

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

_____ червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційне та програмне забезпечення інтернет-магазину одягу»
здобувача групи ІН – 94-1 Чигрина, Артема, Альбертовича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

_____ Артем Чигрин
(підпис)

Керівник,
професор, доктор технічних наук

Микола Будник

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-94-1 Чигрина, Артема, Альбертовича

1. Тема роботи: «Інформаційне та програмне забезпечення інтернет-магазину одягу»
затверджую наказом по СумДУ від «01» червня 2023 р. № 0475-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовуються під час розробки сайтів та адміністративних панелей до них. 3) Розробка інтернет-магазину одягу. 4) Аналіз отриманих результатів
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____ (підпис)

Керівник _____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів кваліфікаційної роботи | Термін виконання | Примітка |
|-------|---|-------------------|----------|
| 1 | <i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i> | 10.01.23-05.02.23 | |
| 2 | <i>Огляд технологій, що використовуються під час розробки сайтів та адміністративних панелей до них</i> | 06.01.23-16.02.23 | |
| 3 | <i>Розробка інформаційного та програмного забезпечення інтернет-магазину одягу</i> | 17.02.23-15.03.23 | |
| 4 | <i>Аналіз отриманих результатів</i> | 16.03.23-20.04.23 | |
| 5 | <i>Оформлення пояснювальної записки до кваліфікаційної роботи</i> | 21.04.23-25.05.23 | |

Здобувач вищої освіти _____ (підпис)

Керівник _____ (підпис)

АНОТАЦІЯ

Записка: 69 стр., 22 рис., 1 додаток, 11 використаних джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки у сучасному світі кожен бренд має свій сайт, особливо коли це бренд націлений на продаж великої кількості товару і створення інтернет-магазинів в цьому допомагають.

Об'єкт дослідження — створення інтернет-магазину та адміністративної панелі для керування сайтом.

Мета роботи — розробка інтернет-магазину та додатку для керування та спостереження за вмістом сайту

Методи дослідження — методи написання сайтів, методи та типи створення адміністративних панелей.

Результати — Розроблено сайт інтернет-магазину одягу та додаток для керування вмістом сайту, а саме товарами та замовленнями.

ІНТЕРНЕТ-МАГАЗИН, АДМІНІСТРАТИВНА ПАНЕЛЬ, САЙТ, PYTHON, SQL,
PHP, HTML, JAVASCRIPT.

ЗМІСТ

| | |
|---|----|
| Вступ..... | 5 |
| 1 ІНФОРМАЦІЙНИЙ ОГЛЯД..... | 7 |
| 1.1 Принципи побудови інтернет-магазину..... | 7 |
| 1.2 Структура інтернет-магазину..... | 7 |
| 1.3 Типи інтернет-магазинів..... | 9 |
| 1.4 Способи написання сайту..... | 9 |
| 1.5 Функції адміністративної панелі..... | 10 |
| 1.6 Типи адміністративних панелей..... | 11 |
| 1.7 Висновки..... | 12 |
| 1.2 Постановка задачі..... | 13 |
| 2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ..... | 14 |
| 2.1 Опис задачі для розробки сайту..... | 14 |
| 2.2 Прототипування інтернет-магазину..... | 14 |
| 2.3 Інструменти для розробки інтернет-магазину..... | 15 |
| 2.4 Розробка дизайну інтернет-магазину..... | 16 |
| 2.5 Опис задачі для додатку..... | 17 |
| 2.6 Прототипування додатку..... | 18 |
| 2.7 Інструменти для розробки додатку..... | 18 |
| 2.8 Розробка дизайну додатку..... | 19 |
| 3 ПРОГРАМНА РЕАЛІЗАЦІЯ..... | 21 |
| 3.1 Архітектура сайту та додатку..... | 21 |
| 3.2 Таблиці даних..... | 22 |
| 3.3 Створення сайту..... | 27 |
| 3.4 Створення додатку..... | 30 |
| ВИСНОВКИ..... | 36 |
| СПИСОК ЛІТЕРАТУРИ..... | 37 |
| ДОДАТОК А ПРОГРАМНИЙ КОД..... | 39 |

ВСТУП

У наш час все більше людей говорять що вони віддають перевагу інтернет покупкам, а не звичайному походу у магазин. За останні роки процес покупки товару дуже змінився, і все це через інтернет магазини. Покупці порівнюють товари, дивляться на їх характеристики та мають майже необмежений час на те щоб вирішити що саме вони хочуть купити.

Багато людей у всьому світі вважають за краще робити покупки в Інтернеті та купувати товари від різних брендів і компаній, які також можуть бути недоступні для покупки в звичайних магазинах у їх країні. У наш час, завдяки новій технології та підтримці Інтернету, люди з усього світу почали купувати товари онлайн, просто сидячи вдома. Тож деякі компанії взагалі відмовилися від продажу товару через точки з товарами, на користь онлайн магазинів. Купівля предметів і продуктів через інтернет є дуже простим завданням. Зараз вона відіграє дуже важливу роль у житті кожного, особливо людей похилого віку, а також людей із дуже напруженим графіком життя, деякі люди просто фізично не можуть дістатися магазинів, або не мають на це достатньо часу. У час коли люди в основному зберігають гроші на кредитній картці, дебетовій картці чи мають рахунок в інтернет-банку, дуже легко купувати саме онлайн. Прямий онлайн-продаж збільшує ваше охоплення. Завдяки інтернет-магазину ваші прибутки більше не обмежуються кількістю клієнтів, які можуть фізично відвідати ваш звичайний заклад. Ви навіть можете продавати за кордоном, маючи таким чином усю планету в якості потенційних покупців. Ваш онлайн-магазин також дозволяє обслуговувати покупців, яким зручніше переглядати та купувати в той час, коли дрібні точки традиційно не працюють

У той самий час, інтернет-магазин потребує регулювання, для цього найкраще вього підійде додаток, який буде виступати у ролі адміністративної панелі. У світі де більшість сфер життя вже цифровізовані, розробка додатків для керування базами даних є надзвичайно актуальною. Такий додаток як

найменше полегшить роботу для працівників магазину, їм не потрібно буде розбиратися як влаштовані БД, як їми користуватися та як працюють SQL запити. Все що потрібно щоб керувати базою даних магазину це форми та таблиці які керуються за допомогою кнопок. Адмін-панелі є у кожного комерційного сайту, які потребують постійного оновлення асортименту. Адмін-панель може бути як вбудованою, доступ до якої буде через посилання, так і у вигляді додатка, що є більш безпечним бо доступ до неї буде більш обмеженим. Конкретно, додаток для керування БД магазину по продажу одягу, який має функції видалення, додавання товару, може додавати та видаляти користувачів та переглядати товари в магазині. Це стає невід'ємною частиною роботи продавців та менеджерів магазину. Такий додаток дозволяє зберігати важливі дані про товари та користувачів у централізованій базі даних, без навичок програмування, що забезпечує зручний та швидкий доступ до інформації, а також ефективне управління магазином. Усе більша кількість компаній виходять на цифровий ринок, тому використання додатків для керування базами даних є необхідним для підвищення конкурентоздатності та забезпечення ефективної роботи підприємства.

Метою розробки додатку для керування базою даних магазину по продажу одягу є створення зручного та ефективного інструменту для управління товарами та персоналом, що пов'язані з діяльністю магазину. Розроблений додаток дозволить забезпечити швидкий та зручний доступ до важливої інформації про товари та замовлення

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Принципи побудови інтернет-магазину

Зайшовши до магазину, користувач бачить список товарів, що є на складі. Оскільки онлайнний «прилавок» зазвичай прив'язаний до системи автоматизації будь-якого підприємства, цей список містить ті ж вироби, що є у продажу й у звичайних (не віртуальних) магазинах. Вміст складу представляється зазвичай як ієрархічної деревоподібної структури, базовими елементами якої є групи товарів.

Існують різноманітні форми оплати: за готівковий розрахунок кур'єру при доставці, за безготівковим розрахунком (для організацій), банківським переказом, електронні платежі, оплата післяплатою або поштовим і телеграфним переказом.

Існують такі способи доставки замовлення: кур'єрськими службами, звичайною поштою, або авіапоштою Росією, країнам ближнього та далекого зарубіжжя, самовивозом та інші способи доставки.

Існують інтернет магазини які націлені на продаж якогось одного виду товару, але з різними типами і користувач повинен мати змогу шукати товар по характеристикам або типам.

Інтернет магазин повинен бути інтуїтивно зрозумілим та зовнішньо привабливим щоб користувач не вийшов лише через поганий дизайн або занадто не зрозуміле меню.

Ці принципи повинні дотримуватися для побудови успішного інтернет магазину

1.2 Структура інтернет-магазину

Структура сайту повинна бути максимально інтуїтивною для користувача, щоб він завжди розумів на якій сторінці знаходиться та не загубився у складній архітектурі. При розробці структури інтернет-магазину краще всього опиратися на те щоб поділити її на 3 основних критерія, а саме

головна сторінка, категорії та продукт. Різні сторінки можна відділити друг від друга за допомогою змін у дизайні та меню, це зробить сайт більш інтуїтивним.

Починаючи с головної сторінки користувач повинен спочатку переходити до окремих типів товару, а вже потім до їх категорій, у разі якщо сайт є віртуальним супермаркетом (рисунок 1.1), або одразу до категорій, якщо це спеціалізований магазин (рисунок 1.2).

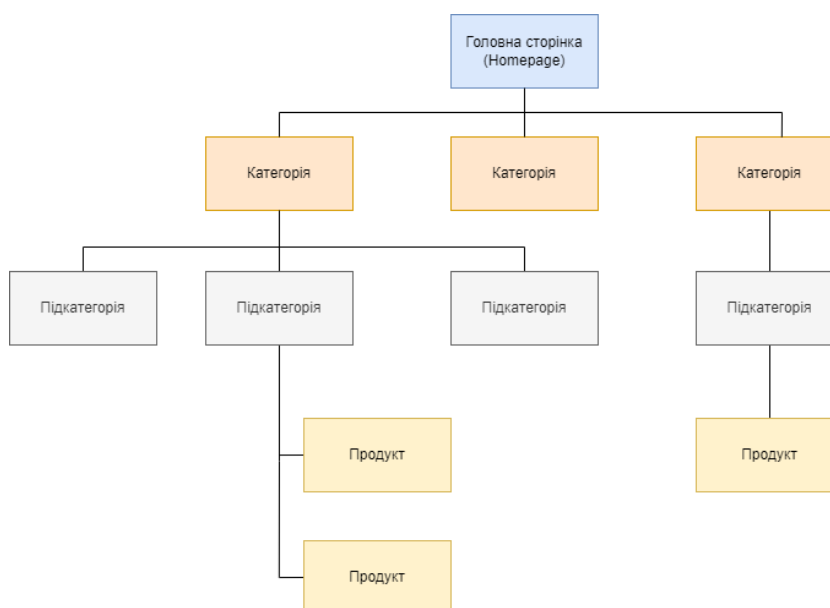


Рисунок 1.1 – Структура віртуального супермаркету

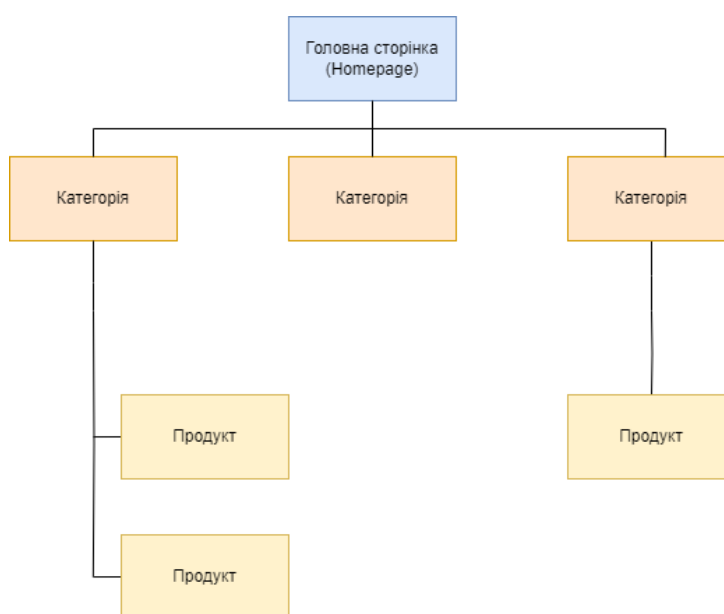


Рисунок 1.2 – Структура спеціалізованого магазину

При цьому на сайті може бути ще декілька допоміжних сторінок по типу підтримки, опису бренду та іншого.

1.3 Типи інтернет-магазинів

Інтернет-магазини частіше за все ділять на декілька типів у різних категоріях, а саме вони можуть відрзнятися за типом товару, по типу продажу товару, по асиртменту товару або навіть по географії продажу.

Якщо ділити інтернет-магазин за типом товару то можна визначити два чітких типи, а саме:

інтернет-магазину які продають будь-який товар, це називають віртуальними супермаркетами і часто може бути що у бренду, який володіє інтернет-магазином немає особистого товару, а сам товар є нішевим.

Спеціалізовані магазини. Такі інтернет-магазину продають конкретний тип товару, частіше за все товар конкретного бренду і тільки його.

1.4 Способи написання сайту

Існує 2 найпопулярніших способи написання сайту, вони відрізняються тим наскільки глибокі знання у програмуванні потрібно задіяти. Самий легкий спосіб написати сайт це написати його за допомогою конструкторів. Сайти-конструктори є простими і доступними засобами для створення веб-сайтів без необхідності вміння програмувати. Основна ідея полягає в тому, що вони надають вам готові шаблони та інструменти для налаштування та створення вашого веб-сайту. Але за легкістю і зручністю є дуже великий мінус, який полягає в їх обмеженій гнучкості та універсальності. Вони пропонують шаблони та налаштування, що не завжди можуть відповідати унікальним потребам замовника. Якщо є специфічні вимоги щодо дизайну, функціональності або інтеграції з іншими системами, то може стати складніше досягти бажаного результату. На фоні цього самостійна верстка має кілька переваг над сайтами-конструкторами. Незважаючи на складність виконання, у

такому способі написання присутня повна гнучкість на універсальність. При самостійній верстці розробник має повний контроль над кожним аспектом вашого веб-сайту. Можна створити унікальний дизайн, враховуючи всі потреби та вимоги. Налаштувати функціональність, розташування елементів, анімацію та взаємодію з користувачем за уподобаннями замовника. Немає обмеження шаблонами або налаштуваннями, наданими платформою.

1.5 Функції адміністративної панелі

Адмін-панель для керування сайтом дозволяє:

- Додавати та видаляти нові товари на сайті;
- Додавати нових користувачів;
- Переглядати замовлення;
- Сортувати дані за характеристиками.

Адмін-панель має полегшити керування вебсайтом та задовільняти усі функції, які потрібні для коректної роботи сайту.

