МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Державний університет “Житомирська політехніка”

Кафедра інженерії програмного забезпечення

**ЗВІТ З ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ**

Студента IV курсу групи ІПЗ-19-1

Галузь знань 12 «Інформаційні технології» спеціальність 121 «Програмна інженерія» ступінь «**бакалавр»**

Драк Тарас Сергійович

**(прізвище ініціали, підпис)**

Керівник практики: Єфремов Юрій Миколайович,

к.т.н., доцент кафедри інженерії програмного забезпечення

Кількість балів: Національна оцінка: ECTS:

Члени комісії:

Чижмотря О.В.

(підпис) (прізвище та ініціали)

Кравченко С.М.

(підпис) (прізвище та ініціали)

Власенко О.В.

(підпис) (прізвище та ініціали)

Житомир – 2023

**РЕФЕРАТ**

Завданням курсового проекту була розробка WEB-додатку інформаційної системи, що дозволяє користувачеві переглядати різноманітну інформацію, а адміністраторові створювати, та редагувати дані на сайті.

Пояснювальна записка до курсового проекту на тему «Магазин годинників» складається з вступу, трьох розділів, списку викорисаної літератури та додатків.

Текстова частина міститься на 37 сторінках друкованого тексту.

Пояснювальна записка має 44 сторінок додатків. Список використаних джерел складється з 10 найменувань та займає 1 сторінку. У роботі наведено 42 рисунки. Загальний обсяг роботи – 82 сторінок.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

PHP — Мова програмування php

PHPMyAdmin — База даних

CSS — Таблиця каскадних стилів

Зміст

[ВСТУП 8](#_Toc61526362)

[РОЗДІЛ 1 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 9](#_Toc61526363)

[1.1 Технічне завдання на розробку системи 9](#_Toc61526364)

[1.2 Аналіз вимог до програмного продукту 10](#_Toc61526365)

[1.3 Технічне завдання курсового проекту 14](#_Toc61526366)

[1.4 Проектування структури бази даних інтернет-магазину з продажу годинників Overclock 14](#_Toc61526366)

[РОЗДІЛ 2 ПРЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 16](#_Toc61526367)

[2.1 Розробка програмного забезпечення 16](#_Toc61526368)

[2.2 Опис роботи з програмним додатком 17](#_Toc61526369)

[Висновки до другого розділу 20](#_Toc61526370)

[РОЗДІЛ 3 ОПИС РОБОТИ З WEB ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ 29](#_Toc61526371)

[3.1 Опис роботи з WEB-додатку (Опис інтерфейсу) 29](#_Toc61526372)

[3.2 Тестування роботи WEB-додатку 38](#_Toc61526373)

[ВИСНОВКИ 39](#_Toc61526374)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 40](#_Toc61526375)

[ДОДАТОК 41](#_Toc61526376)

# ВСТУП

В даному курсовому проекті наведено процес розробки WEB-додатку інтернет магазину що дозволяє користувачеві проводити покупки.

Даний магазин створена для зручного перегляду актуальної інформації про конкретні товари.

Ця тема для курсового проекту є актуальною, адже інтернет магазини користуються попитом повсякчасно, а отже є ходовим товаром і потребують постійного вдосконалення. Цільовою аудиторією інтернет магазину являються усі групи населення.

Метою дослідження є створення власного WEB-додатку за допомогою якого буде можливим перегляд конкретних товарів та їх зручне редагування.

Об’єктом для дослідження виступили декілька інших магазинів, з різноманітним функціоналом.

Отже, необхідно розробити WEB-додаток, що буде мати переваги порівняно з іншими системами перегляду та редагування інформації.

# РОЗДІЛ 1 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

# Технічне завдання на розробку системи

1. Назва системи:

Повне найменування системи: Інтернет-магазин наручних годинників "Overclock" на базі PHP та phpMyAdmin Коротке найменування системи: Overclock Watch Shop, Магазин годинників Overclock, Overclock годинниковий магазин

Призначення системи: розробка частини інтернет-магазину наручних годинників "Overclock" на базі PHP та phpMyAdmin. Ця частина додатку забезпечує функціональність для відображення сторінок зі зручним інтерфейсом, збереження даних в базі даних та їх відображення, а також обробку запитів користувачів.

Вимоги до системи:

Програмний продукт повинен мати наступні функціональні характеристики:

* + Реєстрація та авторизація користувачів.
  + Перегляд каталогу товарів з можливістю фільтрації за різними параметрами (назва, тип, бренд тощо).
  + Зрозумілий та інтуїтивний дизайн;
  + Редагування товару адміністратором.
  + Наявність трьох ролей користувачів (гість, користувач, адміністратор).

