідно створити макети для сторінок сайту (Додаток В). Для сторінок використовуємо, як свої самописні стилі, так і інтегровані стилі з

Bootstrap. Коли весь код готовий, запускаємо сервер і переглядаємо сайт [8,9].

2.3. Інструкції по використанню системи

Розглянемо інструкцію для користувачів без прав адміністратора. На головній сторінці, знаходиться банер з логотипом магазину, групи товарів, опис магазину, режим роботи та контакти.

Вверху сторінки є меню за допомогою якого, здійснюється перехід до каталогу товарів, форми реєстрації, кошика, та інших додаткових сторінок (рис. 2.3.1).

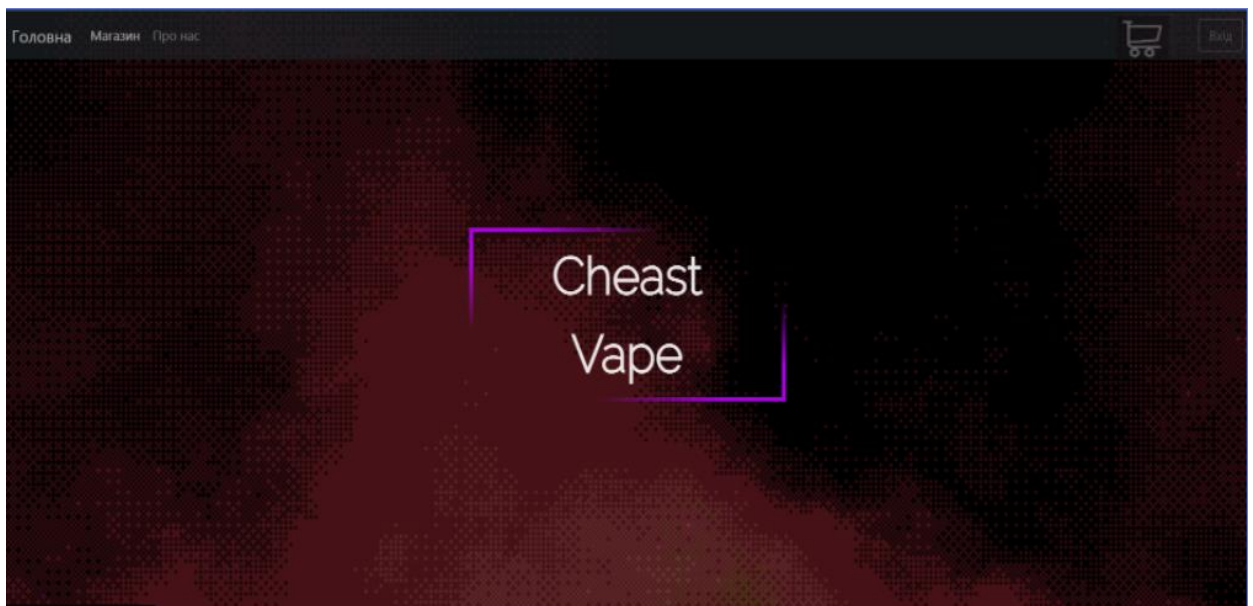


Рисунок 2.3.1 – Головна сторінка сайту

Далі на сторінці перелік категорій товарів (рис.2.3.2) та розділ «Про нас». Для переходу до каталогу товарів, необхідно натиснути на одну з категорій.



Рисунок 2.3.2 – Відділ товарів а головній сторінці сайту

Підвалом головної сторінки являється розділ, як знайти магазин на мапі, графік роботи, то способи зв'язку (рис.2.3.3).

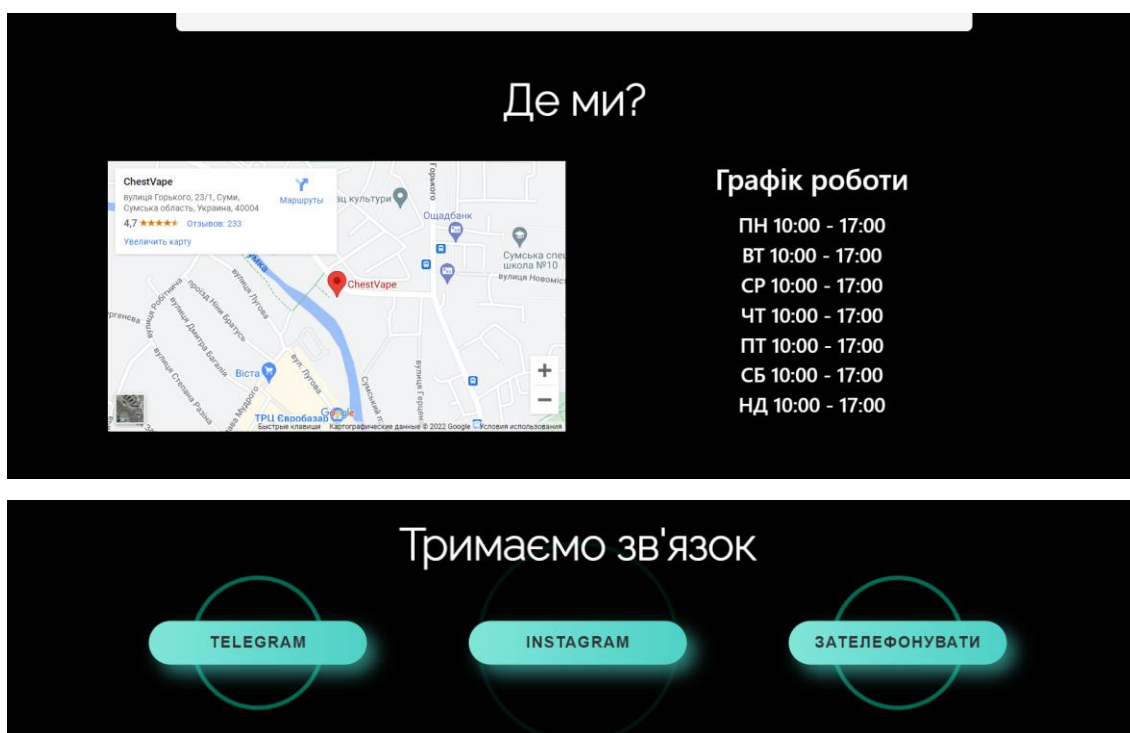


Рисунок 2.3.3 – Підвал головної сторінки сторінка сайту

Для переходу в магазин товарів треба, або натиснути на кнопку «Магазин» в шапці сторінки, або на товар в категоріях. Далі відкривається сторінка з товарами (рис.2.3.4).

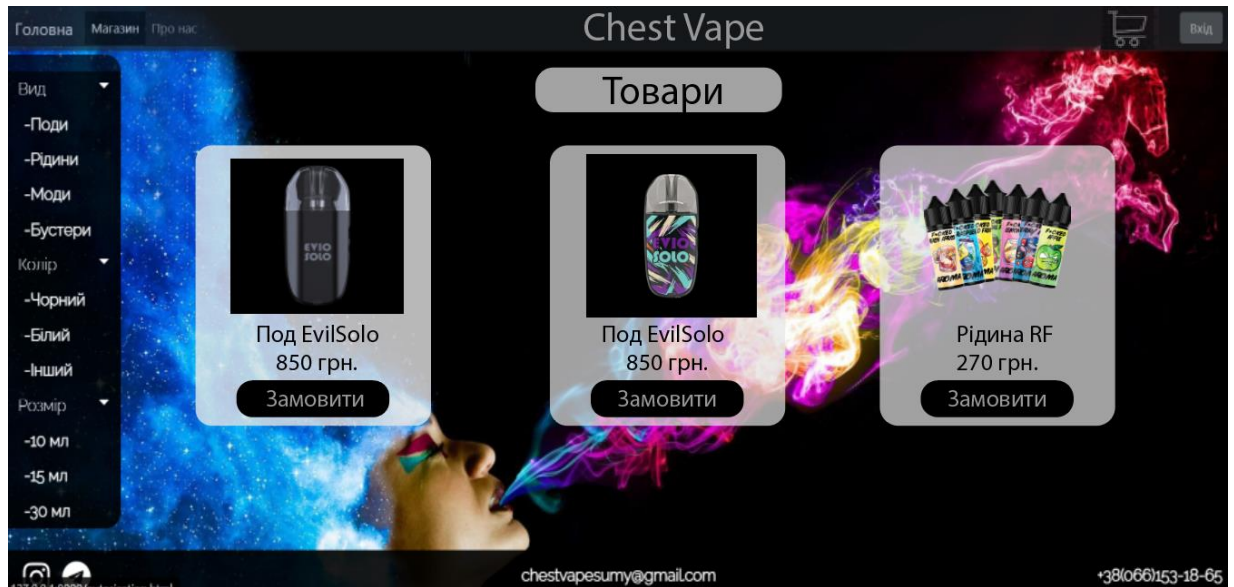


Рисунок 2.3.4 – Сторінка товарів

Щоб переглянути товар треба натиснути на нього, і відкриється картка товару (рис. 2.3.5).