Основні види адмін-панелей:

1.6 Типи адміністративних панелей

Вбудована адмін-панель - це інструмент, який вбудований безпосередньо в код веб-сайту і дозволяє адміністраторам змінювати та керувати контентом веб-сайту, не виходячи зі сторінки. Однією з переваг вбудованих адмін-панелей є їх простота та доступність, як у створенні так і у використанні. Їх створення та налаштування може бути простим і швидким процесом, що не вимагає серйозних знань та навичок програмування. Вбудована адмін-панель може також бути чудовим рішенням для невеликих веб-сайтів, що не потребують складних функцій керування контентом. Графічний інтерфейс для даного типу адмін-панелей потрібен не завжди, так як основна взаємодія проходить за допомогою форм html.

Недоліки такого підходу – вбудовані адмін-панелі можуть бути менш гнучкими та обмеженими у функціональності порівняно з іншими типами адмін-панелей. Крім того, вбудовані адмін-панелі можуть мати вплив на продуктивність сайту, якщо їх функції недостатньо оптимізовані. Також, через те що такі адмін-панелі є частиною сайту то вони не завжди захищені та безпечні, навіть незважаючи на наявність авторизації.

Для розробки такого типу адмін-панелей не потрібні навички в прогармуванні поза межами веб-програмування, так як вбудована адмін-панель прив'язана до сайту то і інструменти для її створення використовуються також самі як і для написання основної частини сайту, а саме HTML, CSS, JavaScript та PHP, або будь-яка інша мова для взаємодії з БД.

Власна адмін-панель - це рішення, що розробляється з нуля. Як правило такі адмін-панелі призначені для одного сайту та створюються саме під нього, щоб задовільнити максимальну кількість потреб користувачів(персоналу). Їх вміст чітко обговорюється та задається в ТЗ. Однією з переваг власних адмін-панелей є те, що їх можна налаштувати під конкретні потреби сайту, додавши необхідні функції та відключивши зайві. Крім того, такі адмін-панелі можуть бути більш простими та інтуїтивно зрозумілими для користувачів, що сприяє більш швидкому та ефективному керуванню сайтом. Саме така адмін-панель була розроблена під час практики. Так як мова програмування для створення таких додатків може бути майже будь-якою це відкриває широкий спектр можливостей для редагування та опрацювання навіть найменших особливостей.

Недоліки такого підходу – створення власної адмін-панелі складніше за створення вбудованої, через це потребуються більш серйозні навички в програмуванні та більше часу на виконання завдання. Також не виключено і проблеми з безпекою, хоча в теорії їх може бути менше ніж у вбудованій панелі, проте це залежить від того як буде налаштована програма та її структура.

Для розробки власної адмін-панелі потрібні знання веб-розробки та програмування, створення графічного інтерфейсу

Готові адмін-панелі - це програмне забезпечення, яке вже розроблене та готове до використання. Їх можна знайти в мережі Інтернет та завантажити, щоб встановити на свій веб-сайт. Як правило функціонал в них доволі широкий, але не гнучкий і не підійде для конкретних задач через неможливість змінити структуру адмін-панелі щоб підлаштувати її для усіх функцій сайту. З переваг можна виділити швидкість встановлення та підключення до сайту. Також такі адмін-панелі або зовсім не потребують навичок в програмуванні, або потребують лише поверхневих навичок.

Недоліки такого підходу – неможливість підтримки або оновлення деяких функцій через те що код писався не під конкретний сайт.

1.7 Висновки

Інтернет-магазин повинен бути інтуїтивно зрозумілим та виглядати приємно, усі основні елементи меню мають бути зрозумілими та розташованими на самих видних місцях, щоб користувач не заплутався у структурі сайту та постійно розумів на якому етапі він знаходиться. Краще за все використовувати самостійну верстку та створювати сайт з нуля, це допоможе досягти бажаних результатів та змінювати любий аспект сайту у подальшому.

Адмін-панель у свою чергу має спростити зв'язок між користувачем, а саме працівником магазину, та системою, надаючи всі необхідні функції для того щоб керувати сайтом без навичок у програмуванні. Зараз на ринку існує 3 найпоширеніших види адмін-панелей, серед яких вбудована адмін-панель, власна адмін-панель та вже готові варіанти адмін-панелей. Найбільш оптимальним варіантом є саме створення власного додатку, який буде працювати окремо від сайту. У такому разі це забезпечить гнучкість у розробці та повний контроль над фінальним продуктом з можливістю змінити будь-

яку з функції у подальшому. При належному виконанні такі види адмін-панелі будуть значно безпечніші та захищеніші від несанкціонованого доступу ніж інші. Хоч це і потребує глибших знань у програмуванні та більшого часу на розробку, такі додатки надають куди більше можливостей та мають кращий графічний інтерфейс.

1.8 Постановка задачі

Після проведення інформаційного огляду було вирішено розробити приклад спеціалізованого інтернет-магазину та додатку для керування. При цьому основним завданням є:

- 1) Вибрати інструменти потрібні для реалізації проекту;
- 2) Розробити дизайн сайту
- 3) виконати проектування архітектури веб-сайту та бази даних;
- 4) Розробити анімації для сайту;
- 5) Виконати тестування веб-сайту
- 6) Розробити дизайн додатку
- 7) Проектування додатку
- 8) Тестування додатку

У разі успішної реалізації буде можливим перегляд товарів різних категорій, перегляд характеристик товарів та заповнення замовлення, а також додавання та видалення нових товарів, моніторинг замовлень та керування персоналом.

2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

2.1 Опис задачі для розробки сайту

Інтернет-магазин повинен бути максимально інформативним, але при цьому не перевантажений інформацією. З ключевих деталей, які повинні бути реалізовані можна виділити:

- Можливість перегляду товарів по типу
- Наявність основних характеристик та опису товару
- Наявність зображень товару
- Наявність кошика замовлень
- Можливість залишити коментарі к товарам

Головною сторінкою інтернет-магазину буде виступати сторінка з товарами через які можна буде переходити до різних типів товару, а у подальшому й до самих товарів та сторінки замовлення(кошику).

2.2 Прототипування інтернет-магазину

Структурно сайт поділено на 3 основних сторінки, які є самими необхідними для коректної роботи інтернет-магазину.

Головна сторінка – містить блоки, які заповнені різними типами товарів та підвантажуються в залежності від того який товар тип товару вибрав користувач серед існуючих, а саме футболки, головні убори, низ, взуття та інші аксесуари.

Сторінка товару – містить усю необхідну інформацію про товар, його зображення, коментарі до товару та блок з вибором необхідного розміру та кількості товару для замовлення.

Кошик – містить два основних блока. У першому блоку користувач заповнює усі необхідні дані для замовлення товару, серед яких ім'я, адреса, телефон та інше. У другому блоку розташований кошик з товарами які користувач до нього додав. Після заповнення усіх даних та відправки форми, замовлення потрапляє до бази даних де його обробляє персонал магазину.

2.3 Інструменти для розробки інтернет-магазину

Так як для розробки сайту найкраще підійде метод самостійної верстки і створення всіх елементів з нуля то для виконання цієї задачі знадобиться декілька мов програмування, а саме HTML, PHP, SQL та JavaScript.

HTML (HyperText Markup Language) - це основна мова розмітки для створення веб-сторінок. Вона визначає структуру та семантику контенту на веб-сторінці, дозволяючи визначити заголовки, абзаци, списки, посилання, зображення та багато іншого. HTML використовує теги для визначення кожного елемента і їхньої функціональності.

PHP - це мова програмування, яка використовується для розробки динамічних веб-сайтів та веб-додатків. Вона працює на стороні сервера, що означає, що PHP-скрипти виконуються на веб-сервері, а результат відправляється на клієнтський браузер у вигляді HTML-сторінки. PHP дозволяє взаємодіяти з базами даних, обробляти форми, генерувати динамічний контент, керувати сесіями користувачів, робити запити до інших серверів і багато іншого.

SQL (Structured Query Language) - це стандартна мова запитів до баз даних, що використовується для збереження, оновлення та видалення даних в реляційних базах даних. SQL використовується для створення, модифікації та управління базами даних, зберігання та вибірки даних, агрегації та фільтрації даних, створення таблиць та інших об'єктів баз даних. SQL є однією з найбільш популярних мов для роботи з базами даних, і використовується у багатьох веб-додатках та програмних засобах. SQL дозволяє зручно та ефективно управляти великими обсягами даних та здійснювати складні запити до баз даних.

JavaScript - це високорівнева мова програмування, що використовується для розробки динамічних веб-сторінок та веб-додатків. Вона працює на стороні клієнта, тобто виконується безпосередньо в браузері користувача. JavaScript надає можливість додавати взаємодію, анімацію та динаміку до веб-

сторінок. З його допомогою можна маніпулювати елементами сторінки, реагувати на події, які відбуваються в браузері (натискання кнопок, рух миші тощо), отримувати та відправляти дані на сервер, виконувати асинхронні запити до сервера без перезавантаження сторінки (AJAX), створювати складні веб-додатки та багато іншого.

2.4 Розробка дизайну інтернет-магазину

Для сайту було обрано мінімалістичний стиль, який складається з трьох основних кольорів, а саме білого, червоного та чорного (рисунок 2.1 – 2.2). Білий колір використовуються як фон для кожної сторінки сайту, червоним позначені важливі елементи сторінки, частіше за все це ті елементи з якими можна взаємодіяти, на чорному кольорі розташована важлива інформація, але тільки для перегляду. Такий дизайн було розроблено з метою щоб користувач легко орієнтувався на сайті та не губився серед великого потоку візуальної інформації, у той самий час звертаючи увагу на яскраві елементи меню.

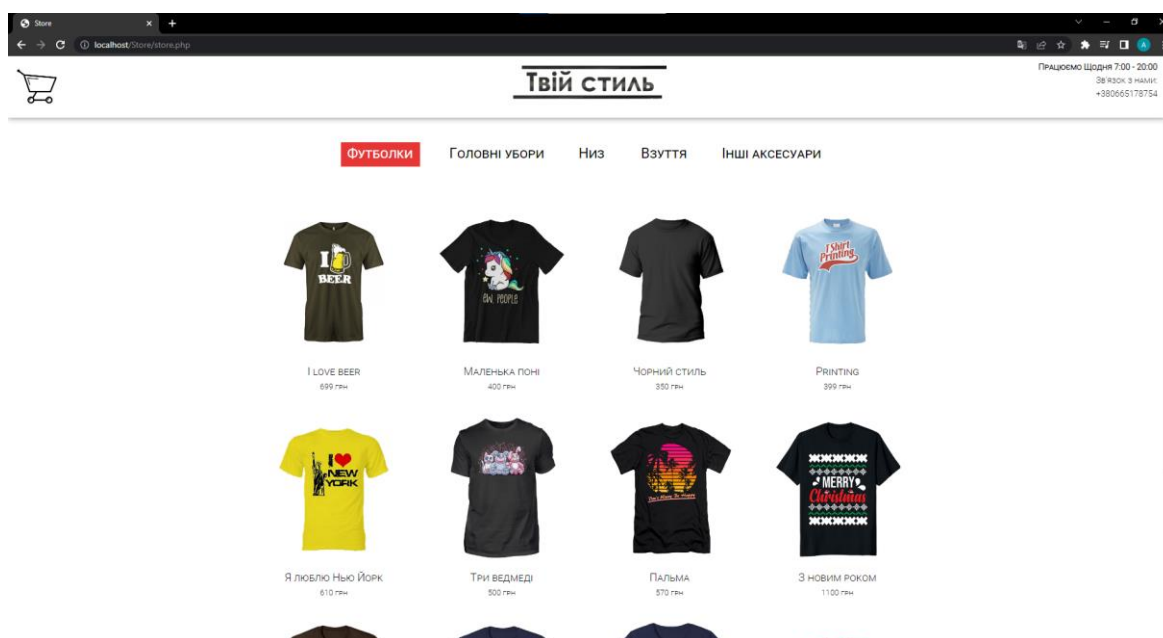


Рисунок 2.1 – Головна сторінка інтернет-магазину

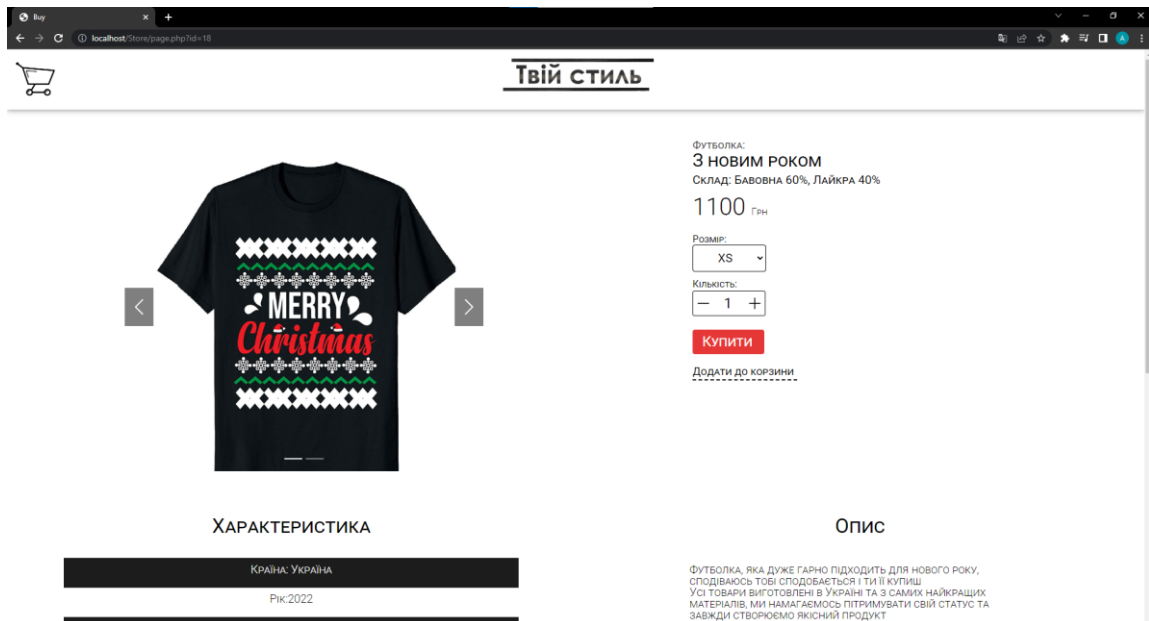


Рисунок 2.2 – Сторінка товару

2.5 Опис задачі для розробки додатку

При розробці макету додатку були визначені наступні критерії, які повинні бути у фінальній версії:

- Можливість додавати та видаляти товари
- Можливість додавати нових користувачів
- Можливість переглядати існуючі замовлення
- Наявність графічного інтерфейсу
- Сортування даних

Після встановлення додатку він вже матиме доступ до БД веб-сайту, а саме інтернет магазину одягу, який знаходиться на сервері. Так як усі функції призначені саме для конкретного сайту, додаток не має можливості підключення до іншого, так як це не матиме сенсу. Для того щоб користуватися адміністративною панеллю потрібно щоб користувача внесли до бази даних персоналу, це може зробити лише головний акаунт адміна. Після реєстрації всього персоналу, кожен зареєстрований юзер може зайти до додатку де йому будуть доступні головні функції для керування БД.