1. Вимоги до інтерфейсу:

Взаємодія користувачів із програмним комплексом повинна відбуватися за допомогою веб-браузера. Інтерфейс програмного комплексу повинен бути зручним у використанні та інтуїтивно зрозумілим.

* Головна сторінка має містити меню навігації, яке дозволить користувачеві швидко знайти потрібний розділ сайту . Інтерфейс має бути зручним та простим у використанні.
* На сторінці кожного товару має бути детальна інформація про товар (назва, фото, опис тощо).
* Кнопки додавання товару до кошика та оформлення замовлення повинні бути чітко виділені на сторінці кожного товару.
* Адміністратор повинен мати змогу вільно додавати, редагувати та видаляти товари.

# 1.2 Аналіз вимог до програмного продукту

Розглянемо вимоги, яким повинен задовільняти програмний продукт, щоб мати можливість виконувати всі покладені на нього фунцій, тим самим задовольнивши всі зазначені стандарти, специфікації та інші формальні документи. Приведені можливості системи, обмеження та показники якості.

**Бізнес-вимоги:**

1. Основні цілі: створення інтернет-магазину для годнників “Overclock”.
2. Можливості: користувачі можуть шукати годинники за назвою або категорією. Адміністратори мають можливість додавати, видаляти та редагувати товари а також мають можливість додавати, видаляти та редагувати усі категорії.
3. Представлення проекту: інтернет-магазин буде реалізовано у вигляді веб- сайту зі зручним інтерфейсом та можливістю пошуку товару по категоріям.

**Вимоги користувачів**

* **Гості – неавторизований користувач**
  1. Можливість перегляду всіх товарів;
  2. Можливість сортування та фільтрації товару за атрибутами, категорією та тегами.
* **Користувачі – неавторизований користувач**
  1. Можливість перегляду всіх товарів;
  2. Можливість сортування та фільтрації товару за атрибутами, категорією та тегами.
  3. Можливість замовити товар.
* **Внутрішній користувач – працівник**
  1. Редагування та додавання постів, категорій, тегів, користувачів та коментарів.

**Характеристика об`єкту комп`ютеризації:**

Користувач буде мати можливість пошуку та перегляду товру з метою ознайомлення з новою інформацією. Адміністратор через веб-інтерфейс буде мати можливість редагувати, додавати та видаляти більше даних та користуватися всіма іншими функціями системи.

**Функціональні вимоги:**

1. Створення модуля для товарів.
2. Створення модуля для категорій.
3. Створення модуля для користувачів .

**Нефункціональні вимоги:**

1. Сприйняття
   * Ефективність: додаток повинен працювати швидко та ефективно навіть при великому обсязі даних та великому навантаженні на сервер.
   * інтерфейс повинен бути інтуїтивно зручним та зрозумілим для користування та не вимагати додаткової підготовки.
2. Продуктивність
   * Система повинна підтримувати велику кількість користувачів, що одночасно працюють з додатком.

**Системні вимоги:**

1. **Вимоги до середовища виконання:**

Система повинна задовольняти зазначеним вимогам на комп’ютері в наступній мінімальній комплектації:

* + підтримка браузера *Google Chrome* версії 45+ або
  + підтримка браузера *Mozilla Firefox* версії 38+ або
  + підтримка браузера *Microsoft Edge* версії 12+ або
  + підтримка браузера *Opera* версії *30*+ або
  + підтримка браузера *Safari* версії *9*+

1. **Вимоги до СУБД та доступу до даних:**

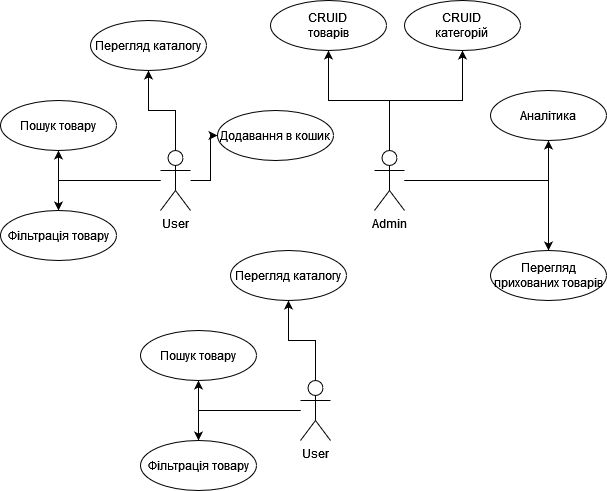
У ядрі системи повинна бути представлена СУБД;

**Бізнес правила:**

Система повинна відповідати всім стандартам побудови сучасних веб- застосунків.