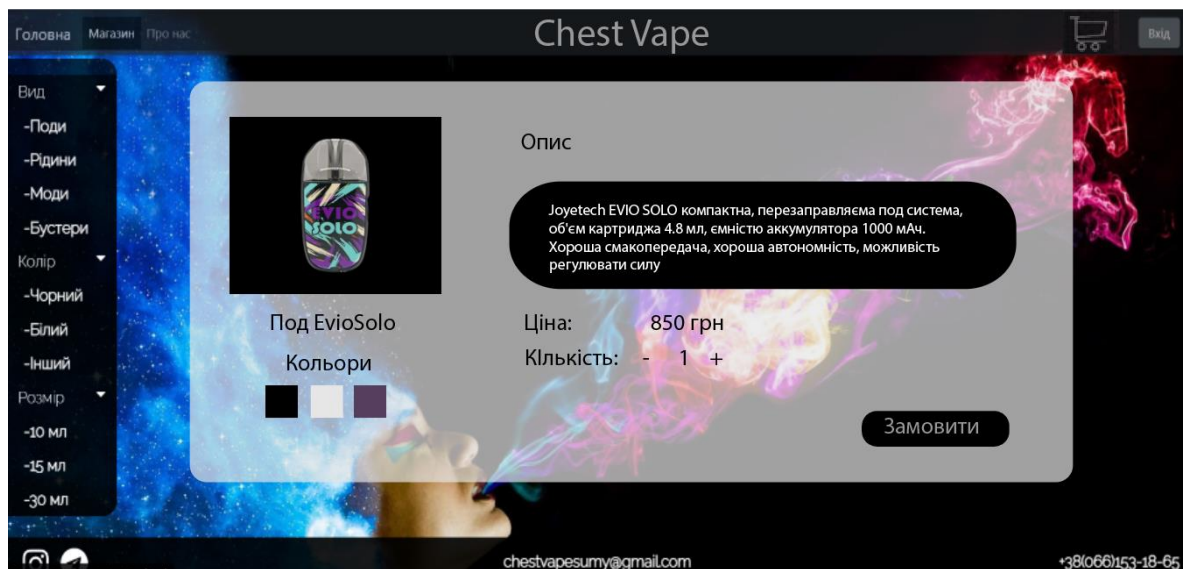


Рисунок 2.3.5 - Картка товару

Для замовлення необхідно натиснути кнопку «Замовити», і товар одразу потрапляє до кошику (рис.2.3.6).



Рисунок 2.3.6 – Додання товару в кошик

Для покупки товару, є два варіанти, перший це зареєструватися, і потім можна спостерігати історію своїх замовлень, другий варіант – швидке замовлення. Для першого необхідно в шапці меню натиснути на кнопку «Вхід», відкривається форма реєстрації (рис.2.3.7), заповнюємо дані, натискаємо Зареєструватись.

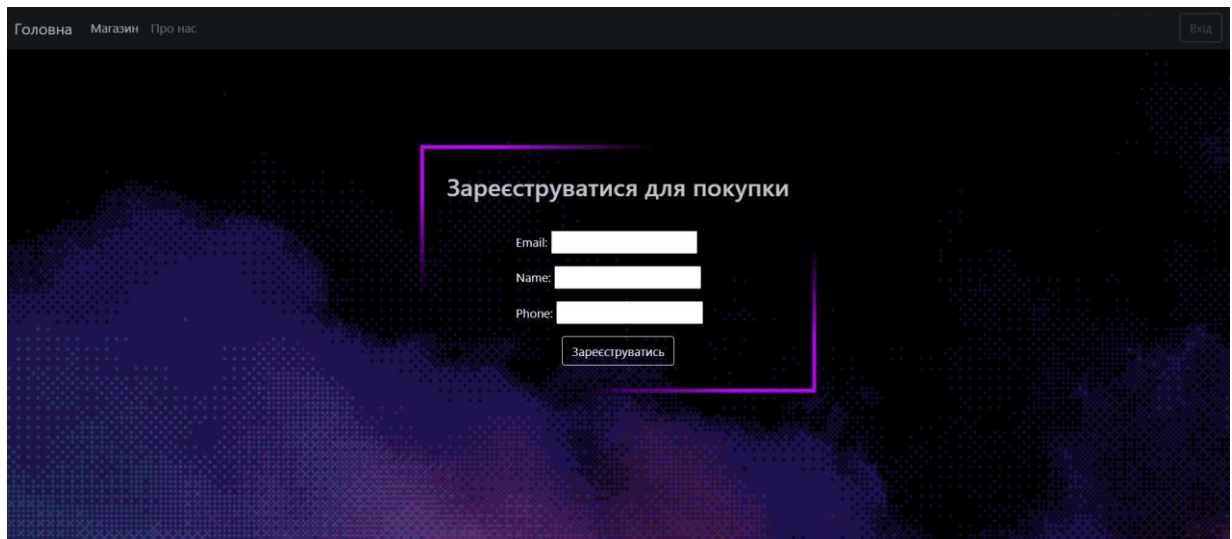


Рисунок 2.3.7 – Форма реєстрації

Якщо ж потрібно швидко замовити, то необхідно перейти до «Кошику», підтвердити замовлення (рис.2.3.8) та вказати дані для зв'язку

(рис.2.3.9). Після цього з'явиться модальне вікно, з повідомленням про підтвердження (рис.2.3.10)