2.6 Прототипування додатку

Інтерфейс додатку містить лише необхідні функції та не відрізняється незалежно від зміни вікон.

Вікно реєстрації – містить поля необхідні для заповнення username(логін) та password(пароль) та кнопку для того щоб зайти до акаунту. Після того як користувач натисне на кнопку “login” функція перевірить персонал, який вже внесений до БД, якщо усі дані співпадуть то користувач отримає доступ до основної частини адміністративної панелі. У разі коли логін або пароль були введені помилково додаток повідомить про це

Вікно головної панелі – містить декілька блоків серед яких блок додавання товару, блок з таблицею товарів та блок керування. Блок додавання товарів містить усі необхідні поля для заповнення інформації про товари. Блок з таблицею товарів виводить на екран вміст бази даних. Блок керування може сортувати дані по критеріям, переходити до інших вікон, видаляти товари та виконати вихід з акаунту.

Вікно для керування персоналом – доступне лише головному акаунту та має можливість додавати або видаляти персонал, який має доступ до додатку

Допоміжні вікна – мають функції повідомити користувача про помилки або зміни у БД перед або після внесення даних. Викликаються при використанні деяких функцій.

Вікно замовлень – містить список у якій з бази даних виводяться замовлення які вже були помічені як виконані та замовлення які очікують на виконання.

2.7 Інструменти для розробки додатку

При розробці додатку використовувалася мова програмування Python та бібліотеки, а саме pymysql, необхідний для підключення до бази даних та customtkinter, за допомогою якого був створений графічний інтерфейс. Також застосовувався SQL.

Python - це високорівнева, інтерпретована, загального призначення мова програмування. Python має багату екосистему, включаючи бібліотеки для різних задач та фреймворки для веб-розробки, наукових обчислень, штучного інтелекту та інших областей. Вона має простий, зрозумілий синтаксис та не потребує багато зусиль для написання простих програм. Python також підтримує багато парадигм програмування, включаючи процедурне, об'єктно-орієнтоване та функціональне програмування.

PyMySQL – одна з найбільш популярних бібліотек для з'єднання з MySQL базами даних за допомогою мови програмування Python. Ця бібліотека дозволяє легко та швидко з'єднуватися з MySQL сервером та взаємодіяти з ним за допомогою Python. PyMySQL підтримує стандартний SQL, такий як SELECT, INSERT, UPDATE та DELETE, а також інші операції, такі як створення таблиць, індексів та зв'язків між ними. Бібліотека також має підтримку транзакцій, які дозволяють гарантувати цілісність даних при виконанні операцій.

Customtkinter - це бібліотека для мови програмування Python, яка дозволяє створювати власні теми та стилі для віджетів Tkinter. Ця бібліотека дозволяє змінювати зовнішній вигляд вбудованих в Tkinter віджетів, що дозволяє створювати привабливі та сучасні інтерфейси.

При написанні коду був використаний програмний додаток Pycharm
PyCharm — це кросплатформове інтегроване середовище розробки мови програмування Python, розроблене компанією JetBrains[5] на основі IntelliJ IDEA. Надає користувачеві комплекс засобів для написання коду та візуальний відладчик.

2.8 Розробка дизайну додатку

Для розробки інтерфейсу додатку був обраний мінімалістичний дизайн з темним, іноді темно-сірим фоном та зеленими частинами інтерфейсу(рисунок. 2.3), таблиця виводу даних з бази даних має білий фон

(рисунок 2.4). Деякі з елементів інтерфейсу відрізняються та мають іншу кольорову схему, для того щоб позначити їх важливість або перехід до іншої частини додатку

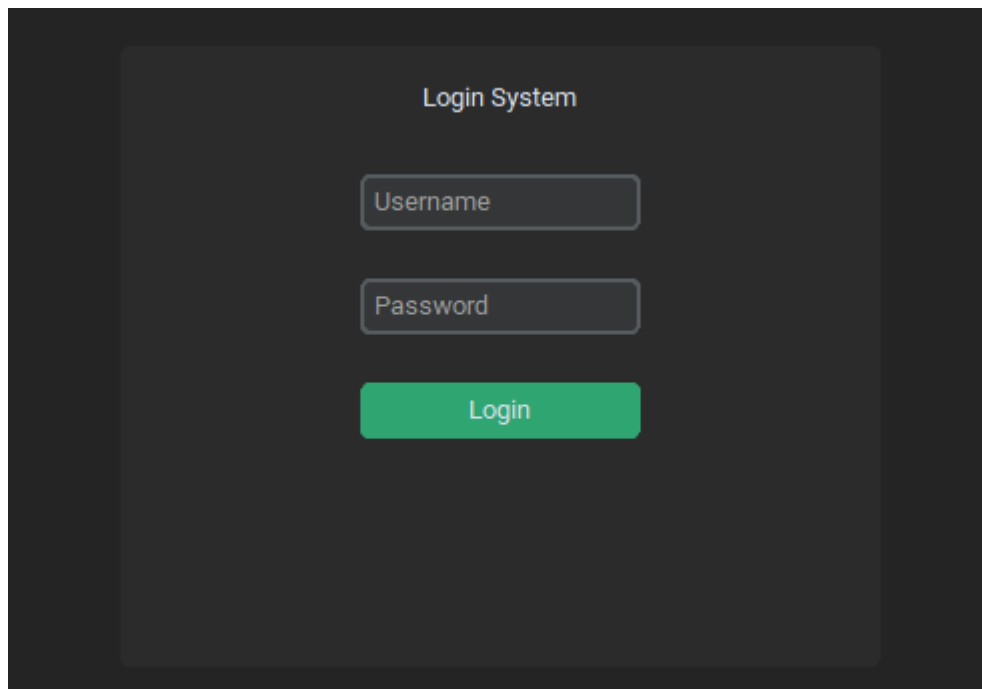


Рисунок 2.3 – Вікно логіну

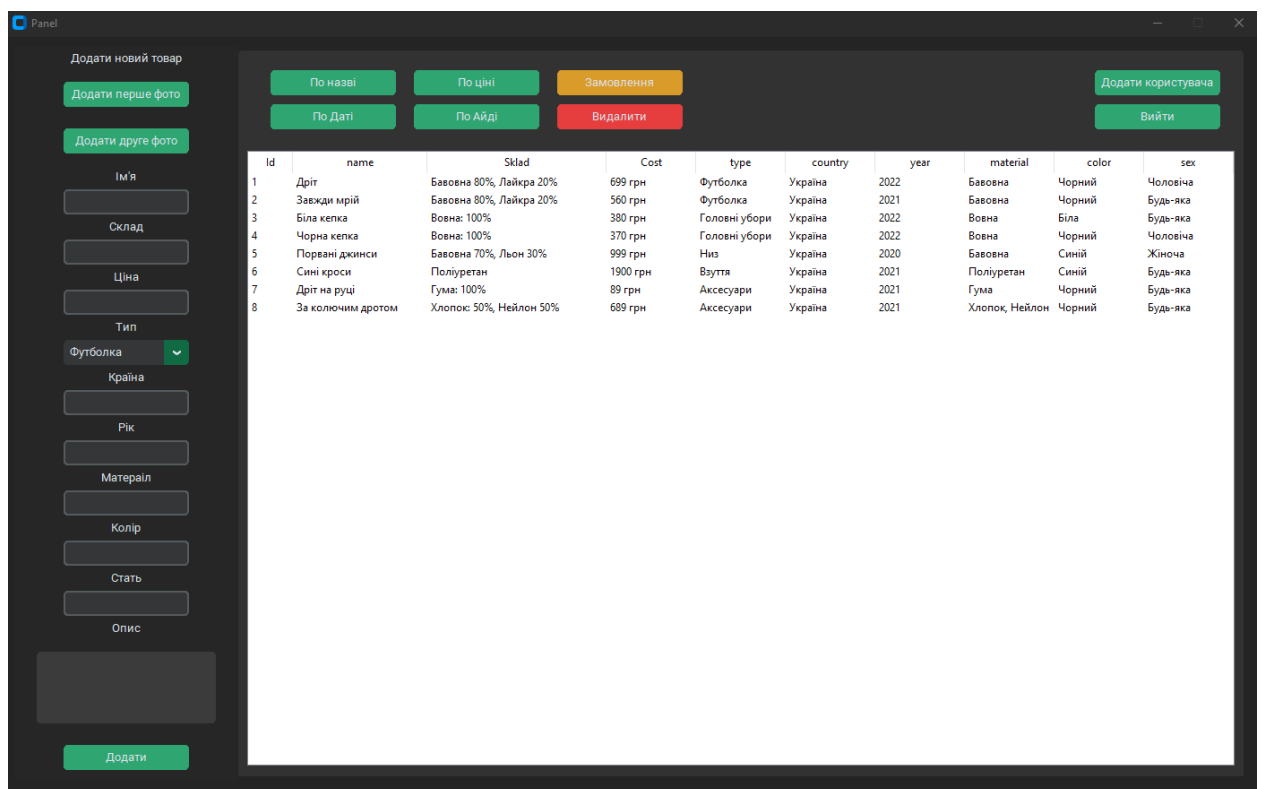


Рисунок 2.4 – Головне вікно адмін-панелі

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Архітектура сайту та додатку

Для реалізації сайту було створено 3 основних сторінки. Головна сторінка, для вибору товару серед декількох категорій. За допомогою функцій JavaScript було розроблено спосіб виведення різних категорій товару при натисканні на відповідну кнопку у верхньому меню. Друга сторінка - детальний опис товару та його характеристик з наведеними фото, які можна перегортати за допомогою слайдеру. Третя сторінка відповідає за заповнення інформації користувачем для подальшого оформлення товару та кошиком з товарами, до якого можна перейти у будь-який час з головної сторінки та додавати до нього товари. Архітектура інтернет-магазину зображена на рисунку 3.1

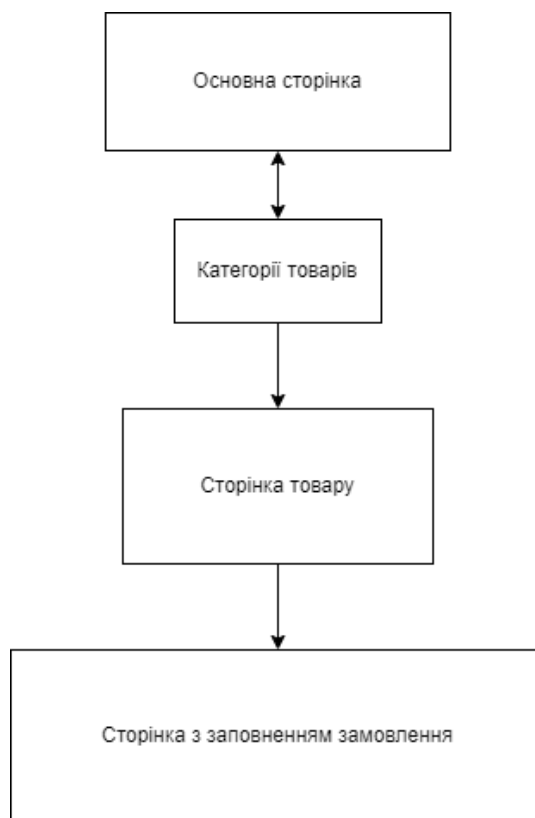


Рисунок 3.1 – Архітектура інтернет-магазину

Для створення додатку була обрана структура з переходом між необхідними вікнами. Для переходу між різними частинами додатку було

створено декілька вікон, які виконують різні функції та поповнені необхідним контентом, Головними вікнами виступають саме вікно логіну та вікно адмін-панелі, при їх закритті додаток перестав працювати, закривається. Під час роботи з вікном адмін-панелі, вікно логіну приховується, це зроблено для того щоб не навантажувати систему неактивними частинами додатку. Архітектура самого додатку доволі проста та централізована у головному вікні адмін-панелі (рисунок 3.1)

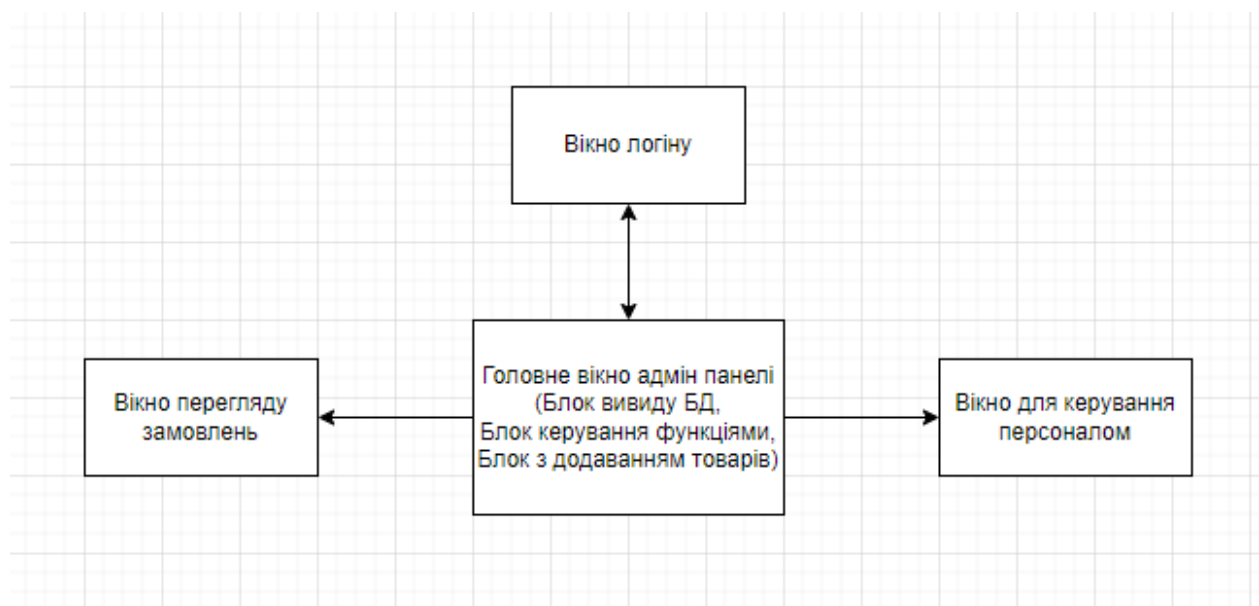


Рисунок 3.2 – Архітектура додатку

Головною метою було створити інтерфейс який буде зрозумілий людині, яка не пов'язана з програмуванням, а сам інтерфейс буде сучасним та приємним, усі його функції інтуїтивно зрозумілі.