**Бізнес-логіка:**

Система пропонує список постів для безкоштовного перегляду. Підсумовуючи, побудуємо діаграму варіантів використання (Use Case Diagram).



На рисунку 1.1 Діаграма варіантів використання для користувачів і адміністратора.

# Алгоритм роботи та стани програмної системи

Під час планування алгоритму роботи та станів програмного додатку спочатку необхідно побудувати діаграму активності(activity diagram).

Діаграма активності — це UML-діаграма, що показує послідовність дій, рішення, вузли, цикли та навіть одночасні дії. Діаграми активності постійно використовуються при моделюванні робочого процесу.

На рисунку 1.2 зображено діаграму активності.

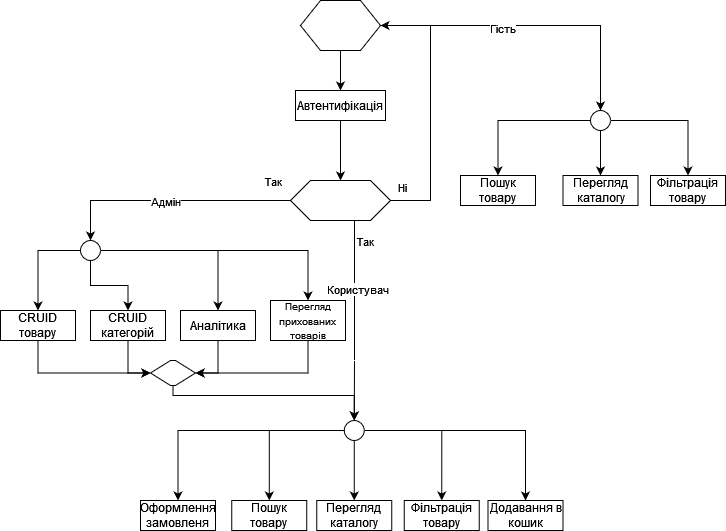


Рисунок 1.2 – Діаграма активності

**Опис діаграми діяльності:**

* Перевірка ролі Адміністратор
  + Автентифікація (так – перехід до використання системи, ні – перехід на початок з повідомленням про помилку автентифікації);
    - CRUD товарів
    - CRUD категорій
    - Перегляд прихованих товарів
    - Аналітика

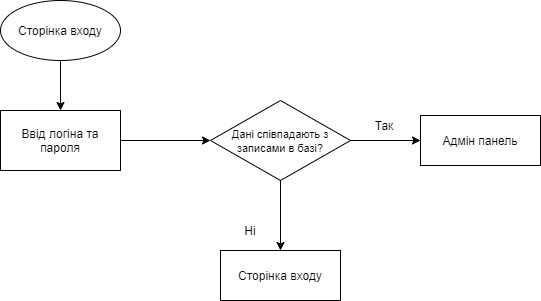


Рис. 1.3. Вхід в адміністративну панель

**Опис діаграми діяльності:**

* Перевірка ролі
  + Admin
    - Автентифікація (так – перехід до використання системи, ні – перехід на початок з повідомленням про помилку автентифікації);
      * CRUD категорій
      * CRUD товарів
  + Registered user
* Пошук товару
* Перегляд каталогу товарів
* Фільтрація товару
* Додавання товару в кошик
* Оформлення замовлення
* Unregistered user
  + Пошук товару
  + Перегляд каталогу товарів
  + Фільтрація товару
  1. **Проектування структури бази даних інтернет-магазину з продажу годинників “Overclock”**

Після отримання даних із браузеру ініціалізуємо ядро, далі перевіряємо наявність дії в контролері з іменем модуля з адресного рядку. В залежності від отриманої інформації створюємо модель, та виконуємо котролер. Далі викликаємо функцію за назвою дії.

Після обробки вхідних даних за допомогою контролера, звертаємося до моделі, що за допомогою Системи Управління Базами Даних (надалі СУБД) допоможе нам отримати дані з БД. Обробка даних відбувається за допомогою контроллеру, а також він дозволяє нам побудувати необхідну структуру сторінки.