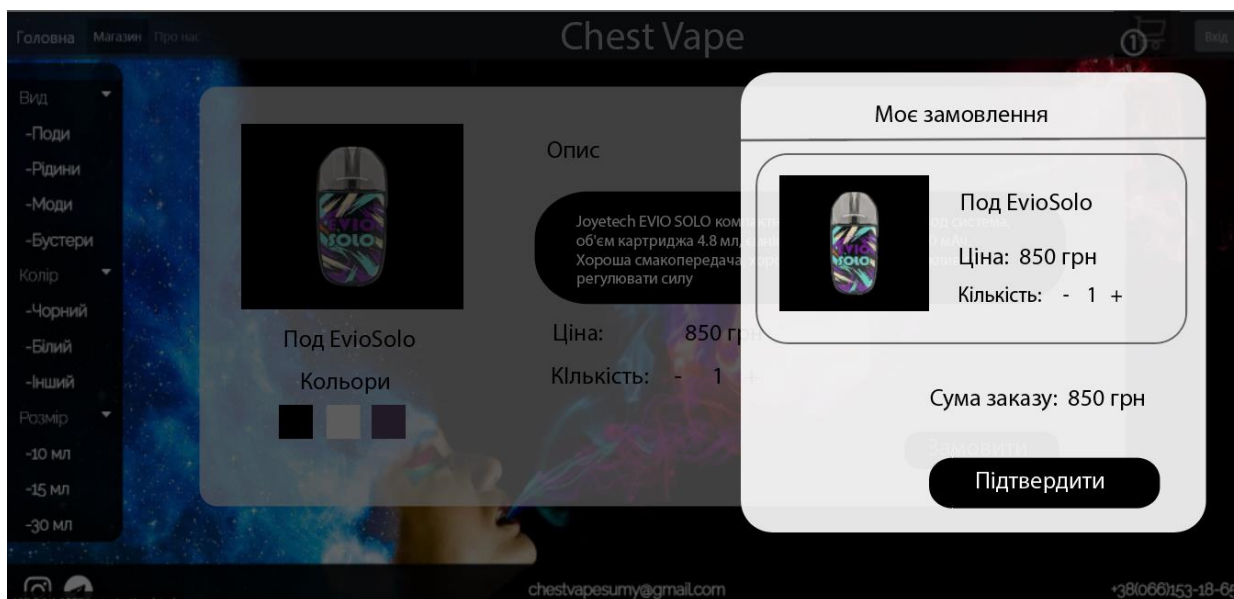


Рисунок 2.3.8 – Форма кошика

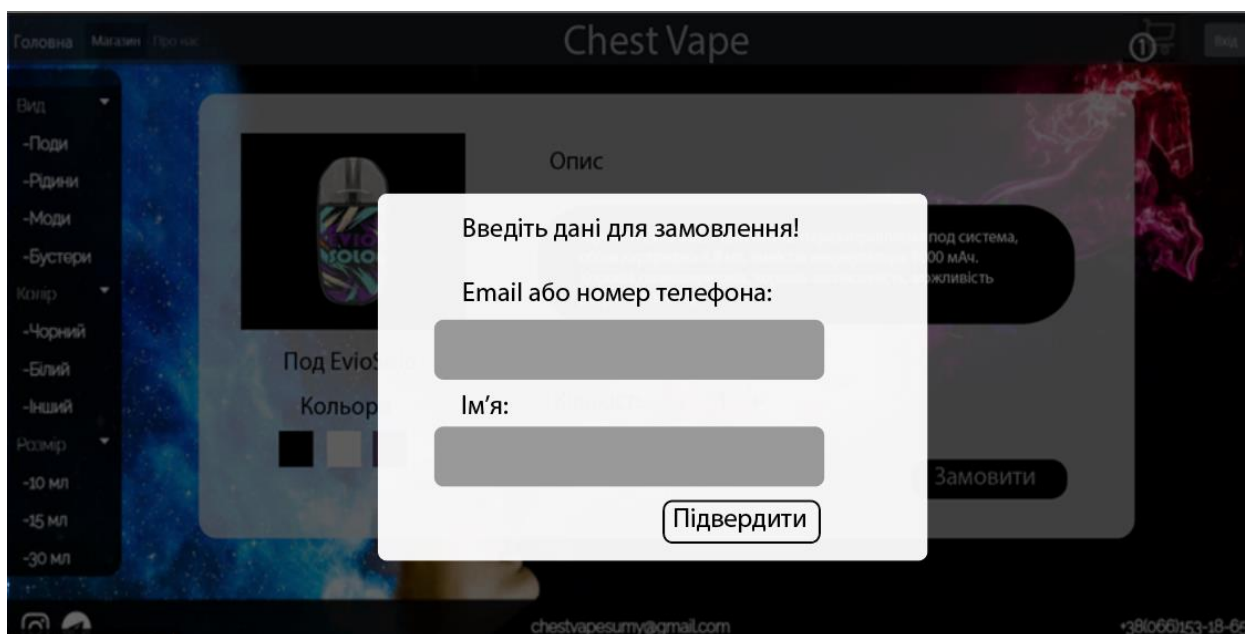


Рисунок 2.3.9 – Форма для заповнення даних для зв'язку

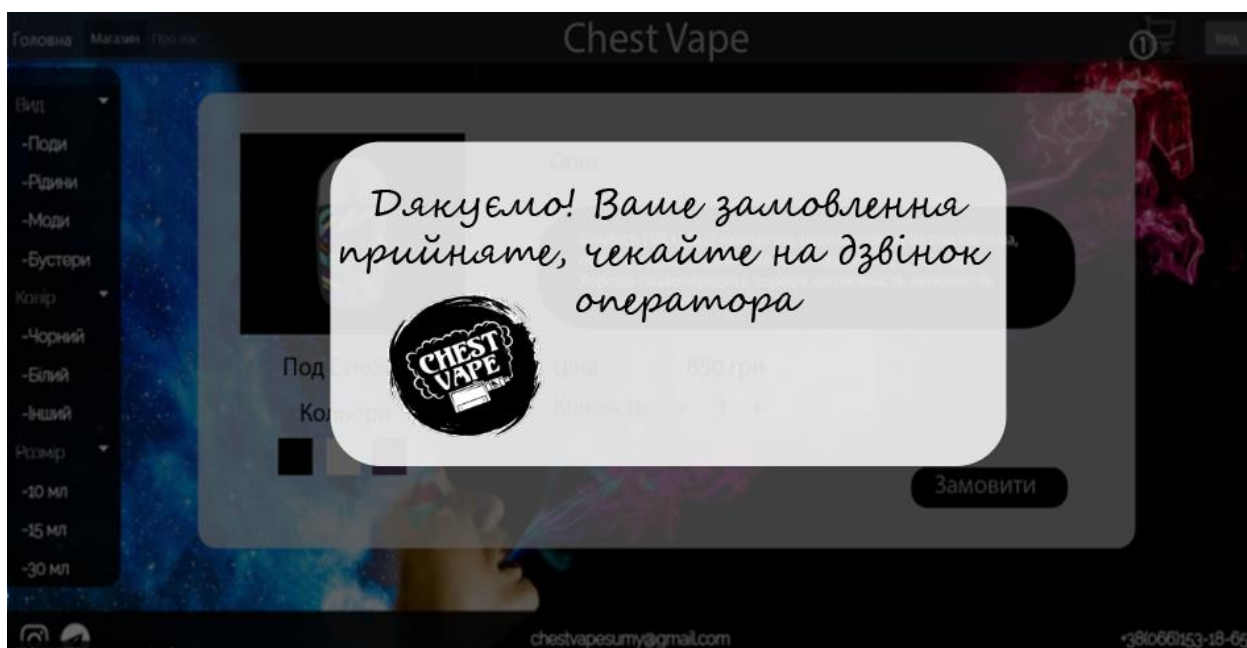


Рисунок 2.3.10 – Підтвердження замовлення

Також, в шапці сторінки присутні додаткові інформативні сторінки, перейшовши на них, бачимо інформацію про магазин, про пристрої різних компаній, та відео-інструкцію по користуванню ними (рис. 2.3.11).

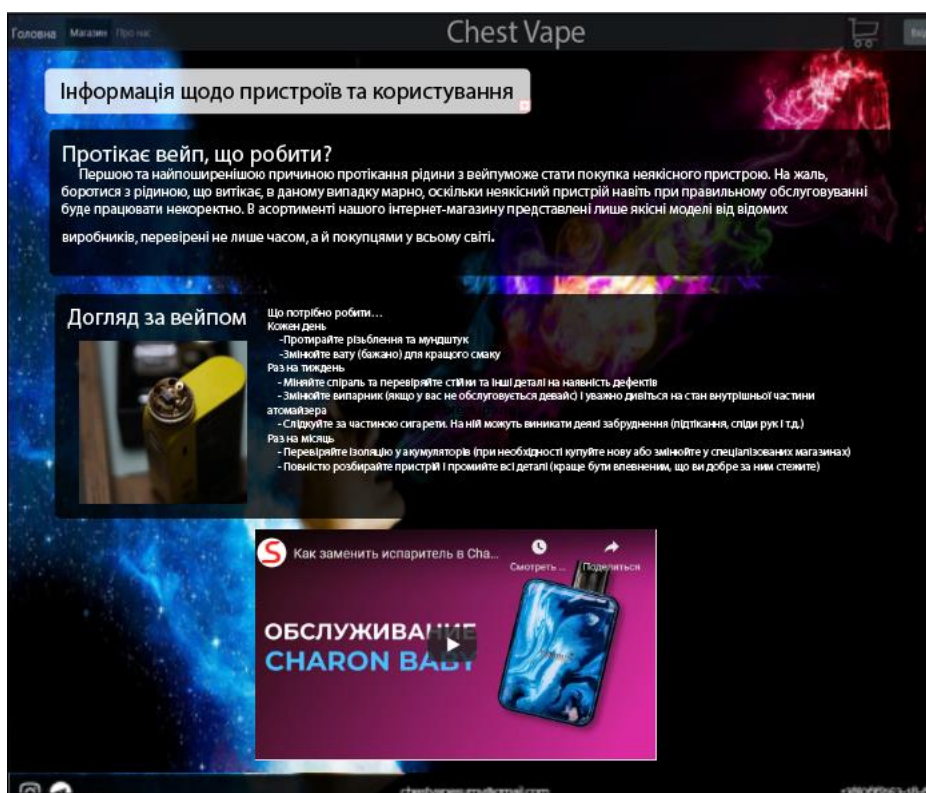


Рисунок 2.3.11 – Інформативна сторінка

Коли користувач вказав дані та підтвердив замовлення, адміністратору сайту на пошту надходить повідомлення з даними замовлення.