3.2 Таблиці даних

Для реалізації сайту та додатку необхідно щоб у базі даних були такі таблиці як таблиця з товарами, таблиця з персоналом, таблиця з коментарями та таблиця з замовленнями. Для цього, за допомогою SQL, було створено необхідні таблиці(таблиця 3.1-3.4).

Таблиця 3.1 – tovar

| Поле | Зміст | Тип | Обмеження | Ключі |
|----------|--------------------------------|---------|-----------|-------|
| ID | Id товару | int | Not null | PK |
| Image1 | Перше зображення товару | varchar | Not null | |
| Image2 | Друге зображення товару | varchar | | |
| Name | Назва товару | varchar | | |
| Sklad | Склад товару | varchar | | |
| Cost | Ціна товару | int | Not null | |
| type | Тип товару, категорія | varchar | Not null | |
| Country | Країна виробник товару | varchar | | |
| year | Рік створення товару | int | | |
| material | Головний матеріал товару | varchar | | |
| Color | Колір товару | varchar | | |
| sex | Стать для якої призначен товар | varchar | | |

| Поле | Зміст | Тип | Обмеження | Ключі |
|-------|-------------|---------|-----------|-------|
| Descr | Опис товару | varchar | | |

Таблиця 3.2 – users

| Поле | Зміст | Тип | Обмеження | Ключі |
|----------|--------------------|---------|-----------|-------|
| ID | Id користувача | int | Not null | PK |
| Name | Логін користувача | varchar | Not null | |
| Password | Пароль користувача | varchar | Not null | |

Після створення таблиці users було додано одного користувача з ім'ям admin, це і буде головний акаунт. Так як у програмі виключений варіант з повторною реєстрацією одного і того ж юзера, так як присутня перевірка при додаванні нових користувачів, то у самому коду головному акаунт видано більші права і можливість видаляти вже існуючих користувачів

Частина коду:

```
if (check_adm == "admin"):
    button_del2 = customtkinter.CTkButton(master=frame12, text="Видалити
користувача", fg_color="#e63e3e", hover_color="#e82525")
    button_del2.pack(pady=12, padx=10)
```


Таблиця 3.3 – order1

| Поле | Зміст | Тип | Обмеження | Ключі |
|--------------|-------------------------------|---------|-----------|-------|
| id | Номер товару | int | Not null | PK |
| data1 | Дата замовлення | date | Not null | |
| Name | Ім'я та прізвище замовника | varchar | Not null | |
| Place_to | Місце доставки | varchar | Not null | |
| Tel | Телефон замовника | int | Not null | |
| Mail | Пошта замовника | varchar | | |
| Ord | Список товарів | int | Not null | |
| Order_status | Статус замовлення | varchar | Not null | |

У таблицю orders через запит у PHP кодї сайту потрапляють дані клієнта, які він вводить при замовленні товару (рисунок 3.2), а саме:

- id: номер товару, який збільшується автоматично при кожному замовленні перевіряючи останній id замовлення, після чого вноситься до бд
- Data1: поточна дата, яка береться з системи
- Name: ім'я та прізвище замовника пишеться разом, на основі двох полів при замовленні
- Place_to: місце доставки, яке вводить замовник

- Tel: номер телефону який вводить замовник
- Mail: пошта яку вводить замовник. Є необов'язковим полем
- Ord: замовлення, яке враховую кількість товару, їх назву, номер та розмір.
- Order_status: статус замовлення, потрібен щоб персонал, який буде розглядати замовлення зрозумів виконано воно, чи ще потребує обробки. Після того як замовлення сформовано у таблицю вводиться статус "Очікує", після виконання замовлення таблицю оновлюється і у статусі замовлення пишеться "Виконано(Ім'я виконавця)", ім'я виконавця залежить від того з якого акаунту було виконане замовлення.



Твій стиль

ДЕТАЛІ ОПЛАТИ

Ім'я*
ВВЕДІТЬ ІМ'Я

Прізвище*
ВВЕДІТЬ ПРІЗВИЩЕ

Місто*(Тільки Україна)
ВВЕДІТЬ ВАШЕ МІСТО

Адреса*
ВВЕДІТЬ ВАШУ АДРЕСУ

Телефон*
ВВЕДІТЬ ВАШ ТЕЛЕФОН

EMAIL
ВВЕДІТЬ ВАШУ ЕЛЕКТРОННУ АДРЕСУ

ВАШЕ ЗАМОВЛЕННЯ

ФУТБОЛКА
ЗАВЖДИ МРІЙ х2 1120 грн
РОЗМІР: XL

ОПЛАТА:

- ГОТІВКОЮ
- КАРТКОЮ
- ЧЕК

ОПЛАТА ПРОВІДИТЬСЯ НА МІСЦІ, ПІСЛЯ ОТРИМАННЯ
ЗАМОВЛЕННЯ ВІД КУР'ЄРА

Підтвердити замовлення

TvuyStil.com@gmail.com
Твій Стиль®
2022

Рисунок 3.3 – Сторінка оформлення замовлення

Таблиця 3.4 – comments1

| Поле | Зміст | Тип | Обмеження | Ключі |
|----------|------------------|---------|-----------|-------|
| id | Номер товару | int | Not null | PK |
| Name_log | Ім'я користувача | varchar | Not null | |

| Поле | Зміст | Тип | Обмеження | Ключі |
|----------|----------|-----|-----------|-------|
| Comm_log | коментар | | Not null | |

Дані потрапляють у таблицю за допомогою SQL запиту у PHP-кодї сторінки редіректу під час відправки форми.

- id: номер товару, який береться за допомогою GET у PHP під час відправки форми користувачем
- name_log: ім'я яке користувач вводить у форму
- comm_log: коментарь користувача

3.3 Створення сайту

Для реалізації сторінок було обрано блочну структуру написання коду. Головна сторінка сайту поділена на 3 основних блока, а саме header, main_div та bottom. Усі товари виводяться до основного блоку сайту, який містить ще 5 схованих блоків, кожен з яких відображається після того як користувач вибере необхідну категорію товарів для його відображення (рисунок 3.4).



Рисунок 3.4 – Категорії товарів

Товари виводяться з бази даних за їх номером за допомогою функції прописаній у PHP кодї.

На сторінці товару у main_div розсташовано вже 2 основних блоки, а саме блок з товаром та коментарями.

У блоці з товаром опимані його головні характеристики та присутня форма за допомогою якої можна обрати кількість товару та його розмір, після чого одразу оформити замовлення або додати твар до кошику (рисунок 3.5)

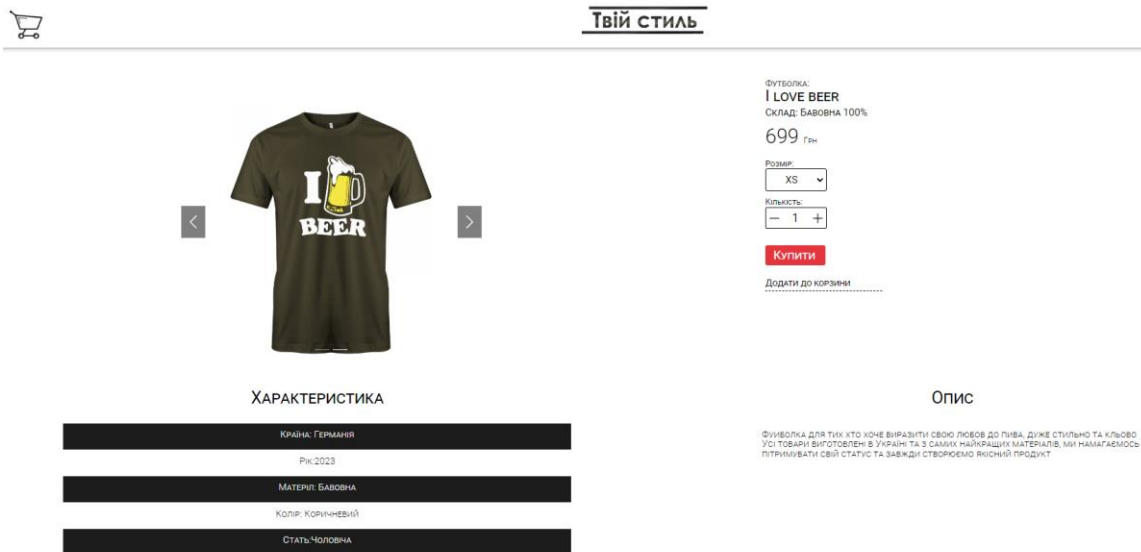


Рисунок 3.5 – Сторінка з товаром

Під описом товару знаходиться секція з коментарями, де користувач може залишити відгук про товар та переглянути вже існуючі коментарі (рисунок 3.6)

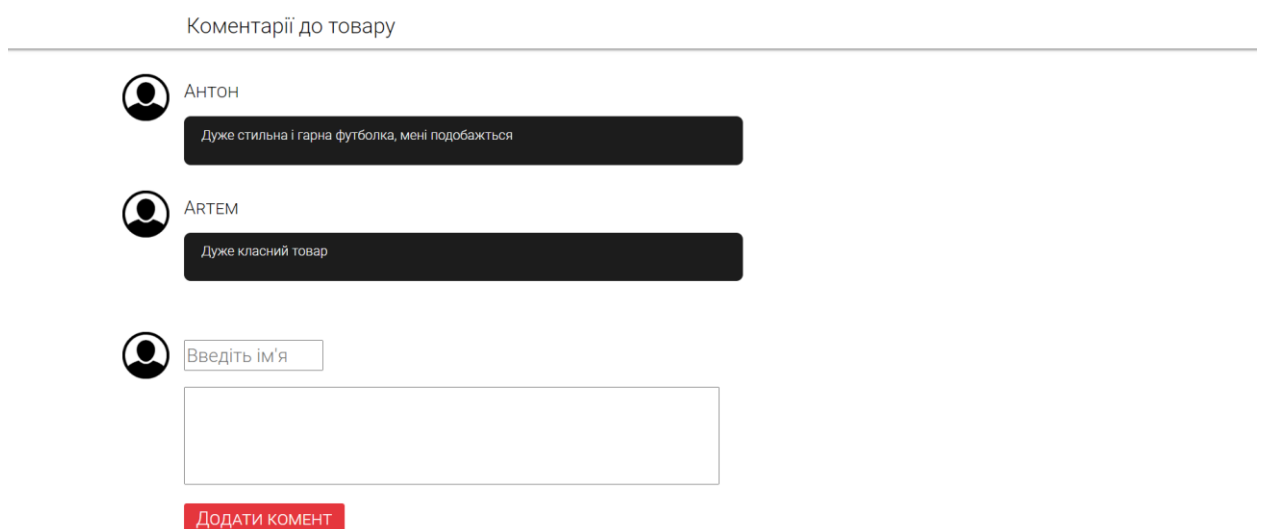


Рисунок 3.6 – Коментарі до товару

Після відправки форми всі введені дані, а також іd товару, який закріплений за сторінкою товару, потрапляють до бази даних

Частина коду:

```
if (strlen($_GET["entername_comm9"])>0 and strlen($_GET["comment"])>0){
    $sql79 = "INSERT INTO comments1(id, name_log, comm_log) VALUES
    ('".$_GET['id']."','".$_GET["entername_comm9"]."',
    '".$_GET["comment"]."');"
    $result79 = $conn->query($sql79);}
```

Після того як користувач додасть до кошику необхідні йому товари він може перейти до сторінки оформлення, де від нього потребується заповнити необхідні для доставки дані. Також він може видалити з кошику непотрібні товари та переглянути суму замовлення (рисунок 3.7).

Твій стиль

| | | | | | | | |
|---|--|--|---------|--|---------|-------------------------|--|
| <p>ДЕТАЛІ ОПЛАТИ</p> <p>Ім'я* <input style="width: 90%;" type="text" value="ВВЕДІТЬ ІМ'Я"/></p> <p>Прізвище* <input style="width: 90%;" type="text" value="ВВЕДІТЬ ПРИЗВИЩЕ"/></p> <p>Місто*(Тільки Україна) <input style="width: 90%;" type="text" value="ВВЕДІТЬ ВАШЕ МІСТО"/></p> <p>Адреса* <input style="width: 90%;" type="text" value="ВВЕДІТЬ ВАШУ АДРЕСУ"/></p> <p>Телефон* <input style="width: 90%;" type="text" value="ВВЕДІТЬ ВАШ ТЕЛЕФОН"/></p> <p>ЕМАІЛ <input style="width: 90%;" type="text" value="ВВЕДІТЬ ВАШУ ЕЛЕКТРОННУ АДРЕСУ"/></p> | <p>ВАШЕ ЗАМОВЛЕННЯ</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="font-size: small;">ФУТБОЛКА ЗАВЖДИ МРІЙ x1 Розмір:XS ВИДАЛИТИ З КОШИКА</td> <td style="text-align: right; vertical-align: top;">560 грн</td> </tr> <tr> <td style="font-size: small;">ФУТБОЛКА I LOVE BEER x1 Розмір:XS ВИДАЛИТИ З КОШИКА</td> <td style="text-align: right; vertical-align: top;">699 грн</td> </tr> <tr> <td colspan="2">Усього: 1259 грн</td> </tr> </table> <p>Оплата:</p> <p><input type="radio"/> Готівкою</p> <p><input type="radio"/> КАРТКОЮ</p> <p><input type="radio"/> ЧЕК</p> <p style="font-size: x-small; margin-top: 5px;">Оплата проводиться на місці, після отримання замовлення від кур'єра</p> <p style="text-align: center; margin-top: 10px;">Підтвердити замовлення</p> | ФУТБОЛКА ЗАВЖДИ МРІЙ x1 Розмір:XS ВИДАЛИТИ З КОШИКА | 560 грн | ФУТБОЛКА I LOVE BEER x1 Розмір:XS ВИДАЛИТИ З КОШИКА | 699 грн | Усього: 1259 грн | |
| ФУТБОЛКА ЗАВЖДИ МРІЙ x1 Розмір:XS ВИДАЛИТИ З КОШИКА | 560 грн | | | | | | |
| ФУТБОЛКА I LOVE BEER x1 Розмір:XS ВИДАЛИТИ З КОШИКА | 699 грн | | | | | | |
| Усього: 1259 грн | | | | | | | |

Рисунок 3.7 – Кошик

Після оформлення товару користувач потрапляє на сторінку де виводиться повідомлення про те що замовлення знаходиться в обробці (рисунок 3.8)