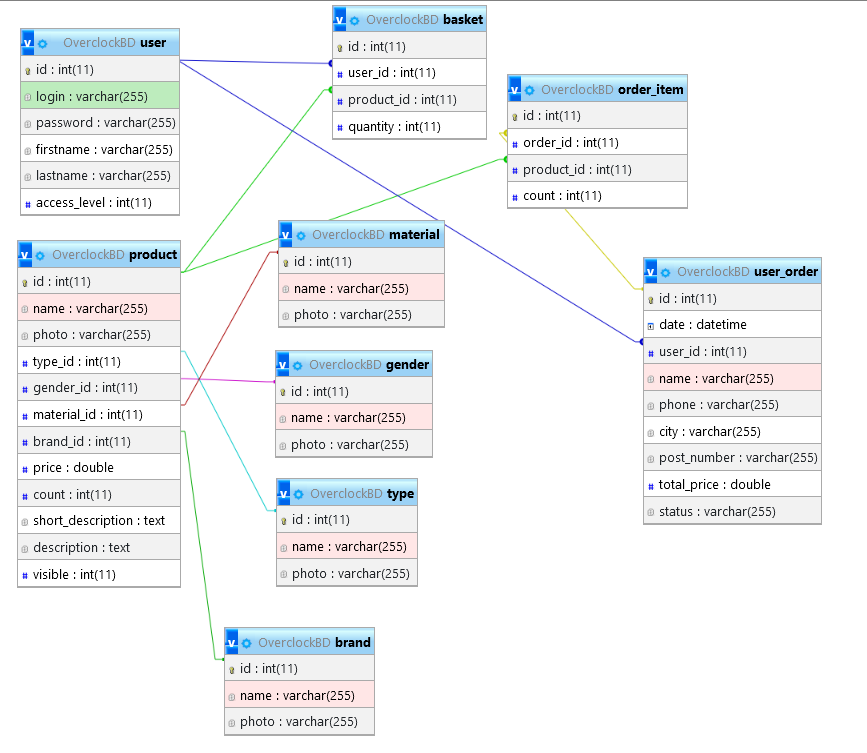


Рисунок 1.4 Колекція OrdersMode

**Висновки з першого розділу**

В даному розділі було отримано технічне завдання на розробку програмного продукту. Було виокремлено та проаналізовано вимоги до програмної системи.

Також, було розроблено модель системи веб-додатку блогу. Було побудовано діаграму активності користувача і адміністратора для демонстрації алгоритму роботи додатку та можливих станів системи. Також була побудовані колекції моделей баз данних.

# РОЗДІЛ 2 ПРЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

# 2.1 Розробка Web-додатку

На даному етапі відбувається розробка спроектованого WEB-додатку. За допомогою середовища програмування PhpStorm створюємо рішення та проєкт store.

Структура рішення:

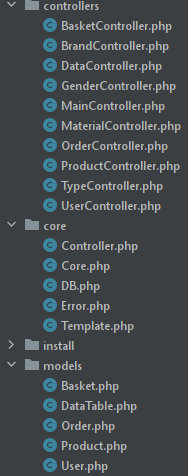


Рис 2.1. Структура файлів класів типу php

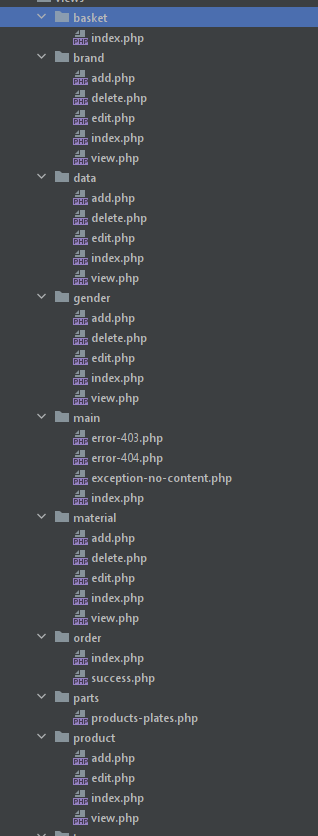


Рис 2.2. Структура файлів view типу php



Рис 2.3. Структура файлів utils та theme

Спочатку створюємо загальні для усіх сторінок ресурси, а саме сторінку layout/index, яка буде слугувати шаблоном для інших сторінок. Створюємо “шапку” та “низ” сайту за допомогою тегів header та footer. У тезі footer створюємо кнопку, за допомогою якої адміністратор зможе авторизуватися на сайті щоб змінювати дані. У центральному блоці сторінки буде знаходитись контент, згенерований контролером, та відмальований view(надалі видом).

Створюємо навігаційне меню угорі сайту, за допомогою якого буде відбуватися перехід на головну сторінку сайту, на перелік товарів, їх категорію, кошик та користувача.

Головна сторінка сайту буде відображати вітання в магазині, а також його фотографію.

Готова реалізація головної сторінки WEB-додатку зображено на Рис.2.4.