Якщо користувач входить під Адміністратором, то шапка сторінки має інший вигляд, в ній присутні: Товари, Замовлення та Користувачі. Натиснувши на Товари, відкривається сторінка з формою, на якій адміністратор переглядає весь товар, скільки його залишилось, а також добавляє новий (рис. 2.3.11).

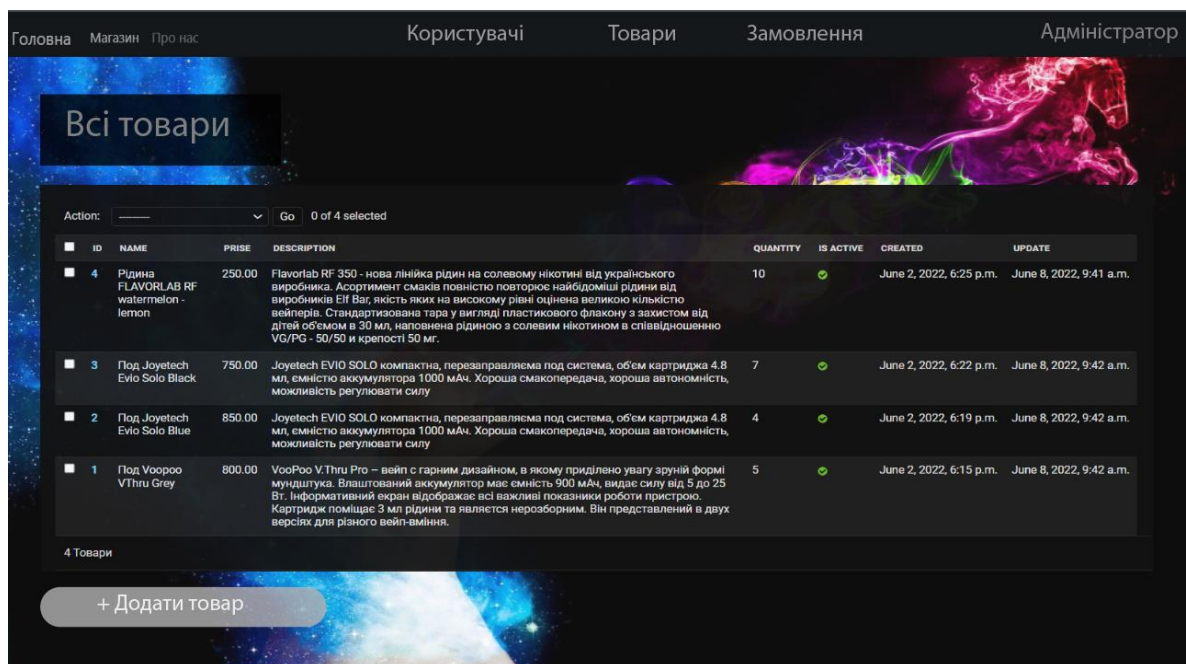


Рисунок 2.3.11 – Адмін панель для додавання товарів

Щоб додати товар, треба натиснути на кнопку «+Додати товар», відкриється форма з даними по товару, заповнюємо, завантажуюмо фото товару, та зберігаємо (рис.2.3.12).

Головна Магазин Про нас Користувачі Товари Замовлення Адміністратор

Товар

Name: Рідина FLAVORLAB RF watermelon - lemon

Price: 250,00

Description: Flavorlab RF 350 - нова лінійка рідин на солевому нікотині від українського виробника. Асортимент смаків повністю повторює найвідоміші рідини від виробників Elf Bar, якість яких на високому рівні оцінена великою кількістю вейперів. Стандартизована тара у вигляді пластикового флакону з застосом від дільця об'ємом в 30 мл, наповнена рідиною з солевим нікотином в співвідношенню VG/PG - 50/50 и крепості 80 мг.

Quantity: 10

Is active

ЗОБРАЖЕННЯ

IMAGE	IS ACTIVE	DELETE?
4	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Currently: products_images/watermelonlemon-600x600.webp

change: Файл не вибран

Файл не вибран

Рисунок 2.3.12 – Форма для заповнення даних товару

Для перегляду всіх замовлень користувачами, натискаємо на пункт меню замовлення, відкривається форма, на якій знаходяться всі замовлення. З такими полями, як: хто замовив, номер, статус замовлення, дата, та адреса доставки (рис.2.3.13).

Головна Магазин Про нас Користувачі Товари Замовлення Адміністратор

Всі замовлення

Action: Go 0 of 5 selected

ID	TOTAL PRICE	CUSTOMER EMAIL	CUSTOMER NAME	CUSTOMER PHONE	IS ACTIVE	COMMENTS	CREATED	UPDATE	CUSTOMER ADDRESS
5	2200.00	igniteregorr@gmail.com	Егор Панфілов	0661531865	Завершений		June 8, 2022, 8:10 p.m.	June 8, 2022, 8:10 p.m.	Сад, Нова пошта 1
4	1350.00	fedora@gmail.com	Максим Федора	0508735674	В роботі		June 8, 2022, 8:07 p.m.	June 8, 2022, 8:07 p.m.	Суми, Нова пошта 7
3	850.00	zmyslya@gmail.com	Назар Змисля	0957766433	Принятый		June 8, 2022, 8:07 p.m.	June 8, 2022, 8:07 p.m.	Суми, Нова пошта 7
2	500.00	Katya@gmail.com	Екатерина Чорна	+380668805874	Завершений		June 2, 2022, 8:41 p.m.	June 8, 2022, 9:47 a.m.	Суми, Нова пошта 17
1	850.00	Katya@gmail.com	Екатерина Чорна	+380668805874	В роботі		June 2, 2022, 6:53 p.m.	June 8, 2022, 9:47 a.m.	Суми, Нова пошта 1

5 Замовлення

Рисунок 2.3.13 – Форма всіх замовлень клієнтів

Щоб побачити повну інформацію по замовленню, натискаємо на замовлення, та бачимо всі дані (рис.2.3.14). Адміністратор може змінювати статус замовлення.

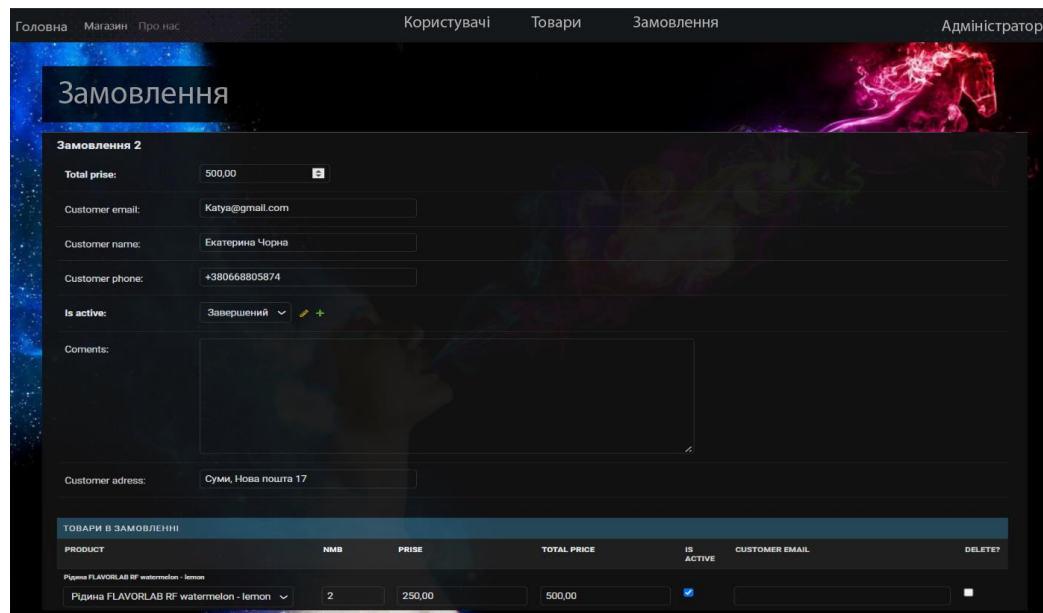


Рисунок 2.3.14 – Форма конкретного замовлення

Для перегляду адміністратору всіх зареєстрованих користувачів на сайті, треба натиснути на пункт меню Користувачі, відкриється форма з даними (рис.2.3.15).