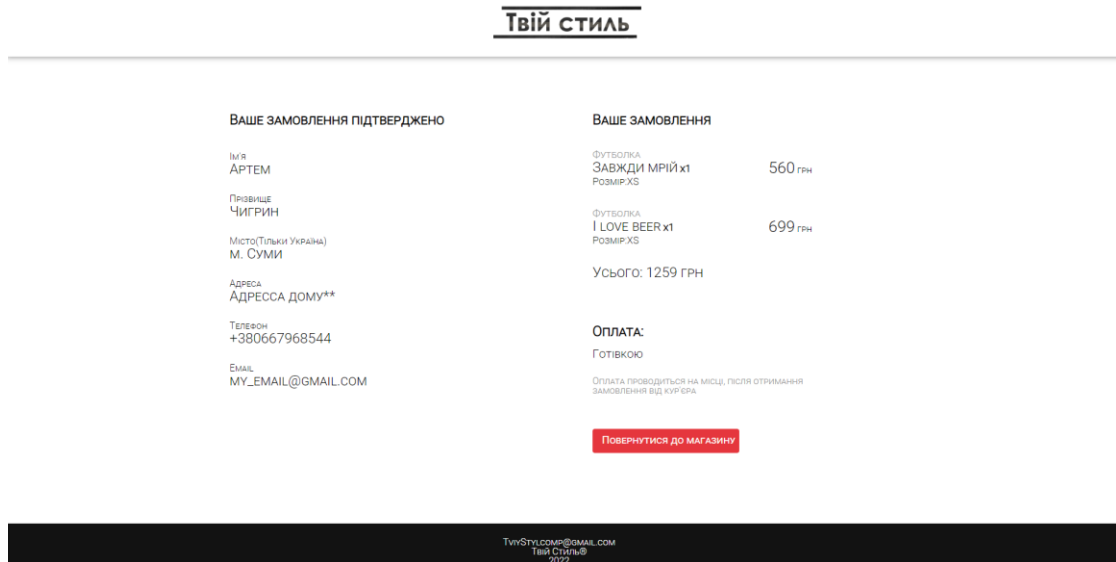


Рисунок 3.8 – Сторінка підтвердження замовлення

Коли замовлення оформлено воно потрапляє до списку замовлень у бд, а вже з нього виводиться і до додатку (рисунок 3.9)

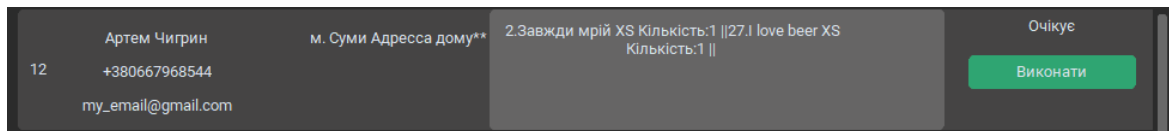


Рисунок 3.8 – Замовлення у додатку

3.4 Створення додатку

Програма поділена на декілька вікон, кожне вікно має свою назву та певну функцію у кодї, яка викликається при переходї до вікна та може містити інші функції.

Вікно логіну містить просту форму, яка знаходиться у фреймі та вміщує у себе label(назву форми), два entry поля для вводу необхідних даних та кнопку login для того щоб запустити функцію перевірки на наявність користувача у базї даних (рисунок 3.10)

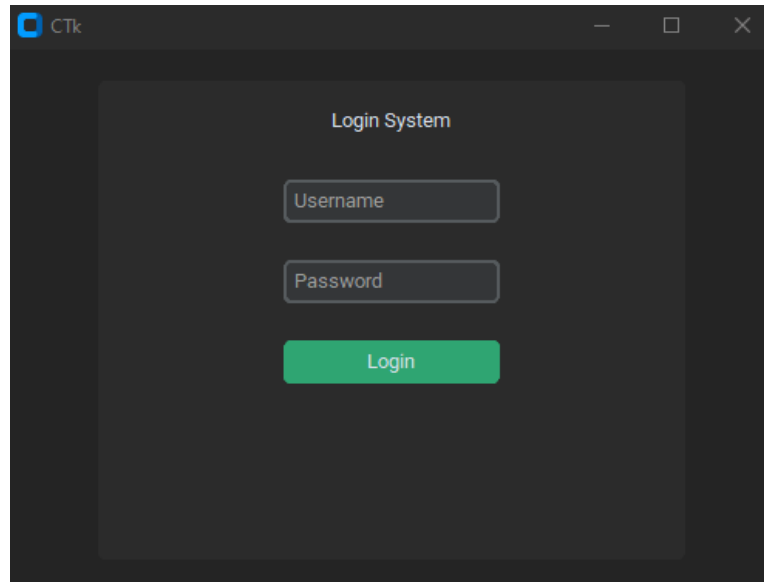


Рисунок 3.10 – Логін до адмін-панелі

Частина коду:

```
def allusers():
    with connection.cursor() as cursor:
        select_all = "SELECT * FROM users"
        cursor.execute(select_all)
        rows = cursor.fetchall()
        return rows
for row in conn.allusers():
    if (str(log1) == row[1] and str(pass1) == row[2]):
```

Якщо дані не співпадуть з жодним з полів у базі даних то програма повідомить користувачу про помилковість введених даних

Головне вікно поділено на 3 основні блоки, які виконують різні функції. Блок введення даних до БД містить поля необхідні для того щоб додати новий товар на сайт (рисунок 3.11)

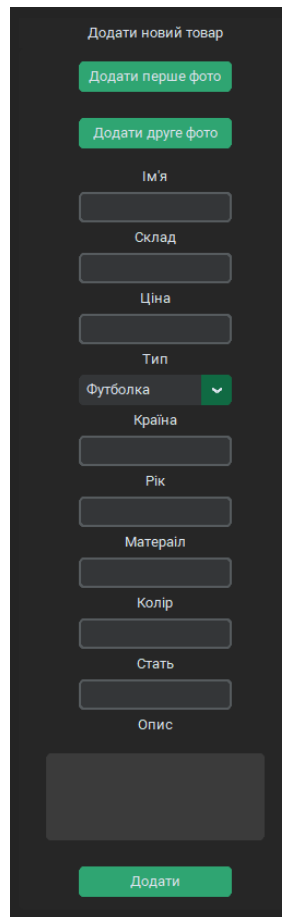


Рисунок 3.11 – Блок введення даних

Після того як користувач заповнить дані, а фото для товару вибере з файлів на комп'ютері, введені дані пройдуть перевірку у кодї, на правильність типу даних, таким чином у поле “ціна” можна вводити лише числовий тип даних, а у поле “країна” лише символи. У випадку коли користувач невірно ввів типи даних або забув заповнити поле, додаток повідомить про це, вказавши яка саме помилка була допущена (рисунок 3.12-рисунок 3.13)

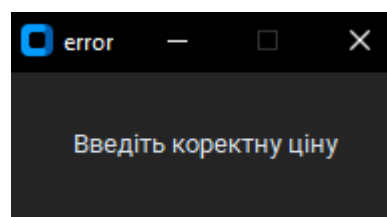


Рисунок 3.12 – помилка не вірно введеної ціни

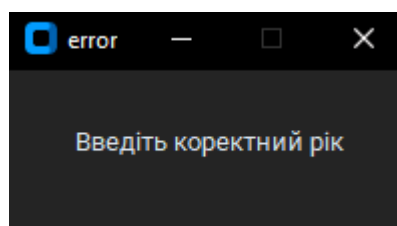


Рисунок 3.13 – помилка не вірно введеного року

Після того як користувач коректно введе усі дані, додаток додасть їх у базу даних та новий товар з'явиться у магазину

Блок таблиці даних містить у собі таблицю, яка була створена за допомогою функції Treeview у бібліотеці tkinter, та представляє собою вивід усіх товарів з БД (рисунок 3.14). Також є можливість перейти до сторінки товару через таблицю, для цього потрібно просто клацнути на потрібний товар, це відкриє браузер з посиланням на сторінку товару

| Id | name | Sklad | Cost | type | country | year | material | color | sex |
|----|------------------|-------------------------|----------|---------------|---------|------|----------------|--------|----------|
| 6 | Сині кроси | Поліуретан | 1900 грн | Взуття | Україна | 2021 | Поліуретан | Синій | Будь-яка |
| 5 | Порвані джинси | Бавовна 70%, Льон 30% | 999 грн | Низ | Україна | 2020 | Бавовна | Синій | Жіноча |
| 1 | Дріт | Бавовна 80%, Лайкра 20% | 699 грн | Футболка | Україна | 2022 | Бавовна | Чорний | Чоловіча |
| 8 | За ключим дротом | Хлопок: 50%, Нейлон 50% | 689 грн | Акcesуари | Україна | 2021 | Хлопок, Нейлон | Чорний | Будь-яка |
| 2 | Завжди мрій | Бавовна 80%, Лайкра 20% | 560 грн | Футболка | Україна | 2021 | Бавовна | Чорний | Будь-яка |
| 3 | Біла кепка | Вовна: 100% | 380 грн | Головні убори | Україна | 2022 | Вовна | Біла | Будь-яка |
| 4 | Чорна кепка | Вовна: 100% | 370 грн | Головні убори | Україна | 2022 | Вовна | Чорний | Чоловіча |
| 7 | Дріт на руці | Гума: 100% | 89 грн | Акcesуари | Україна | 2021 | Гума | Чорний | Будь-яка |

Рисунок 3.14 – таблиця товарів

Блок керування містить у собі кнопки, які потрібні для того щоб сортувати дані у таблиці за критеріями, а також кнопку замовлень, яка відкриває вікно з замовленнями, кнопку видалення товарів, яка видалає обрані товари з бази даних, після того як уточнить це, кнопку “додати користувача”, яка відкриває нове вікно та кнопку виходу з акаунту (рисунок 3.15)



Рисунок 3.15 – блок керування

Вікно замовлень містить у собі список замовлень з усіми необхідними даними щоб оператор, який їх перевіряє міг уточнити усе та перевести замовлення у статус “виконано”. Після переведення замовлення у статус “виконано” рядом з цим також написано ім’я оператора, який відповів на замовлення. (рисунок 3.16)

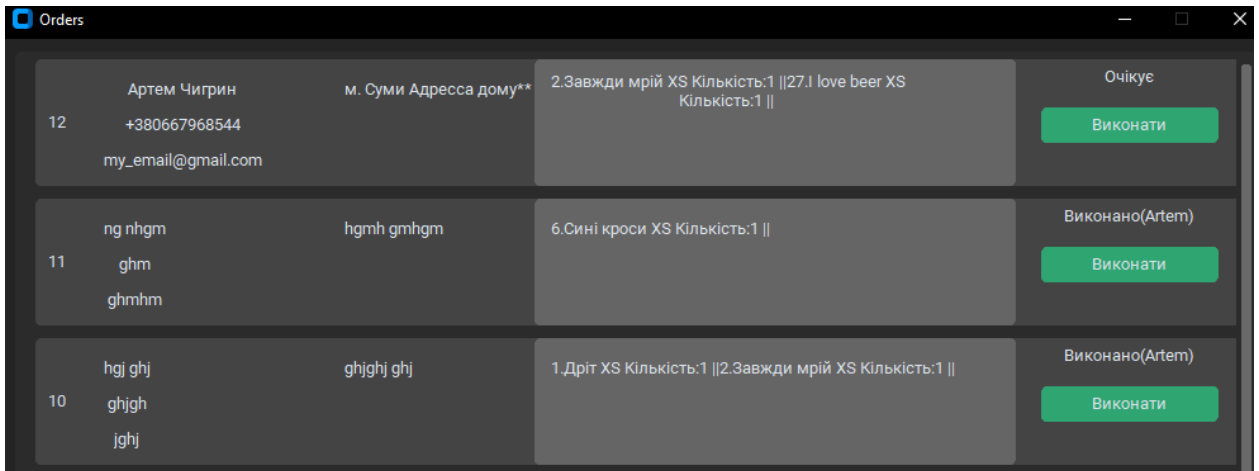


Рисунок 3.16 – вікно замовлень

Вікно додавання користувачів має у собі форму, як при реєстрації, але з уточненням введеного паролю та функцію видалення користувача, яка доступна лише адміну. Після введених даних, система перевіряє наявність користувача в базі даних, при додаванні вже існуючого користувача система повідомить про те що він вже є у базі даних (рисунок 3.17)

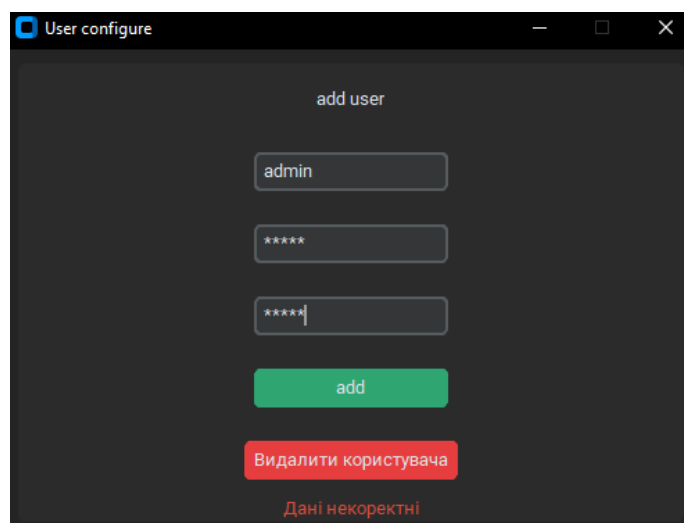


Рисунок 3.17 – вікно керування персоналом

Допоміжні вікна та вікна помилок створені для того щоб користувач розумів коли щось іде не так та що до цього призвело або у ситуаціях коли з базою даних мають статися незворотні речі, наприклад видалення існуючого товару (рисунок 3.18)