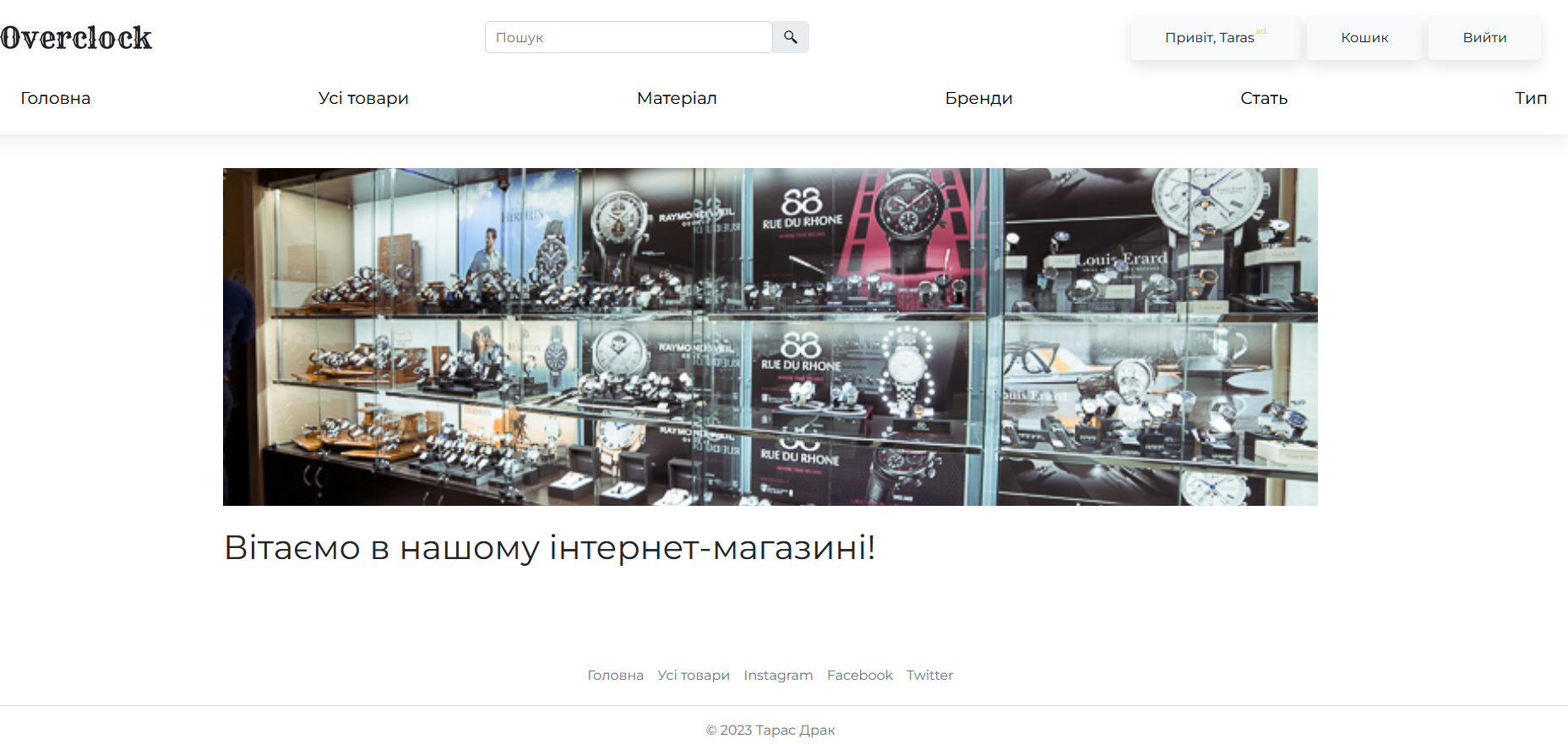


Рис. 2.4. Головна сторінка магазину

Переходимо до модуля усіма товарами. (Рис. 2.5.).