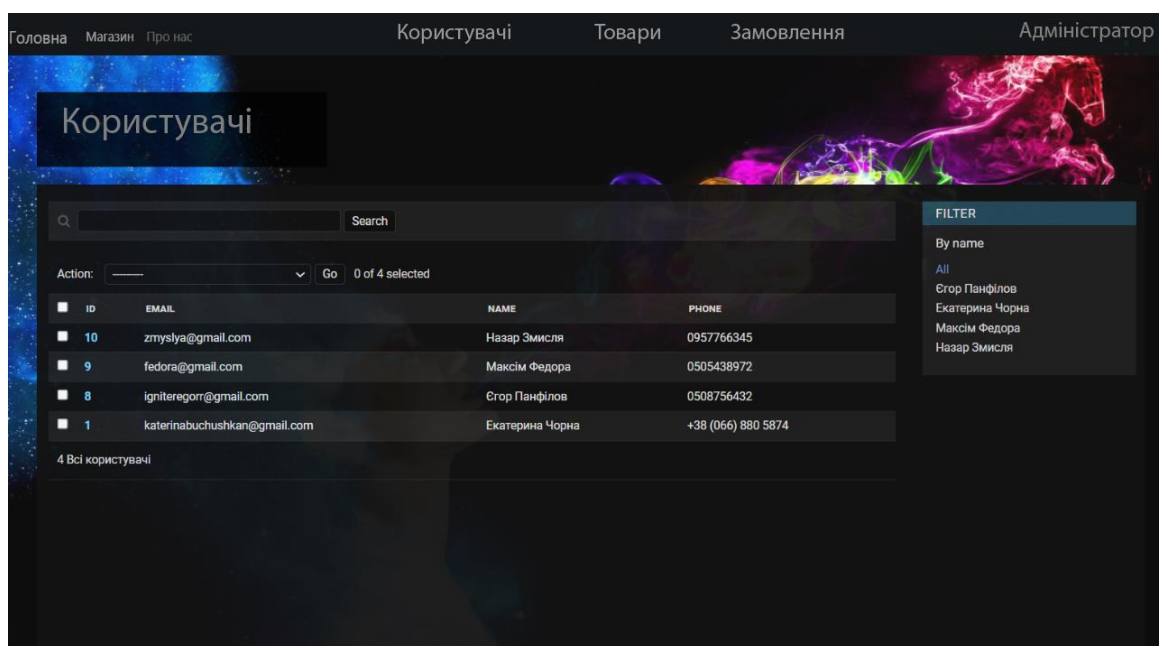


Рисунок 2.3.15 – Форма всіх зареєстрованих користувачів сайту

ВИСНОВКИ

На даний час створення інтернет магазину, являється одним з найважливіших моментів для ведення успішного бізнесу. Це дуже зручно, на це не треба постійно витратити великий бюджет коштів, та такий сервіс надає швидку передачу інформації. Все вище перераховане служить основою монетизації бізнесу та підвищенню конкурентоспроможності.

В результаті виконання кваліфікаційної роботи детально вивчено рівень автоматизації підприємства «Vape Shop» та вказано на необхідність впровадження веборієнтованої інформаційної системи.

Очікуваним результатом власника підприємства після впровадження онлайн-магазину є:

- збільшення прибутку;
- збільшення вільного часу у власника для розвитку бізнеса;
- збільшення кількості покупців;
- збільшення об'єму продаж;
- посилення конкурентоспроможності;
- розширення ринку продаж.

Виконані задачі в результаті кваліфікаційної роботи:

- надано характеристику діяльності підприємства «Vape Shop»;
- проаналізовано стан автоматизації бізнес-процесів;
- сформовані вимоги до вебзастосунку;
- змодельовані бізнес-процеси;
- створена архітектури інформаційної системи та визначені технології вирішення поставлених задач;
- спроектована веборієнтована інформаційна система та розроблений її прототип;
- описані інструкції по використанню системи.

Після запуску сайту, проведено аналіз, і результат такий, що за день було виконано 200 замовлень, замість 80 в звичайному офлайн режимі та телеграм каналі.

Таким чином, можна вважати, що мета кваліфікаційної роботи була досягнута, поставлені завдання є виконаними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматизація бізнесу. URL: <https://a4.com.ua/avtomatizatsiya-biznesu-dlya-chogo-avtomatizuyut-pidpriemstva/>
2. Автоматизація підприємства. URL: <https://ittel.com.ua/avtomatizaciya-pidpriemstva/>
3. Админ панель. URL: <https://smarticle.ru/admin-area-admin-panel-cmswordpress/> (Дата звертання 07.06.2020)
4. Блог, як створити сайт. URL: <https://ua.weblium.com/blog/yak-ctvoriti-sajt-bezkoshtovno-ta-shvidko-9-krokv>
5. Вебтехнологии. URL: <https://plarson.ru/technologies.html>
6. Вимоги до оформлення бібліографічного опису літературних джерел. URL: <https://www.narodnaosvita.kiev.ua//>;
7. Зандстра М. PHP: объекты, шаблоны и методики программирования. 3-е издание. СПб. Вильямс, 2012. 480с
8. Євтушенко Д.Д. Електронний бізнес, електронна комерція, Інтернет-торгівля: сутність та взаємозв'язок понять [Електронний ресурс] Бізнес Інформ. 2014. URL: http://nbuv.gov.ua/UJRN/binf_2014_8_33.
9. ТОП 10 кращих фреймворків для Python. URL: <https://django.fun/>
10. Д. Форс, П. Бісекс, У. Чан Django: Розробка вебдодатків на Символ-плюс, 2017.– 267 с.
11. Інформаційні системи. URL:<https://nauka-online.com/ua/publications/informatsionnye-tehnologii/2018/12/obzor-frejmworkov-i-razbor-plyusov-i-minusov-django/>
12. Методология IDEF0. URL: <https://itteach.ru/bpwin/metodologiya-idef0> (Дата звернення 21.04.2021);
13. Методология IDEF (Integrated DEFinition) моделирования бизнес-процессов. URL: <https://www.r-p-c.ru/> (Дата звернення 17.04.2021);
14. Кравець В. Інтернет-комерція в Україні. Вісник НБУ. 2014. № 3. С. 917.

15. Кроудер Д. Разработка Web-узлов для «чайников» [Текст]. Диалектика: Москва, С.-Петербург, Киев, 2010 р. 336 с.
16. Особливості плагіну. URL: <https://siteactiv.ru/> .(Дата звернення 30.04.2021);
17. Переваги та недоліки Python. URL: <https://xn----7sbbaqhlkm9ah9aiq.net/ua/news-new/preimushchestva-i-nedostatki-sayta-na-python.html>
18. Розробка сайта. URL: <https://artjoker.ua/ua/uslugi/razrobotkastruktury-saita/> (Дата звернення 21.04.2021);
19. Реляционные базы данных. Понятие ключа. Виды ключей. Отношения. Главная и подчиненная таблицы. URL: <https://www.bestprog.net/ru/>
20. Світові технології. URL: <https://www.ltcompany.com/ru/>. (Дата звернення 15.05.2021);
21. Трьохшарова архітектура. URL: https://cad.kpi.ua/attachments/093_2016d_Kravchuk.pdf(Дата звернення 21.04.2021);
22. Піддубна Л.П. шляхи розвитку сучасних технологій у документно інформаційному забезпеченні державного управління. Наукові розвідки з державного та муніципального управління, No 1. 2015. <https://is.gd/ПаVsl> .
23. Плагін – що це таке простими словами. URL: <https://ktonanovenkogo.ru/voprosy-i-otvety/chto-takoe-plaginy-skachat-kakustanovit-obnovit.html>
24. Прототип. URL: <https://uk.wikipedia.org/wiki/>
25. Чибісов Ю. В. Застосування методів багатокритеріальної оптимізації Збірник наукових праць Дніпропетровського національного університету залізничного транспорту імені академіка В. Лазаряна. Транспортні системи та технології перевезень. 2014. Вип. 7. С. 65-72.
26. Attribute. URL: <https://whatis.techtarget.com/definition/>
27. Business Process Definition. URL:<https://appian.com/bpm/businessprocess-definition.html>