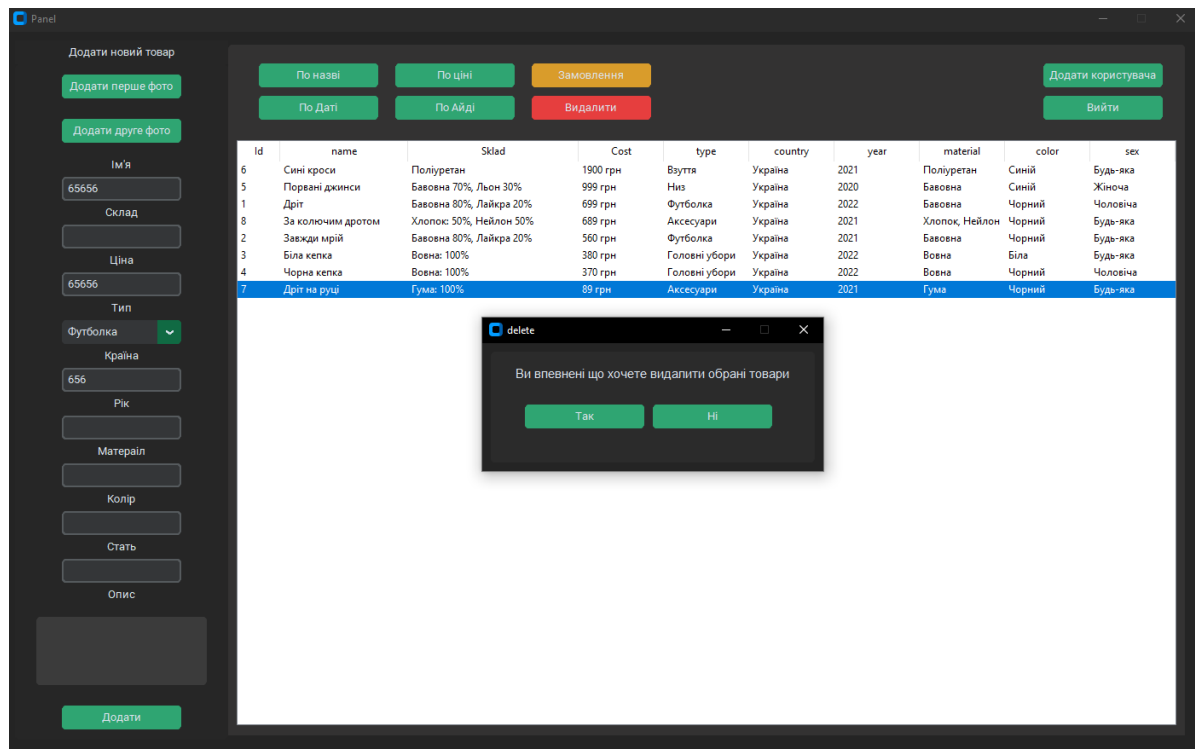


Рисунок 3.18 – вікно уточнення

Майже кожне вікно має функції для отримання або введення даних до БД, це відбувається за допомогою SQL запитів у самому кодї програми

Частина коду:

```
def delets(selected):
    srt = ""
    for i in selected:
        srt = srt+str(i)+", "
    with connection.cursor() as cursor:
        insr_q = "DELETE FROM tovar WHERE id IN ({})".format(srt[:-1])
        cursor.execute(insr_q)
        connection.commit()
```

ВИСНОВКИ

У ході виконання бакалаврської роботи було розроблено сайт інтернет-магазину та додаток для керування БД сайту. Як результат я ознайомився з великою кількістю інтернет-магазинів та вивчив основні вимоги для їх коректної роботи. Створив інтернет магазин для невеликого бренду одягу, використовуючи такі мови як JavaScript, CSS та HTML для клієнтської частини і PHP та SQL для серверної. Після літературного огляду згідно теми практики я прийшов до висновку що з 3-х основних видів адмін-панелей які існують зараз на ринку, власна адміністративна панель, розроблена з нуля, є найкращим варіантом через великий спектр можливостей додавання нових функцій та зміни графічного інтерфейсу під потреби замовника. І хоч такий варіант є найскладнішим, фінальний продукт виходить значно більш функціональним та гнучким з точки зору оновлення та підтримки додатку.

Після вибора підходу до рішення задачі була розроблена просто блок-схема майбутнього додатку, яка передбачає наявність декількох вікон з різними функціями, для взаємодії з базою даних, був розроблений дизайн додатку та його кольорова схема, яка передбачала темні кольори та мінімалістичний стиль.

За допомогою мови Python та SQL у програмному додатку PyCharm було написано код програми, у якому реалізоване підключення до бази даних, зміна даних та графічний інтерфейс адміністративної панелі, який у свою чергу було створено за допомогою бібліотеки Customtkinter. Зв'язок між сайтом та додатком здійснюється за допомогою бібліотеки PyMySQL.

СПИСОК ЛІТЕРАТУРИ

1. Mysql documentation [Електронний ресурс] // dev.mysql.com. – 2023. – Режим доступу до ресурсу: <https://dev.mysql.com/doc/> (Дата звернення: 07.05.2023 р.).
2. Python documentation [Електронний ресурс] // docs.python.org. – 2023. – Режим доступу до ресурсу: <https://docs.python.org/3/> (Дата звернення: 07.05.2023 р.).
3. customtkinter [Електронний ресурс] // <https://pypi.org/>. – 2023. – Режим доступу до ресурсу: <https://pypi.org/project/customtkinter/0.3/> (Дата звернення: 07.05.2023 р.).
4. Connect to MySQL using PyMySQL in Python [Електронний ресурс] // www.geeksforgeeks.org. – 2023. – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/connect-to-mysql-using-pymysql-in-python/> (Дата звернення: 07.05.2023 р.).
5. Customize a treeview [Електронний ресурс] // riptutorial.com. – 2023. – Режим доступу до ресурсу: <https://riptutorial.com/tkinter/example/31885/customize-a-treeview> (Дата звернення: 07.05.2023 р.).
6. PHP cookie [Електронний ресурс] // php.net. – 2023. – Режим доступу до ресурсу: <https://www.php.net/manual/ru/reserved.variables.cookies.php> (Дата звернення: 07.05.2023 р.).
7. How nested functions are used in Python? [Електронний ресурс] // analyticsvidhya.com. – 2023. – Режим доступу до ресурсу: <https://www.analyticsvidhya.com/blog/2021/08/how-nested-functions-are-used-in-python/> (Дата звернення: 07.05.2023 р.).
8. MySQL IN Operator [Електронний ресурс] // www.w3schools.com. – 2023. – Режим доступу до ресурсу:

- https://www.w3schools.com/mysql/mysql_in.asp (Дата звернення: 07.05.2023 р.).
9. Install, uninstall, and upgrade packages [Електронний ресурс] // jetbrains.com. – 2023. – Режим доступу до ресурсу: <https://www.jetbrains.com/help/pycharm/installing-uninstalling-and-upgrading-packages.html> (Дата звернення: 07.05.2023 р.).
10. auto-py-to-exe [Електронний ресурс] // pypi.org. – 2023. – Режим доступу до ресурсу: <https://pypi.org/project/auto-py-to-exe/> (Дата звернення: 07.05.2023 р.).
11. How to Set Text of Tkinter Text Widget With a Button? [Електронний ресурс] // geeksforgeeks.org. – 2023. – Режим доступу до ресурсу: www.geeksforgeeks.org (Дата звернення: 07.05.2023 р.).

ДОДАТОК А ПРОГРАМНИЙ КОД

main.py

```

import customtkinter
import conn
from tkinter import ttk
import shutil
import webbrowser
import os

customtkinter.set_appearance_mode("dark")
customtkinter.set_default_color_theme("green")

root = customtkinter.CTk()
x = (root.winfo_screenwidth() / 2) - (500 / 2)
y = (root.winfo_screenheight() / 2) - (350 / 2)
root.geometry("%dx%d+%d+%d" % (500, 350, x, y))

def login():
    check_adm = entry1.get()
    log1 = entry1.get()
    pass1 = entry2.get()
    conn.allusers()
    for row in conn.allusers():
        if (str(log1) == row[1] and str(pass1) == row[2]):
            def closingall():
                root.destroy()
            window = customtkinter.CTkToplevel(root)
            window.title('Panel')
            x = (root.winfo_screenwidth() / 2) - (1400 / 2)
            y = (root.winfo_screenheight() / 2) - (850 / 2)
            window.geometry("%dx%d+%d+%d" % (1400, 850, x, y))
            root.withdraw()
            window.protocol("WM_DELETE_WINDOW", closingall)
            window.resizable(False, False)
            frame1 = customtkinter.CTkFrame(master=window,
            fg_color="#242424", height=600, width=600)
            frame1.pack(pady=10, padx=10, fill="both")
            def adduser():
                window2 = customtkinter.CTkToplevel(window)
                window2.title('User configure')
                x = (root.winfo_screenwidth() / 2) - (500 / 2)

```

```

y = (root.winfo_screenheight() / 2) - (350 / 2)
window2.geometry("%dx%d+%d+%d" % (500, 350, x, y))
window2.grab_set()
window2.resizable(False, False)
frame12 = customtkinter.CTkFrame(master=window2, height=600,
width=600)
frame12.pack(pady=10, padx=10, fill="both", expand=True)
label = customtkinter.CTkLabel(master=frame12, text="add
user")
label.pack(pady=12, padx=10)
entry1 = customtkinter.CTkEntry(master=frame12,
placeholder_text="Username")
entry1.pack(pady=12, padx=10)
entry2 = customtkinter.CTkEntry(master=frame12,
placeholder_text="Password", show="*")
entry2.pack(pady=12, padx=10)
entry22 = customtkinter.CTkEntry(master=frame12,
placeholder_text="Password", show="*")
entry22.pack(pady=12, padx=10)
label13 = customtkinter.CTkLabel(master=frame12, text="Дані
некоректні")
label13.configure(text_color="#d14d3f")
label14 = customtkinter.CTkLabel(master=frame12,
text="Нового користувача додано")
label14.configure(text_color="#34c74a")
label15 = customtkinter.CTkLabel(master=frame12, text="Такий
користувач вже існує")
label15.configure(text_color="#d14d3f")
def godo():
    if (entry1.get() != "" and entry1.get() != " " and
entry2.get() == entry22.get() and entry2 != ""):
        check = 0
        for row in conn.allusers():
            if (entry1.get() == row[1]):
                check = check+1
                break
        if (check == 0):
            label15.pack_forget()
            label14.pack_forget()
            label13.pack_forget()
            label14.pack()
            conn.adduser(entry1.get(), entry2.get())

```



```

        else:
            label14.pack_forget()
            label13.pack_forget()
            label15.pack()

    else:
        label14.pack_forget()
        label15.pack_forget()
        label13.pack_forget()
        label13.pack()

    button = customtkinter.CTkButton(master=frame12, text="add",
command=godo)
    button.pack(pady=12, padx=10)

    if (check_adm == "admin"):
        button_del2 = customtkinter.CTkButton(master=frame12,
text="Видалити користувача", fg_color="#e63e3e", hover_color="#e82525")
        button_del2.pack(pady=12, padx=10)

    def exitacc():
        entry1.delete(0, "end")
        entry2.delete(0, "end")
        root.deiconify()
        window.destroy()

    gr1 = customtkinter.CTkFrame(master=frame1, fg_color="#242424",
height=450, width=250)
    gr2 = customtkinter.CTkFrame(master=frame1, fg_color="#333232" ,
height=820, width=650)
    gr1.grid(row=0, column=0)
    gr2.grid(row=0, column=1)
    gr22 = customtkinter.CTkFrame(master=gr2, fg_color="#333232" ,
height=100, width=650)
    gr221 = customtkinter.CTkFrame(master=gr22, fg_color="#333232",
height=100, width=600)

    def sortname():
        BDshow.delete(*BDshow.get_children())
        for row in conn.byname():
            BDshow.insert(parent= ' ', index="end", values=(row[0],
row[3], row[4], str(row[5])+" грн", row[6], row[7], row[8], row[9],
row[10], row[11]))

        button_name.configure(command=sortname2)

    def sortname2():

```

```

BDshow.delete(*BDshow.get_children())
for row in conn.byname2():
    BDshow.insert(parent=' ', index="end", values=(row[0],
row[3], row[4], str(row[5])+" грн", row[6], row[7], row[8], row[9],
row[10], row[11]))
    button_name.configure(command=sortname)

def sortcost():
    BDshow.delete(*BDshow.get_children())
    for row in conn.bycost():
        BDshow.insert(parent=' ', index="end", values=(row[0],
row[3], row[4], str(row[5])+" грн", row[6], row[7], row[8], row[9],
row[10], row[11]))
        button_cost.configure(command=sortcost2)
    def sortcost2():
        BDshow.delete(*BDshow.get_children())
        for row in conn.bycost2():
            BDshow.insert(parent=' ', index="end", values=(row[0],
row[3], row[4], str(row[5])+" грн", row[6], row[7], row[8], row[9],
row[10], row[11]))
            button_cost.configure(command=sortcost)

def sortdate():
    BDshow.delete(*BDshow.get_children())
    for row in conn.bydate():
        BDshow.insert(parent=' ', index="end", values=(row[0],
row[3], row[4], str(row[5])+" грн", row[6], row[7], row[8], row[9],
row[10], row[11]))
        button_date.configure(command=sortdate2)
    def sortdate2():
        BDshow.delete(*BDshow.get_children())
        for row in conn.bydate2():
            BDshow.insert(parent=' ', index="end", values=(row[0],
row[3], row[4], str(row[5])+" грн", row[6], row[7], row[8], row[9],
row[10], row[11]))
            button_date.configure(command=sortdate)

def sortid():
    BDshow.delete(*BDshow.get_children())
    for row in conn.byid():

```

```

        BDshow.insert(parent='', index="end", values=(row[0],
row[3], row[4], str(row[5])+" ррн", row[6], row[7], row[8], row[9],
row[10], row[11]))
        button_id.configure(command=sortid2)
def sortid2():
    BDshow.delete(*BDshow.get_children())
    for row in conn.byid2():
        BDshow.insert(parent='', index="end", values=(row[0],
row[3], row[4], str(row[5])+" ррн", row[6], row[7], row[8], row[9],
row[10], row[11]))
        button_id.configure(command=sortid)

selected = []
def ithem_select(_):
    selected.clear()
    for i in BDshow.selection():
        selected.append(BDshow.item(i) ["values"][0])

def dellithems():
    if (len(selected)<1):
        return 0
    else:
        def yep():
            conn.delets(selected)
            for ev in selected:
                try:
os.remove(f"C:/OpenServer/domains/localhost/Store/tovar/t{ev}.png")

os.remove(f"C:/OpenServer/domains/localhost/Store/tovar/t{ev}-2.png")
                except:
                    return 0
            window3.destroy()
            BDshow.delete(*BDshow.get_children())
            for row in conn.alltovars():
                BDshow.insert(parent='', index=row[0], values=(
                    row[0], row[3], row[4], str(row[5]) + " ррн",
row[6], row[7], row[8], row[9], row[10], row[11]))
            def nope():
                window3.destroy()
            window3 = customtkinter.CTkToplevel(window)
            window3.title('delete')

```

```

x = (root.winfo_screenwidth() / 2) - (400 / 2)
y = (root.winfo_screenheight() / 2) - (150 / 2)
window3.geometry("%dx%d+%d+%d" % (400, 150, x, y))
window3.grab_set()
window3.resizable(False, False)
frame15 = customtkinter.CTkFrame(master=window3,
height=300, width=300)
frame15.pack(pady=10, padx=10, fill="both", expand=True)
label15 = customtkinter.CTkLabel(master=frame15,
text="Ви впевнені що хочете видалити обрані товари", font=("Arial", 15))
label15.pack(pady=12, padx=10)
frame16 = customtkinter.CTkFrame(master=frame15,
height=100, width=200, fg_color="#2b2b2b")
frame16.pack(pady=10, padx=10, fill="both")
button_yes = customtkinter.CTkButton(master=frame16,
text="Так", command=yep)
button_yes.grid(row=0, column=0, padx=(30, 0))
button_no = customtkinter.CTkButton(master=frame16,
text="Ні", command=nope)
button_no.grid(row=0, column=1, padx=(10, 0))

button_name = customtkinter.CTkButton(master=gr221, text="По
назві", command=sortname)
button_name.grid(row=0, column=0)

button_cost = customtkinter.CTkButton(master=gr221, text="По
ціні", command=sortcost)
button_cost.grid(row=0, column=1, padx=(10, 10), pady=(10, 10))

button_date = customtkinter.CTkButton(master=gr221, text="По
Даті", command=sortdate)
button_date.grid(row=1, column=0, padx=(10, 10))

button_id = customtkinter.CTkButton(master=gr221, text="По
Айді", command=sortid)
button_id.grid(row=1, column=1, padx=(10, 10))

def order_window():
defahaha(del_var):
conn.changes1(del_var, entry1.get())
window4 = customtkinter.CTkToplevel(window)
window4.title('Orders')