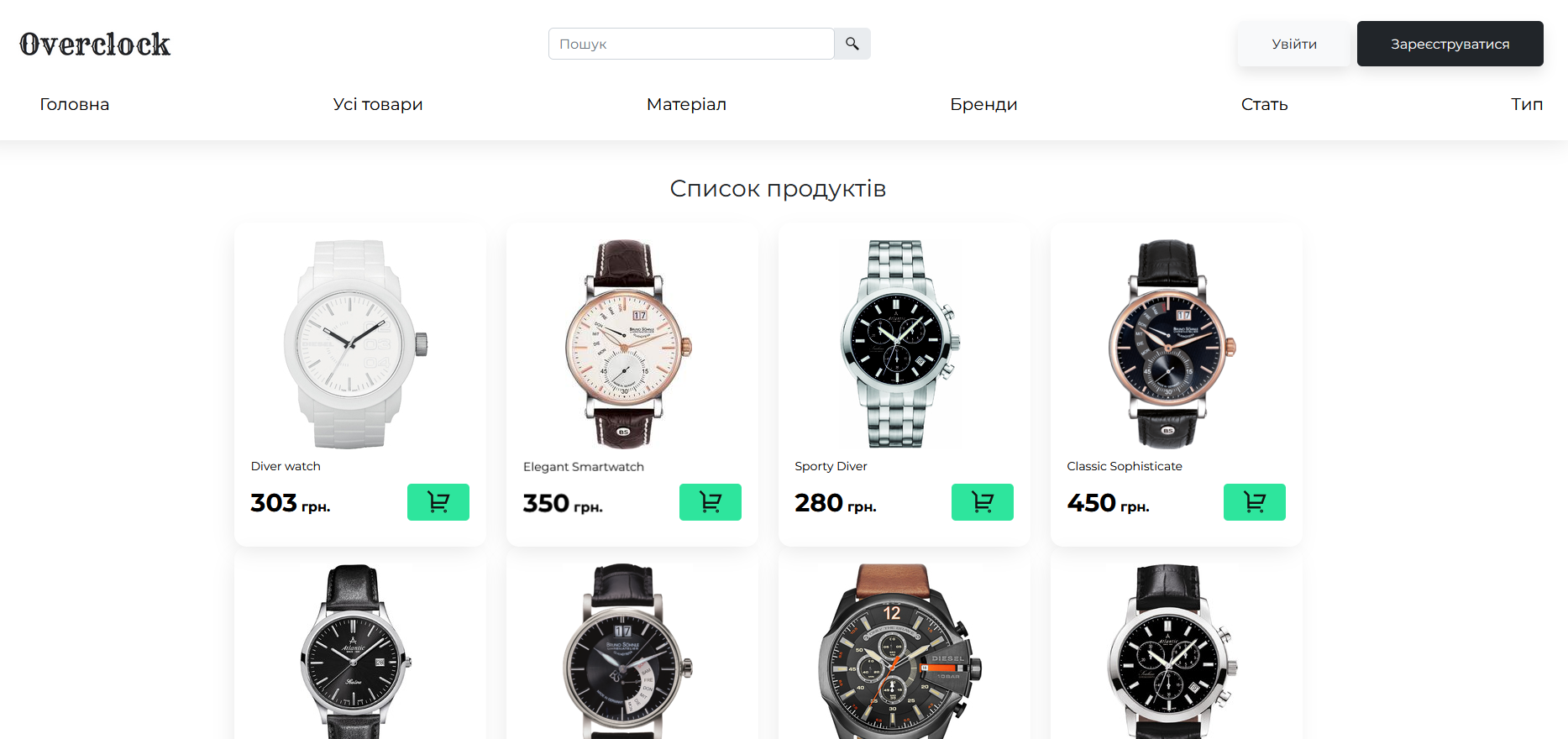


Рис. 2.5. Сторінка усіх товарів

Після головної сторінки переходимо до брендів. На ній ми бачимо список брендів (Рис. 2.6.).