28. Bootstrap – The most popular HTML, CSS, and JS library in the world.
URL: <https://getbootstrap.com/docs/4.2/getting-started/introduction/>.
29. Getbootstrap. URL: <https://getbootstrap.com/docs/5.2/examples/blog/>
30. Bootstrap. URL: <https://www.bootstrapcdn.com/>
31. Client-Side vs Server-Side Web Development.
URL: <https://careerkarma.com/blog/client-vs-server-side-development>
32. CSS меню. URL: https://puzzleweb.ru/css/15_navbar.php
33. DreamHost Helps You Succeed. URL: <https://www.dreamhost.com/>
(Дата звертання 07.06.2020)
34. Definition and Benefits. IBM case studies. Institute of Robotic Process Automation & Artificial Intelligence website, accessed September 25, 2017. URL: <http://irpaai.com/definition-and-benefits/> (дата звернення: 10.06.2020).
35. IDEF0 - Part 1 (understanding it). URL: http://syque.com/quality_tools/tools/Tools19.htm (дата звернення: 04.05.2021).
36. MySQL to the web. URL: <https://www.phpmyadmin.net/>(Дата звернення 21.05.2021);
37. Steps to Find Your Target Audience. URL: <https://www.marketingevolution.com/marketing-essentials/target-audience>
38. The benefits of using web-based applications. URL: <https://www.geeks.ltd.uk/about-us/blog/details/eQU5Ip/the-benefits-of-using-webbased-applications> .
39. The disadvantages of single page applications. URL: <https://adamsilver.io/articles/the-disadvantages-of-single-page-applications/>.
40. The basics of site architecture: organizing your small business website.
URL: <https://www.namecheap.com/guru-guides/the-basics-of-site-architectureorganizing-your-small-business-website/>.

ДОДАТОК А

SUMMARY

Chorna K.O. Automation of processing orders for online store. – Bachelor's Qualification Thesis. Sumy State University, Sumy, 2020.

The analysis of Vape Shop activity is analyzed, the characteristic of research object is analyzed and given, the state of business process automation is analyzed, requirements to web application are formed, business processes are modeled, information system architecture is created and technologies of solving tasks are defined. designed a web-based information system, developed a prototype web application, described instructions for using the system.

Keywords: automation, online store, web-based system, python, Django, PyCharm.

АНОТАЦІЯ

Чорна К.О. Автоматизація обробки онлайн замовлень інтернет-магазину. – Кваліфікаційна робота бакалавра. Сумський державний університет, Суми, 2022р.

В роботі досліджено аналіз діяльності підприємства «Vape Shop», проаналізовано та надано характеристику об'єкта дослідження, проаналізований стан автоматизації бізнес-процесів, сформовані вимоги до вебзастосунку, змодельовані бізнес-процеси, створена архітектура інформаційної системи та визначено технології вирішення поставлених задач, спроектовано веборієнтовану інформаційну систему, розроблено прототип вебзастосунку, описано інструкції по використанню системи.

Ключові слова: автоматизація, інтернет-магазин, веборієнтована система, python, Django, PyCharm.

ДОДАТОК Б

Файл shop/models.py

```

from django.db import models

class Subscribe(models.Model):
    email = models.EmailField()
    name = models.CharField(max_length=128)
    phone = models.CharField(max_length=18, default='')

    def __str__(self):
        return 'Пользователь %s %s %s' % (self.name,
self.email, self.phone)

    class Meta:
        verbose_name = 'MySubscribe'
        verbose_name_plural = 'A lot of Subscribers'

```

shop/forms.py

```

from django import forms
from .models import *

class SubscribeForm(forms.ModelForm):
    class Meta:
        model = Subscribe
        exclude = [""]

```

shop/admin.py

```

from django import forms
from .models import *

class SubscribeForm(forms.ModelForm):
    class Meta:
        model = Subscribe
        exclude = [""]

```

shop/urls.py

```

from django.urls import path, include
from django.contrib import admin

```

```
from shop import views
```

```
urlpatterns = [
    path('', views.home, name='home'),
    path('home.html', views.home, name='home'),
    path("autorisation.html", views.autorisation,
name='autorisation'),
    path("market.html", views.market, name='market'),
]
```

shop/views.py

```
from django.shortcuts import render
from .forms import SubscribeForm
```

```
def home(request):
    return render(request, 'home.html', locals())
```

```
def autorisation(request):
    form = SubscribeForm(request.POST or None)
    if request.method == 'POST' and form.is_valid():
        print(request.POST)
        print(form.cleaned_data)
        new_form = form.save()
    return render(request, 'autorisation.html',
locals())
```

```
def market(request):
    return render(request, 'market.html', locals())
```

orders/models.py

```
from django.db import models
from products.models import Product
from django.db.models.signals import post_save
```

```
class Status(models.Model):
```

```
    name = models.CharField(max_length=24, blank=True,
null=True, default=None)
    coments = models.TextField( blank=True, null=True,
```

```

default=None)
    is_active = models.BooleanField(default=True)
    created = models.DateTimeField(auto_now_add=True,
auto_now=False)
    update = models.DateTimeField(auto_now_add=False,
auto_now=True)

    def __str__(self):
        return 'Статус%s' % (self.name)

    class Meta:
        verbose_name = 'Статус'
        verbose_name_plural = 'Статусы'

class Order(models.Model):
    total_praise = models.DecimalField(max_digits=10,
decimal_places=2, default=0) # total amount
    customer_email = models.EmailField(blank=True,
null=True, default=None)
    customer_name = models.CharField(max_length=128,
blank=True, null=True, default=None)
    customer_phone = models.CharField(max_length=48,
blank=True, null=True, default=None)
    is_active = models.ForeignKey(Status,
on_delete=models.CASCADE)
    comments = models.TextField(blank=True, null=True,
default=None)
    created = models.DateTimeField(auto_now_add=True,
auto_now=False)
    update = models.DateTimeField(auto_now_add=False,
auto_now=True)
    customer_address = models.CharField(max_length=128,
blank=True, null=True, default=None)

    def __str__(self):
        return 'Заказ %s' % (self.id)

    class Meta:
        verbose_name = 'Заказ'
        verbose_name_plural = 'Заказы'

    def save(self, *args, **kwargs):
        super(Order, self).save(*args, **kwargs)

class ProductInOrder(models.Model):

```

```

    order = models.ForeignKey(Order, blank=True,
null=True, default=None, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, blank=True,
null=True, default=None, on_delete=models.CASCADE)
    nmb = models.IntegerField(default=1)
    prise = models.DecimalField(max_digits=10,
decimal_places=2, default=0)
    total_price = models.DecimalField(max_digits=10,
decimal_places=2, default=0) #prise*nmb
    is_active = models.BooleanField(default=True)
    customer_email = models.EmailField(blank=True,
null=True, default=None)
    created = models.DateTimeField(auto_now_add=True,
auto_now=False)
    update = models.DateTimeField(auto_now_add=False,
auto_now=True)

    def __str__(self):
        return '%s' % (self.product.name)

    class Meta:
        verbose_name = 'Товар'
        verbose_name_plural = 'Товары в заказе'

    def save(self, *args, **kwargs):
        prise_per_item = self.product.prise
        self.prise = prise_per_item
        self.total_price = self.nmb * prise_per_item

        super(ProductInOrder, self).save(*args,
**kwargs)

    def product_post_save(sender, instance, created,
**kwargs):
        order = instance.order
        all_products =
ProductInOrder.objects.filter(order=order,
is_active=True)
        order_total_price = 0
        for item in all_products:
            order_total_price += item.total_price

        instance.order.total_price = order_total_price
        instance.order.save(force_update=True)