```

```

x = (window.wininfo_screenwidth() / 2) - (400 / 2)
y = (window.wininfo_screenheight() / 2) - (150 / 2)
window4.geometry("%dx%d+%d+%d" % (1000, 550, x, y))
window4.grab_set()
window4.resizable(False, False)
frame16 = customtkinter.CTkScrollableFrame(master=window4,
height=300, width=300)
frame16.pack(pady=10, padx=10, fill="both", expand=True)
frame16.pack_propagate(0)
i=0
hy =[]
for row in conn.find_orders():
    ord1 = customtkinter.CTkFrame(master=frame16,
height=100, width=980, fg_color="#454444")
    ids1 = customtkinter.CTkFrame(master=ord1, height=100,
width=50, fg_color="#454444")
    ids1.grid(row=0, column=0)
    ids1.grid_propagate(0)
    labelsm = customtkinter.CTkLabel(master=ids1,
text=row[0])
    labelsm.pack(pady=12, padx=10)

    nameltellmail1 = customtkinter.CTkFrame(master=ord1,
height=100, width=200, fg_color="#454444")
    nameltellmail1.grid(row=0, column=1)
    nameltellmail1.grid_propagate(0)
    labelsm = customtkinter.CTkLabel(master=nameltellmail1,
text=row[2])
    labelsm.grid(row=0, column=0, padx=(20, 10), pady=(10, 0))
    labelsm = customtkinter.CTkLabel(master=nameltellmail1,
text=row[4])
    labelsm.grid(row=1, column=0, padx=(20, 10))
    labelsm = customtkinter.CTkLabel(master=nameltellmail1,
text=row[5])
    labelsm.grid(row=2, column=0, padx=(20, 10))

    placetol = customtkinter.CTkFrame(master=ord1,
height=100, width=160, fg_color="#454444")
    placetol.grid(row=0, column=2)
    placetol.grid_propagate(0)

```

```

        labelsm      =      customtkinter.CTkLabel(master=placetol,
text=row[3], wraplength=150)
        labelsm.grid(row=0, column=0, padx=(10, 10),pady=(10,0))

        ordds1 = customtkinter.CTkFrame(master=ordl, height=100,
width=380, fg_color="#666565")
        ordds1.grid(row=0, column=3)
        ordds1.grid_propagate(0)
        labelsm      =      customtkinter.CTkLabel(master=ordds1,
text=row[6], wraplength=350)
        labelsm.grid(row=0, column=0, padx=(10, 10),pady=(10,0))
        # ordd_text = customtkinter.CTkTextbox(master=ordds1,
height=70, width=170)
        # ordd_text.grid(row=0, column=0)

        stattl = customtkinter.CTkFrame(master=ordl, height=100,
width=300, fg_color="#454444")
        stattl.grid(row=0, column=4)
        stattl.grid_propagate(0)
        labelsm      =      customtkinter.CTkLabel(master=stattl,
text=row[7])
        labelsm.grid(row=0, column=0,padx=(20, 10))
        hy.append(row[0])
        button_take  =      customtkinter.CTkButton(master=stattl,
text="Виконати", command= lambda i=i:ahaha(hy[i]))
        button_take.grid(row=1,          column=0,          pady=(10,
10),padx=(20, 10))

        ordl.grid(row=i, column=0, padx=(10, 10),pady=(0,10))
        ordl.grid_propagate(0)
        i=i+1

        button_orders      =      customtkinter.CTkButton(master=gr221,
text="Замовлення",          command=order_window,          fg_color="#d99c2b",
hover_color="#d9912b")
        button_orders.grid(row=0, column=2, padx=(10, 10))

```

```

        button_delete = customtkinter.CTkButton(master=gr221,
text="Видалити", command=dellithems, fg_color="#e63e3e",
hover_color="#e82525")
        button_delete.grid(row=1, column=2, padx=(10, 10))

        gr222 = customtkinter.CTkFrame(master=gr22, fg_color="#333232",
height=100, width=200)
        button_add = customtkinter.CTkButton(master=gr222, text="Додати
користувача", command=adduser)
        button_exit = customtkinter.CTkButton(master=gr222,
text="Вийти", command=exitacc)
        button_add.grid(row=0, column=0, pady=(10, 10))
        button_exit.grid(row=1, column=0)
        gr2211 = customtkinter.CTkFrame(master=gr22, fg_color="#333232",
height=100, width=450)
        gr221.grid(row=0, column=0)
        gr2211.grid(row=0, column=1)
        gr222.grid(row=0, column=2)
        gr23 = customtkinter.CTkFrame(master=gr2, fg_color="#333232",
height=720, width=650)
        gr22.grid(row=0, column=0)
        gr23.grid(row=1, column=0)
        BDshow = ttk.Treeview(master=gr23,
columns=('id', 'name', 'Sklad', 'Cost', 'type', 'country', 'year',
'material', 'color', 'sex'), show='headings', height=33)
        BDshow.column("id", width=50)
        BDshow.heading('id', text='Id')
        BDshow.column("name", width=150)
        BDshow.heading('name', text='name')
        BDshow.column("Sklad", width=200)
        BDshow.heading('Sklad', text='Sklad')
        BDshow.column("Cost", width=100)
        BDshow.heading('Cost', text='Cost')
        BDshow.column("type", width=100)
        BDshow.heading('type', text='type')
        BDshow.column("country", width=100)
        BDshow.heading('country', text='country')
        BDshow.column("year", width=100)
        BDshow.heading('year', text='year')
        BDshow.column("material", width=100)
        BDshow.heading('material', text='material')
        BDshow.column("color", width=100)

```

```

BDshow.heading('color', text='color')
BDshow.column("sex", width=100)
BDshow.heading('sex', text='sex')
BDshow.pack(pady=12, padx=10, fill='both', expand=True)
def site_go(_):
    for i in BDshow.selection():

webbrowser.open(f'http://localhost/Store/page.php?id={BDshow.item(i) ["v
alues"][0]}')
BDshow.bind("<<TreeviewSelect>>", ithem_select)
BDshow.bind("<Double-1>", site_go)
for row in conn.alltovars():
    BDshow.insert(parent= ' ',index=row[0], values=(row[0],
row[3], row[4], str(row[5])+" грн", row[6], row[7], row[8], row[9],
row[10], row[11]))

Name = customtkinter.CTkFrame(master=gr1, fg_color="#262626",
height=50, width=250)
AddtoDB = customtkinter.CTkFrame(master=gr1, fg_color="#262626",
height=350, width=250)
Name.grid(row=0, column=0)
AddtoDB.grid(row=1, column=0)
First = customtkinter.CTkLabel(master=Name, text="Додати новий
товар", width=250)
First.pack()
def open1():
    global smth
    smth = customtkinter.filedialog.askopenfilename()
    if(smth is not None and smth != ""):
        Photo1.configure(fg_color="grey")
    else:
        smth = ""
def open2():
    global smth2
    smth2 = customtkinter.filedialog.askopenfilename()
    if(smth2 is not None and smth2 != ""):
        Photo2.configure(fg_color="grey")
def addtovar():
    if (len(tname.get())<=1):
        windowex = customtkinter.CTkToplevel(window)
        windowex.title('error')

```



```

        x = (root.winfo_screenwidth() / 2) - (200 / 2)
        y = (root.winfo_screenheight() / 2) - (80 / 2)
        windowex.geometry("%dx%d+%d+%d" % (200, 80, x, y))
        windowex.grab_set()
        windowex.resizable(False, False)
        frame1 = customtkinter.CTkFrame(master=windowex,
fg_color="#242424", height=600, width=600)
        frame1.pack(pady=10, padx=10, fill="both")
        labelr = customtkinter.CTkLabel(master=frame1,
text="Занадто коротка назва")
        labelr.pack(pady=12, padx=10)

elif (len(country.get())<=1):
    windowex = customtkinter.CTkToplevel(window)
    windowex.title('error')
    x = (root.winfo_screenwidth() / 2) - (200 / 2)
    y = (root.winfo_screenheight() / 2) - (80 / 2)
    windowex.geometry("%dx%d+%d+%d" % (200, 80, x, y))
    windowex.grab_set()
    windowex.resizable(False, False)
    frame1 = customtkinter.CTkFrame(master=windowex,
fg_color="#242424", height=600, width=600)
    frame1.pack(pady=10, padx=10, fill="both")
    labelr = customtkinter.CTkLabel(master=frame1,
text="Введіть країну")
    labelr.pack(pady=12, padx=10)
else:
    try: value = int(cost.get())
    except:
        windowex = customtkinter.CTkToplevel(window)
        windowex.title('error')
        x = (root.winfo_screenwidth() / 2) - (200 / 2)
        y = (root.winfo_screenheight() / 2) - (80 / 2)
        windowex.geometry("%dx%d+%d+%d" % (200, 80, x, y))
        windowex.grab_set()
        windowex.resizable(False, False)
        frame1 = customtkinter.CTkFrame(master=windowex,
fg_color="#242424", height=600, width=600)
        frame1.pack(pady=10, padx=10, fill="both")
        labelr = customtkinter.CTkLabel(master=frame1,
text="Введіть коректну ціну")
        labelr.pack(pady=12, padx=10)

```

```

else:
    try: value = int(year.get())
    except:
        windowex = customtkinter.CTkToplevel(window)
        windowex.title('error')
        x = (root.winfo_screenwidth() / 2) - (200 / 2)
        y = (root.winfo_screenheight() / 2) - (80 / 2)
        windowex.geometry("%dx%d+%d+%d" % (200, 80, x,
y))

        windowex.grab_set()
        windowex.resizable(False, False)
        frame1 = customtkinter.CTkFrame(master=windowex,
fg_color="#242424", height=600, width=600)
        frame1.pack(pady=10, padx=10, fill="both")
        labelr = customtkinter.CTkLabel(master=frame1,
text="Введіть коректний рік")
        labelr.pack(pady=12, padx=10)
    else:
        ids = conn.findmid()
        try:
            new1 = smth[:::-1].split("/")
            ph1 = new1[0][:::-1]
            shutil.copy(smth,
f"C:/OpenServer/domains/localhost/Store/tovar/t{ids}.png")
        except NameError:
            ph1 = ""
        try:
            new2 = smth2[:::-1].split("/")
            ph2 = new2[0][:::-1]
            shutil.copy(smth2,
f"C:/OpenServer/domains/localhost/Store/tovar/t{ids}-2.png")
        except NameError:
            ph2 = ""
        conn.addtovar(f"t{ids}.png", f"t{ids}-
2.png", tname.get(), sklad.get(), cost.get(), type.get(), country.get(), year
.get(),

material.get(), color.get(), sex.get(), desc.get("1.0", "end-1c"))
        BDshow.delete(*BDshow.get_children())
        for row in conn.alltovars():
            BDshow.insert(parent='',          index=row[0],
values=(

```

```

        row[0], row[3], row[4], str(row[5]) + "
рн", row[6], row[7], row[8], row[9], row[10],
        row[11]))
    Photo1.configure(fg_color="#2fa572")
    Photo2.configure(fg_color="#2fa572")
    tname.delete(0, "end")
    sklad.delete(0, "end")
    cost.delete(0, "end")
    country.delete(0, "end")
    year.delete(0, "end")
    material.delete(0, "end")
    color.delete(0, "end")
    sex.delete(0, "end")
    desc.delete("1.0", "end-1c")

    Photo1 = customtkinter.CTkButton(master=AddtoDB, text="Додати
перше фото", command=open1)
    Photo1.pack(pady=12, padx=10)
    Photo2 = customtkinter.CTkButton(master=AddtoDB, text="Додати
друге фото", command=open2)
    Photo2.pack(pady=12, padx=10)
    label1 = customtkinter.CTkLabel(master=AddtoDB, text="Ім'я",
width=250)
    label1.pack()
    tname = customtkinter.CTkEntry(master=AddtoDB)
    tname.pack()
    label2 = customtkinter.CTkLabel(master=AddtoDB, text="Склад",
width=250)
    label2.pack()
    sklad = customtkinter.CTkEntry(master=AddtoDB)
    sklad.pack()
    label3 = customtkinter.CTkLabel(master=AddtoDB, text="Ціна",
width=250)
    label3.pack()
    cost = customtkinter.CTkEntry(master=AddtoDB)
    cost.pack()
    label4 = customtkinter.CTkLabel(master=AddtoDB, text="Тип",
width=250)
    label4.pack()
    type = customtkinter.CTkOptionMenu(master=AddtoDB,
values=["Футболка", "Головні убори", "Низ", "Взуття", "Аксесуари"])

```

```

type.configure(fg_color="#343638")
type.pack()
# type = customtkinter.CTkEntry(master=AddtoDB)
# type.pack()
label5 = customtkinter.CTkLabel(master=AddtoDB, text="Країна",
width=250)
label5.pack()
country = customtkinter.CTkEntry(master=AddtoDB)
country.pack()
label6 = customtkinter.CTkLabel(master=AddtoDB, text="Рік",
width=250)
label6.pack()
year = customtkinter.CTkEntry(master=AddtoDB)
year.pack()
label7 = customtkinter.CTkLabel(master=AddtoDB, text="Матеріал",
width=250)
label7.pack()
material = customtkinter.CTkEntry(master=AddtoDB)
material.pack()
label8 = customtkinter.CTkLabel(master=AddtoDB, text="Колір",
width=250)
label8.pack()
color = customtkinter.CTkEntry(master=AddtoDB)
color.pack()
label9 = customtkinter.CTkLabel(master=AddtoDB, text="Стать",
width=250)
label9.pack()
sex = customtkinter.CTkEntry(master=AddtoDB)
sex.pack()
label10 = customtkinter.CTkLabel(master=AddtoDB, text="Опис",
width=250)
label10.pack()
desc = customtkinter.CTkTextbox(master=AddtoDB, width=200,
height=80)
desc.pack(pady=12, padx=10)
button1 = customtkinter.CTkButton(master=AddtoDB, text="Додати",
command=addtoovar)
button1.pack(pady=12, padx=10)