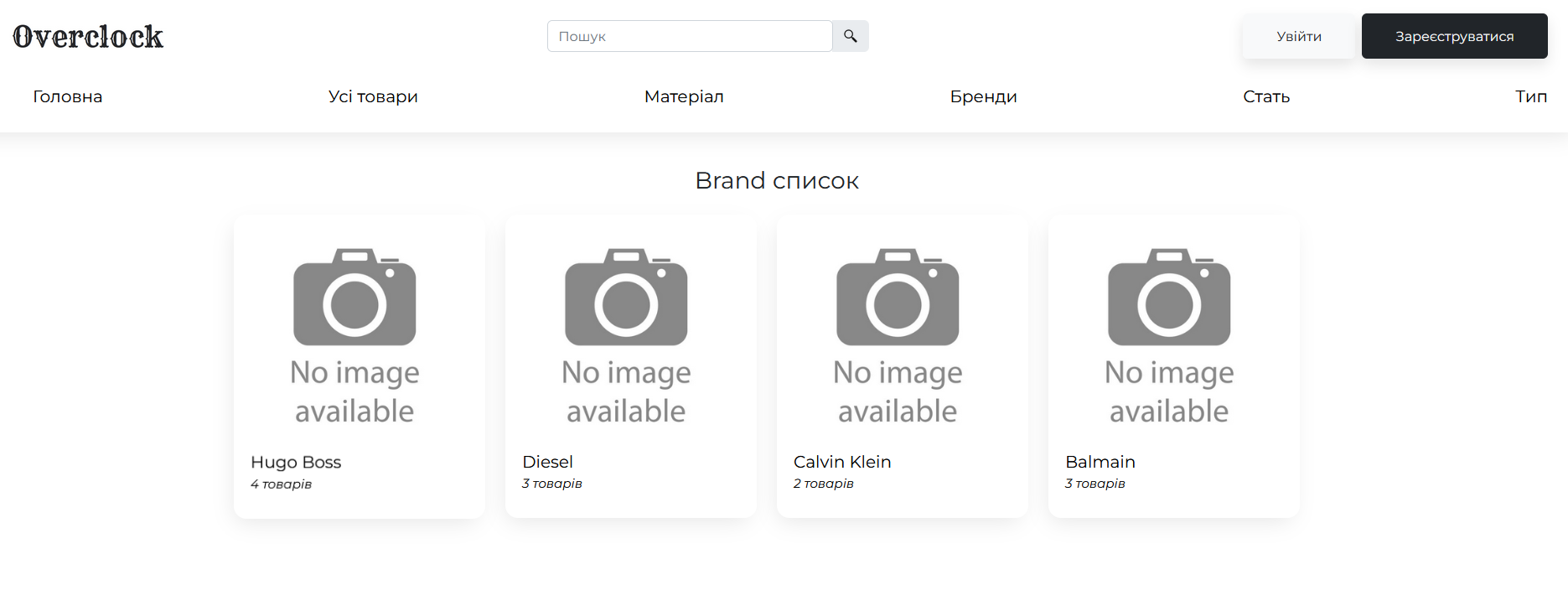


Рис. 2.6. Сторінка зі списком брендів

При одного з брендів, ми бачимо товари які йдуть до цього

бренду. (Рис. 2.7.).

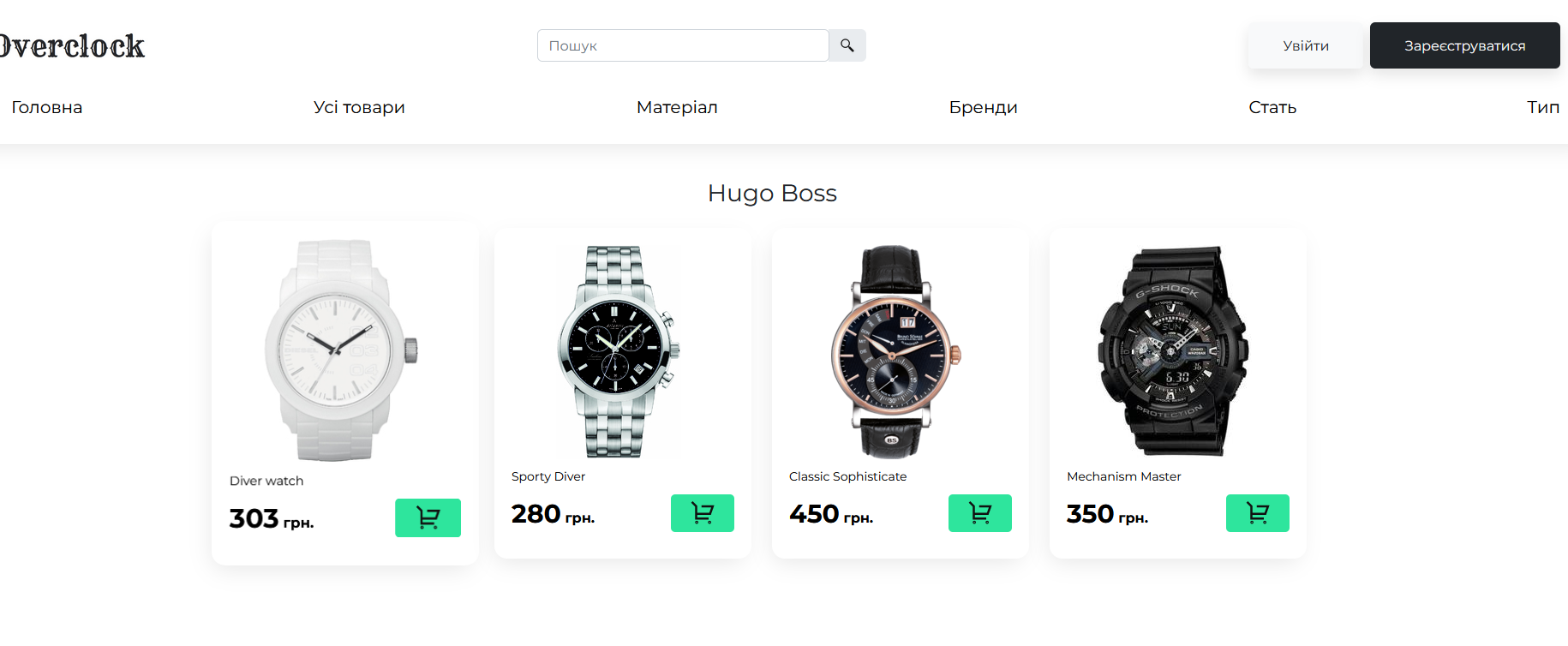


Рис. 2.7. Перелік брендів Hugo Boss

Переглянемо товар який ми обрали наприклад Diver watch. (Рис. 2.8.)