```

```
post_save.connect(product_post_save, sender =
ProductInOrder)
```

orders/admin.py

```
from django.contrib import admin
from .models import *

class ProductIOrderInline(admin.TabularInline):
    model = ProductInOrder
```

```
class StatusAdmin(admin.ModelAdmin):
    list_display = [field.name for field in
Status._meta.fields]
    class Meta:
        model = Status
```

```
admin.site.register(Status, StatusAdmin)
```

```
class OrderAdmin(admin.ModelAdmin):
    list_display = [field.name for field in
Order._meta.fields]
    inlines = [ProductIOrderInline]
    class Meta:
        model = Order
```

```
admin.site.register(Order, OrderAdmin)
```

```
class ProductInOrderAdmin(admin.ModelAdmin):
    list_display = [field.name for field in
ProductInOrder._meta.fields]

    class Meta:
        model = ProductInOrder
```

```
admin.site.register(ProductInOrder,
ProductInOrderAdmin)
```

products/models.py

```
from django.db import models
```

```
class Product(models.Model):
    name = models.CharField(max_length=64, blank=True,
```



```

null=True, default=None)
    price = models.DecimalField(max_digits=10,
decimal_places=2, default=0)
    description = models.TextField(blank=True,
null=True, default=None)
    is_active = models.BooleanField(default=True)
    created = models.DateTimeField(auto_now_add=True,
auto_now=False)
    update = models.DateTimeField(auto_now_add=False,
auto_now=True)

    def __str__(self):
        return '%s' % (self.name)

    class Meta:
        verbose_name = 'Товар'
        verbose_name_plural = 'Товары'

class ProductImage(models.Model):
    product = models.ForeignKey(Product, blank=True,
null=True, on_delete=models.CASCADE)
    image =
models.ImageField(upload_to='products_images/')
    is_active = models.BooleanField(default=True)
    created = models.DateTimeField(auto_now_add=True,
auto_now=False)
    update = models.DateTimeField(auto_now_add=False,
auto_now=True)

    def __str__(self):
        return '%s' % (self.id)

    class Meta:
        verbose_name = 'Фото'
        verbose_name_plural = 'Фотографии'

```

products/admin.py

```

from django.contrib import admin
from .models import *

```

```

class ProductImageInline(admin.TabularInline):
    model = ProductImage

class ProductAdmin(admin.ModelAdmin):
    list_display = [field.name for field in
Product._meta.fields]
    inlines = [ProductImageInline]
    class Meta:
        model = Product

admin.site.register(Product, ProductAdmin)

class ProductImageAdmin(admin.ModelAdmin):
    list_display = [field.name for field in
ProductImage._meta.fields]

    class Meta:
        model = ProductImage

admin.site.register(ProductImage, ProductImageAdmin)

```

ДОДАТОК В

autorisation.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    {% load static %}
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/boot
strap.min.css">
    <link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>
<body>
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-
dark" style="opacity: 65%;">
    <div class="container-fluid">

```

```

    <a class="navbar-brand" href="home.html">Головна</a>
    <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse" aria-
controls="navbarCollapse" aria-expanded="false" aria-
label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
    <ul class="navbar-nav me-auto mb-2 mb-md-0">
    <li class="nav-item">
    <a class="nav-link active" aria-current="page"
href="#">Магазин</a>
    </li>
    <li class="nav-item">
    <a class="nav-link" href="#">Про нас</a>
    </li>

    </ul>
    <form class="d-flex" role="search">
    <a class="btn btn-outline-secondary" type="submit"
href="autorisation.html" >Вхід</a>
    </form>
</div>
</div>
</nav>
<header>
    <div class="gradient-border mb-3" >
    <h2 style="color: aliceblue; padding: 30px; opacity:
80%;">Зареєструватися для покупки</h2>
    <div>
    <form action="" method="post" style="color:
aliceblue; padding: 0px 30px 120px;">{% csrf_token %}
    {{ form.as_p }}
    <div style="padding-left: 60px;">
    <button class="btn btn-outline-light"
type="submit">Зареєструватись</button>
    </div>
    </form>
    </div>
</div>

</header>
</body>

```

home.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    {% load static %}
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/boot
strap.min.css">
    <link rel="stylesheet" href="{% static 'css/style.css'%}">

</head>
<body>
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-
dark" style="opacity: 65%;">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">ГОЛОВНА</a>
        <button class="navbar-toggler " type="button" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse" aria-
controls="navbarCollapse" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
            <ul class="navbar-nav me-auto mb-2 mb-md-0">
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page"
href="market.html">Магазин</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Про нас</a>
                </li>
            </ul>
            <form class="d-flex" role="search">
                <a class="btn btn-outline-secondary" type="submit"
href="autorisation.html" >Вхід</a>
            </form>
        </div>
    </div>
</nav>

</div>
<header>
<div class="gradient-border" id="box">
    Chest<br>
    Vape
</div>
</header>

<div class="container">
    <div class="row">

```

```

        <div class="col-12 col-sm-4" id="mainmenu" > <a
href="market.html" style="text-decoration: none; color:white">
        
        <p>Моди</p></a>
    </div>
    <div class="col-12 col-sm-4" id="mainmenu"><a
href="market.html" style="text-decoration: none; color:white">
    
    <p>Подики</p></a>
    </div>
    <div class="col-12 col-sm-4" id="mainmenu"><a
href="market.html" style="text-decoration: none; color:white">
    
    <p>Рідини/Бустери</p></a>
    </div>
</div>
</div>

```

```

<div class="about">
    <div class="logs">
        
    </div>

```

```

    <div class="info">
        <h1>Про нас</h1>

```

Ми співробітничемо з найкращими світовими брендами, такими як:

Eleaf, Joyetech, Kangertech, GeekVape та інші.

У нас представлені найліпші оригінальні веп девайси, ціни на котрі Вас приємно здивують!

В тому числі, в нас величезний вибір рідин для вейпа.

Ассортимент завжди збільшується, і ми раді радувати Вас постійно новими смаками.

```

    </div>

```

```

</div>

```

```

    <div class="centertext">Де ми?</div>

```

```

    <div class="container">

```

```

        <div class="row">

```

```

            <div class="col-md-6">

```

```

                <iframe

```

```

src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d1935.
212603222825!2d34.79741269554079!3d50.91785678192966!2m3!1f0!2f0
!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x4129022f34e74375%3A0xec
1a3824840d7e4d!2sChestVape!5e0!3m2!1sru!2sde!4v1654042308293!5m2
!1sru!2sde" width="100%" height="375" style="border:0;"
allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-

```

```

when-downgrade"></iframe>
    </div>
    <div class="col-md-6 contact">
        <h1> Графік роботи
        <h3>ПН 10:00 - 17:00</h3>
        <h3>ВТ 10:00 - 17:00</h3>
        <h3>СР 10:00 - 17:00</h3>
        <h3>ЧТ 10:00 - 17:00</h3>
        <h3>ПТ 10:00 - 17:00</h3>
        <h3>СБ 10:00 - 17:00</h3>
        <h3>НД 10:00 - 17:00</h3></div>
    </div>
</div>

<div class="centertext1">Тримаємо зв'язок</div>

<div class="container soc" >
    <div class="row">
        <div class="col-12 col-sm-4">
            <button class="button">Telegram</button>
        </div>
        <div class="col-12 col-sm-4" >
            <button class="button">Instagram</button>
        </div>
        <div class="col-12 col-sm-4" >
            <button class="button">Зателефонувати</button>
        </div>
    </div>
</div>
<div class="footer"></div>

</body>
</html>

```

market.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    {% load static %}
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

```

```

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/boot
strap.min.css">
    <link rel="stylesheet" href="{% static 'css/stylesheet.css'
%}">

</head>
<body>
    <header>
        <nav class="navbar navbar-expand-md navbar-dark fixed-top
bg-dark" style="opacity: 65%;">
            <div class="container-fluid">
                <a class="navbar-brand" href="#">Головна</a>
                <button class="navbar-toggler " type="button" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse" aria-
controls="navbarCollapse" aria-expanded="false" aria-
label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarCollapse">
                    <ul class="navbar-nav me-auto mb-2 mb-md-0">
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page"
href="market.html">Магазин</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="#">Про нас</a>
                        </li>
                    </ul>

                    <form class="d-flex" role="search">
                        <a class="btn btn-outline-secondary" type="submit"
href="authorisation.html" >Вхід</a>
                    </form>
                </div>
            </div>
        </nav>

    </header>

    <div class="sidenav">
        <button class="dropdown-btn">Вид
            <i class="fa fa-caret-down"></i>
        </button>
        <div class="dropdown-container">
            <a href="#">-Поди</a>
            <a href="#">-Рідини</a>
            <a href="#">-Моди</a>
            <a href="#">-Бустери</a>
        </div>

```

```

<button class="dropdown-btn">Колір
  <i class="fa fa-caret-down"></i>
</button>
<div class="dropdown-container">
  <a href="#">-Чорний</a>
  <a href="#">-Білий</a>
  <a href="#">-Інший</a>
</div>
<button class="dropdown-btn">Розмір
  <i class="fa fa-caret-down"></i>
</button>
<div class="dropdown-container">
  <a href="#">-10 мл</a>
  <a href="#">-15 мл</a>
  <a href="#">-30 мл</a>
</div>
</div>