```

```

frame = customtkinter.CTkFrame(master=root)
frame.pack(pady=20, padx=60, fill="both", expand=True)

label = customtkinter.CTkLabel(master=frame, text="Login System")
label.pack(pady=12, padx=10)

entry1 = customtkinter.CTkEntry(master=frame,
placeholder_text="Username")
entry1.pack(pady=12, padx=10)

entry2 = customtkinter.CTkEntry(master=frame,
placeholder_text="Password", show="*")
entry2.pack(pady=12, padx=10)

button = customtkinter.CTkButton(master=frame, text = "Login",
command=login)
button.pack(pady=12, padx=10)

root.mainloop()

```

conn.py

```

import pymysql
host = "localhost"
user = "mysql"
passw = "mysql"
db_name = "store2"
try:
    connection = pymysql.connect(host=host, port=3306, user=user,
password=passw, database=db_name)
    def delets(selected):
        srt = ""
        for i in selected:
            srt = srt+str(i)+", "
        with connection.cursor() as cursor:
            insr_q = "DELETE FROM tovar WHERE id IN ({0})".format(srt[:-1])
            cursor.execute(insr_q)
            connection.commit()
    def allusers():
        with connection.cursor() as cursor:

```

```

        select_all = "SELECT * FROM users"
        cursor.execute(select_all)
        rows = cursor.fetchall()
        return rows

def adduser(name1, passw1):
    with connection.cursor() as cursor:
        nameq = name1
        passw = passw1
        select_max = "SELECT MAX(id)+1 FROM users"
        cursor.execute(select_max)
        rows = cursor.fetchall()
        maxnow = rows[0][0]
        insr_q = "INSERT INTO users (id,name, password) VALUES ({0}, '{1}',
' {2} )".format(maxnow,nameq,passw)
        cursor.execute(insr_q)
        connection.commit()

    def addtovar(image1q, image2q, Nameq, Skladq, Costq, typeq, Countryq,
yearq, materialq, Colorq, sexq, Descrq):
        with connection.cursor() as cursor:
            insr_q = "INSERT INTO tovar
(id,image1,image2,Name,Sklad,Cost,type,Country,year,material,Color,sex,Descr)
" \
                "VALUES ({0}, '{1}',
' {2}', '{3}', '{4}', {5}, '{6}', '{7}', {8}, '{9}', '{10}', '{11}', '{12} )".format(fin
dmid(), image1q, image2q, Nameq, Skladq, Costq, typeq, Countryq,
yearq, materialq, Colorq, sexq, Descrq)
            cursor.execute(insr_q)
            connection.commit()

def findmid():
    with connection.cursor() as cursor:
        select_max = "SELECT MAX(id)+1 FROM tovar"
        cursor.execute(select_max)
        rows = cursor.fetchall()
        maxnow = rows[0][0]
        return maxnow

def alltovars():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

```

```
def allorders():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM orders"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def byname():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar ORDER BY name ASC"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def byname2():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar ORDER BY name DESC"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def bycost():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar ORDER BY Cost DESC"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def bycost2():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar ORDER BY Cost ASC "
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def bydate():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar ORDER BY year DESC"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def bydate2():
```

```

with connection.cursor() as cursor:
    selectt = "SELECT * FROM tovar ORDER BY year ASC"
    cursor.execute(selectt)
    rows = cursor.fetchall()
    return rows

def byid():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar ORDER BY Id DESC"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def byid2():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM tovar ORDER BY Id ASC"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def find_orders():
    with connection.cursor() as cursor:
        selectt = "SELECT * FROM order1 ORDER BY id DESC"
        cursor.execute(selectt)
        rows = cursor.fetchall()
        return rows

def changes1(idvar, arr):
    with connection.cursor() as cursor:
        insr_q = "UPDATE order1 SET statt = 'Виконано({0})' WHERE id
={1}".format(arr, idvar)
        cursor.execute(insr_q)
        connection.commit()

except Exception as ex:
    print("conn refused...")
    print(ex)

```

store.php

```

<!DOCTYPE html>
<html>
<head>

```



```

    <link rel="stylesheet" type="text/css" href="CSS/store.css">
<style type="text/css">
    .modal{
        display: none;
        position: fixed;
        z-index: 1001;
        left: 0;
        top: 0;
        width: 100%;
        height: 100%;
        overflow: hidden;
        background-color: rgba(0,0,0,0.8);
    }
    .modal-cont{
        background: white;
        margin: 0;
        padding: 0;
        border: 1px solid grey;
        width: 20%;
        height: 100vh;
        min-width: 350px;
        box-shadow: 0 0 10px rgba(0,0,0,0.5);
    }
</style>
<title>Store</title>
</head>
<body>
    <?php
    include 'connect.php';
    ?>
<div id="head_store">
    <div id="head_store1">
        <a href="page_buy_korz.php"></a>
    </div>
    <div id="head_store2">
        <div id="logo_b">
            
        </div>
    </div>
    <div id="head_store3">

```

```

        <div id="head_store3-1">
            <b>Працюємо Щодня 7:00 - 20:00</b><br>
            Зв'язок з нами:<br>
            +380665178754

        </div>
    </div>

</div>
<div id="underhead_store0">
<div id="underhead_store">
    <div id="underhead_store1">
        Футболки
    </div>
    <div id="underhead_store2">
        Головні убори
    </div>
    <div id="underhead_store4">
        Низ
    </div>
    <div id="underhead_store5">
        Взуття
    </div>
    <div id="underhead_store3">
        Інші аксесуари
    </div>
</div>
</div>
<div id="main_store">
    <div id="main_store1">

        <?php

        $sql = "SELECT * FROM tovar where type = 'Футболка'
ORDER BY Id DESC";

        $result = $conn->query($sql);
        while($row = $result->fetch_assoc() ) {

```

```

        $t = $row['Id'];

        echo '
        <div id="store_tov" class="countlt">
        <a href="page.php?id='.$row['Id'].'">
        <div id="store_tov1" class="scale">
        <div id="store_tov1_1">
        

        </div>
        <div id="store_tov1_2">
        <font size="4">
        '. $row["Name"] .'
        </font><br>
        <font size="2">'. $row["Cost"] .' грн</font>
        </div>
        </div></a>
        </div>

        ';
    }
    ?>

    <div id="list-over">
        Це всі товари на даний момент
    </div>

</div>
<div id="main_store2">

<?php

```

```

ORDER BY Id DESC";

$sql = "SELECT * FROM tovar where type = 'Головні убори'";
$result = $conn->query($sql);
while($row = $result->fetch_assoc() ) {

    $t = $row['Id'];

    echo '
<div id="store_tov" class="countlt1">
<a href="page.php?id='.$row['Id'].'">
<div id="store_tov1" class="scale">
<div id="store_tov1_1">

</div>
<div id="store_tov1_2">
<font size="4">
'. $row["Name"] .'
</font><br>
<font size="2">'. $row["Cost"] .' грн</font>
</div>
</div></a>
</div>

';
}
?>

<div id="list-over">
    Це всі товари на даний момент
</div>

</div>

<div id="main_store4">

```

```

<?php

$sql = "SELECT * FROM tovar where type = 'Низ' ORDER BY
Id DESC";

$result = $conn->query($sql);
while($row = $result->fetch_assoc() ) {

    $t = $row['Id'];

    echo '
<div id="store_tov" class="countlt2">
<a href="page.php?id='.$row['Id'].'">
<div id="store_tov1" class="scale">
<div id="store_tov1_1">


</div>
<div id="store_tov1_2">
<font size="4">
'. $row["Name"] .'
</font><br>
<font size="2">'. $row["Cost"] .' грн</font>
</div>
</div></a>
</div>

';
}
?>

<div id="list-over">
    Це всі товари на даний момент
</div>

</div>

```

```

<div id="main_store5">

    <?php

        $sql = "SELECT * FROM tovar where type = 'Взуття' ORDER
BY Id DESC";

        $result = $conn->query($sql);
        while($row = $result->fetch_assoc() ) {

            $t = $row['Id'];

            echo '
<div id="store_tov" class="countlt3">
<a href="page.php?id='.$row['Id'].'">
<div id="store_tov1" class="scale">
<div id="store_tov1_1">


</div>
<div id="store_tov1_2">
<font size="4">
'. $row["Name"] .'
</font><br>
<font size="2">'. $row["Cost"] .' грн</font>
</div>
</div></a>
</div>

';

        }
    ?>

<div id="list-over">
        Це всі товари на даний момент
    </div>

```

```
</div>
```

```
<div id="main_store3">
```

```
<?php
```

```

$sql = "SELECT * FROM tovar where type = 'Аксессуары'
ORDER BY Id DESC";

$result = $conn->query($sql);
while($row = $result->fetch_assoc() ) {

    $t = $row['Id'];

    echo '
<div id="store_tov" class="countlt4">
<a href="page.php?id='.$row['Id'].'">
<div id="store_tov1" class="scale">
<div id="store_tov1_1">


</div>
<div id="store_tov1_2">
<font size="4">
'. $row["Name"] .'
</font><br>
<font size="2">'. $row["Cost"] .' ррн</font>
</div>
</div></a>
</div>

';
}

```

```

        ?>

<div id="list-over">
        Це всі товари на даний момент
    </div>

</div>
</div>

<div id="bottom_store">
    <div id="bottom_store-1">

    </div>
    <div id="bottom_store-2">

        TviyStylcomp@gmail.com<br>
        Твій Стиль@<br>
        2022
    </div>
    <div id="bottom_store-3">

    </div>
</div>

<div id="mymodal" class="modal">
    <div class="modal-cont">
        
        <div id="menu_content1">
            <?php
            include 'menu_cont.php';
            ?>
        </div>
    </div>
</div>
<script type="text/javascript"></script>
<script>
    var under1 = document.getElementById('underhead_store1');
    var under2 = document.getElementById('underhead_store2');
```



```
var under3 = document.getElementById('underhead_store3');
var under4 = document.getElementById('underhead_store4');
var under5 = document.getElementById('underhead_store5');
var blu1 = document.getElementById('main_store1');
var blu2 = document.getElementById('main_store2');
var blu3 = document.getElementById('main_store3');
var blu4 = document.getElementById('main_store4');
var blu5 = document.getElementById('main_store5');
under1.onclick = function (){
    under1.style.background = "#e63737";
    under1.style.color = "white";
    blu1.style.display = "block"

    under2.style.background = "none";
    under2.style.color = "black";
    blu2.style.display = "none"

    under3.style.background = "none";
    under3.style.color = "black";
    blu3.style.display = "none"

    under4.style.background = "none";
    under4.style.color = "black";
    blu4.style.display = "none"

    under5.style.background = "none";
    under5.style.color = "black";
    blu5.style.display = "none"
}
under2.onclick = function (){
    under2.style.background = "#e63737";
    under2.style.color = "white";
    blu2.style.display = "block"

    under1.style.background = "none";
    under1.style.color = "black";
    blu1.style.display = "none"

    under3.style.background = "none";
    under3.style.color = "black";
    blu3.style.display = "none"
```

```
under4.style.background = "none";
under4.style.color = "black";
blu4.style.display = "none"

under5.style.background = "none";
under5.style.color = "black";
blu5.style.display = "none"
}
under3.onclick = function (){
    under3.style.background = "#e63737";
    under3.style.color = "white";
    blu3.style.display = "block"

    under1.style.background = "none";
    under1.style.color = "black";
    blu1.style.display = "none"

    under2.style.background = "none";
    under2.style.color = "black";
    blu2.style.display = "none"

    under4.style.background = "none";
    under4.style.color = "black";
    blu4.style.display = "none"

    under5.style.background = "none";
    under5.style.color = "black";
    blu5.style.display = "none"
}

under4.onclick = function (){
    under4.style.background = "#e63737";
    under4.style.color = "white";
    blu4.style.display = "block"

    under1.style.background = "none";
    under1.style.color = "black";
    blu1.style.display = "none"

    under2.style.background = "none";
    under2.style.color = "black";
    blu2.style.display = "none"
```

```

    under3.style.background = "none";
    under3.style.color = "black";
    blu3.style.display = "none"

    under5.style.background = "none";
    under5.style.color = "black";
    blu5.style.display = "none"
}

under5.onclick = function (){
    under5.style.background = "#e63737";
    under5.style.color = "white";
    blu5.style.display = "block"

    under1.style.background = "none";
    under1.style.color = "black";
    blu1.style.display = "none"

    under2.style.background = "none";
    under2.style.color = "black";
    blu2.style.display = "none"

    under4.style.background = "none";
    under4.style.color = "black";
    blu4.style.display = "none"

    under3.style.background = "none";
    under3.style.color = "black";
    blu3.style.display = "none"
};

var firsth1 = document.getElementsByClassName('countlt').length/4 * 320+20;

window.addEventListener('scroll', () => {
    const documentRect = document.documentElement.getBoundingClientRect();

    var blu11 = document.getElementById('main_store1');

```

```

if( documentRect.bottom < document.documentElement.clientHeight + 350
    && blu11.clientHeight <= firsth1){
    blu11.style.height = blu11.clientHeight + 300 + "px";
console.log('block high', blu11.clientHeight);
    if (blu11.clientHeight >= firsth1){
        blu11.style.height = "auto";
    }
}

var firsth2 = document.getElementsByClassName('countlt1').length/4 * 320+20;

var blu22 = document.getElementById('main_store2');
if( documentRect.bottom < document.documentElement.clientHeight + 350
    && blu22.clientHeight <= firsth2){
    blu22.style.height = blu22.clientHeight + 300 + "px";
    if (blu22.clientHeight >= firsth2){
        blu22.style.height = "auto";
    }
}

var firsth3 = document.getElementsByClassName('countlt2').length/4 * 320+20;

var blu33 = document.getElementById('main_store3');
if( documentRect.bottom < document.documentElement.clientHeight + 350
    && blu33.clientHeight <= firsth3){
    blu33.style.height = blu33.clientHeight + 300 + "px";
    if (blu33.clientHeight >= firsth3){
        blu33.style.height = "auto";
    }
}

var firsth4 = document.getElementsByClassName('countlt3').length/4 * 320+20;

var blu44 = document.getElementById('main_store4');
if( documentRect.bottom < document.documentElement.clientHeight + 350
    && blu44.clientHeight <= firsth4){
    blu44.style.height = blu44.clientHeight + 300 + "px";
    if (blu44.clientHeight >= firsth4){
        blu44.style.height = "auto";
    }
}

```

```
        }  
    }  
  
    var firsth5 = document.getElementsByClassName('countlt4').length/4 * 320+20;  
  
    var blu55 = document.getElementById('main_store5');  
    if( documentRect.bottom < document.documentElement.clientHeight + 350  
        && blu55.clientHeight <= firsth5){  
        blu55.style.height = blu55.clientHeight + 300 + "px";  
        if (blu55.clientHeight >= firsth5){  
            blu55.style.height = "auto";  
        }  
    }  
  
    })  
  
</script>  
  
</body>  
</html>
```