<div class="main">

</div>

<script>

var dropdown = document.getElementsByClassName("dropdown-
btn");
var i;

for (i = 0; i < dropdown.length; i++) {
  dropdown[i].addEventListener("click", function() {
    this.classList.toggle("active");
    var dropdownContent = this.nextElementSibling;
    if (dropdownContent.style.display === "block") {
      dropdownContent.style.display = "none";
    } else {
      dropdownContent.style.display = "block";
    }
  });
}
</script>
<footer>
  <div class="container-fluid">
    <div class="row">
      <div class="col-12 col-sm-4" id="leftfoot">
        
        
      </div>
      <div class="col-12 col-sm-4" id="centerfoot">
        chestvapesumy@gmail.com
      </div>
      <div class="col-12 col-sm-4" id="rightfoot">

```



```

65</a>
    <a href="tel:+380661531865">+38 (066) 153-18-
  </div>
</div>
</div>
</footer>
</body>
</html>

```

style.css

```

@import
url('https://fonts.googleapis.com/css?family=Raleway:20
0');
html,body{
    padding: 0;
    margin: 0;
}

html, body {
    height: 100%;
    background-color: black;
}

header {
    width: 100%;
    height: 700px;
    display: flex;
    justify-content: center;
    align-items: center;
    background-image: url("https://i.gifer.com/b6u.gif");
    background-size: cover;
}

#box {
    text-align: center;
    display: flex;
    align-items: center;
    justify-content: center;
    width: 400px;
    height: 200px;
}

```

```
    color: white;
    font-family: 'Raleway';
    font-size: 60px;
    font-weight:bold;
    opacity: 90%;
}
.gradient-border {
    --borderWidth: 3px;
    position: relative;
    border-radius: var(--borderWidth);
    border: 5px solid transparent;
    border-image: linear-gradient(to left top,#BC00FF
10%, transparent 30%, transparent 70%, #BC00FF 90%);
    border-image-slice: 1;
}

.row{
    margin-top: 40px;
}

#mainmenu{
    padding: 10px;
    text-align: center;
    align-items: center;
    outline:2px solid wheat;
    outline-offset: -10px;
    background-color: black;
    font-family: 'Raleway';
    font-weight:bold;
    font-size: 30px;
    color: white;
}

.topimage{
    width: 300px;
    height: 300px;
}

.topimage1{
    padding-top: 10px;
    width: 300px;
    height: 300px;
}

.about{
    margin-top: 47px;
```

```
}

.logs{
  width: 100px;
  margin-left: auto;
  margin-right: auto;
  margin-bottom: -47px;
}
.info{
  background-color: whitesmoke;
  border-radius: 8px;
  padding-left: 10px;
  padding-right: 10px;
  width: 70%;
  margin-left: auto;
  margin-right: auto;
  text-align: center;
  padding-top: 50px;
  padding-bottom: 30px;
}
.logoimg{
  width: 100px;
  height: 100px;
}

.centertext{
  margin-top: 50px;
  font-family: 'Raleway';
  font-size: 60px;
  font-weight:bold;
  color: white;
  width: 100%;
  text-align: center;
  margin-left: auto;
  margin-right: auto;
}
.glmmap{
  display: flex;
  flex-direction: row;
  order: 2;
  width: 80%;
  margin-left: auto;
  margin-right: auto;
```

```
    border: 2px solid red;
}

.contact{
    color: white;
    text-align: center;
    align-items: center;
}

.contact h1{
    padding-bottom: 15px;
}

.centertext1{
    margin-top: 50px;
    font-family: 'Raleway';
    font-size: 60px;
    font-weight:bold;
    color: white;
    width: 100%;
    text-align: center;
}

.soc{

    width: 100%;
    align-items: center;
    text-align: center;
}

.wrap {
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
}

.button {
    min-width: 300px;
    min-height: 60px;
    font-family: 'Nunito', sans-serif;
    font-size: 22px;
    text-transform: uppercase;
```

```
    letter-spacing: 1.3px;
    font-weight: 700;
    color: #313133;
    background: #4FD1C5;
background: linear-gradient(90deg,
rgba(129,230,217,1) 0%, rgba(79,209,197,1) 100%);
    border: none;
    border-radius: 1000px;
    box-shadow: 12px 24px rgba(79,209,197,.64);
    transition: all 0.3s ease-in-out 0s;
    cursor: pointer;
    outline: none;
    position: relative;
    padding: 10px;
    margin-top: 20px;
}
```

```
.button::before {
content: '';
border-radius: 1000px;
min-width: calc(300px + 12px);
min-height: calc(60px + 12px);
border: 6px solid #00FFCB;
box-shadow: 0 0 60px rgba(0,255,203,.64);
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
opacity: 0;
transition: all .3s ease-in-out 0s;
}
```

```
.button:hover, .button:focus {
color: #313133;
transform: translateY(-6px);
}
```

```
.button:hover::before, .button:focus::before {
opacity: 1;
}
```

```
.button::after {
content: '';
width: 30px; height: 30px;
border-radius: 100%;
}
```

```

border: 6px solid #00FFCB;
position: absolute;
z-index: -1;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
animation: ring 1.5s infinite;
}

.button:hover::after, .button:focus::after {
  animation: none;
  display: none;
}

@keyframes ring {
  0% {
    width: 30px;
    height: 30px;
    opacity: 1;
  }
  100% {
    width: 300px;
    height: 300px;
    opacity: 0;
  }
}

.footer{
  width: 100%;
  height: 150px;
}

```

styleshop.css

```

@import
url('https://fonts.googleapis.com/css?family=Raleway:20
0');
html,body{
  padding: 0;
  margin: 0;
}

```

```
body{
    background-color: black;

    background-image:
url("/static/image/bodysmoke.jpeg");
    background-size: cover;
    background-attachment: fixed;
}

.sidenav {
    min-height: 40%;
    width: 150px;
    position: fixed;
    z-index: 1;
    top: 0;
    left: 0;
    background-color: rgb(0, 0, 0, 0.7);
    overflow-x: hidden;
    padding-top: 20px;
    margin-top: 60px;
    border-top-right-radius: 20px;
    border-bottom-right-radius: 20px;
    font-family: 'Raleway';
    font-weight: bold;
}

.sidenav a, .dropdown-btn {
    padding: 6px 8px 6px 16px;
    text-decoration: none;
    font-size: 20px;
    color: white;
    display: block;
    border: none;
    background: none;
    width: 100%;
    text-align: left;
    cursor: pointer;
    outline: none;
```

```
}

.sidenav a:hover, .dropdown-btn:hover {
    color: #f1f1f1;
}

.main {

    margin-left: 150px;
    font-size: 20px;
    padding: 0px 10px;
    height: 100%;
    margin-top: 10px;
}

.active {
    background-color: rgb(0, 0, 0, 0.6);
    color: white;
}

.dropdown-container {
    display: none;
    background-color: rgb(0, 0, 0, 0.6);
    padding-left: 8px;
}

.fa-caret-down {
    float: right;
    padding-right: 8px;
}

@media screen and (max-height: 450px) {
    .sidenav {
        padding-top: 15px;
    }
    .sidenav a {font-size: 18px;}
}
```



```
header{
  width: 100%;
  height: 50px;
  background-color: rgb(0, 0, 0, 0.6);
}

footer{
  width: 100%;
  height: 60px;
  background-color: rgb(0, 0, 0, 0.9);
  position: absolute;
  left: 0;
  bottom: 0;
}

.logoicon{
  width: 40px;
  height: 40px;
  margin-left: 2%;
}

.logoicon1{
  width: 35.5px;
  height: 35.5px;
  margin-left: 2%;
}

#leftfoot{
  text-align: left;
  height: 60px;
  line-height: 60px;
}

#centerfoot{
  text-align: center;
  color: white;
  font-family: 'Raleway';
  font-weight: bold;
  font-size: 20px;
  height: 60px;
  line-height: 60px;
}

.rightfoot a{
text-decoration: none;
```

```
color: white;

}

.rightfoot{
    font-family: 'Raleway';
    font-weight: bold;
    font-size: 20px;
    text-align: right;
    height: 60px;
    line-height: 60px;

}
```