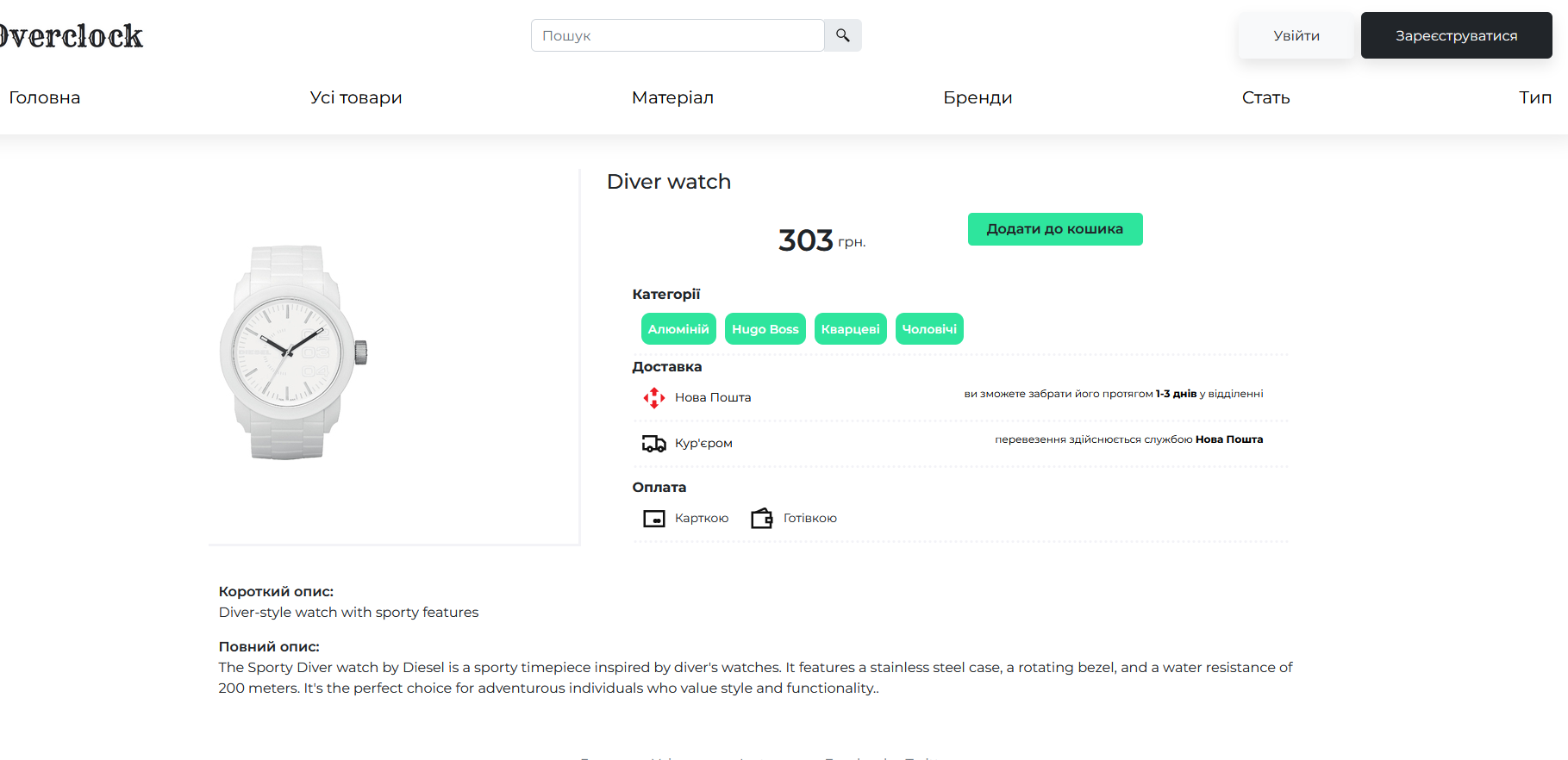


Рис. 2.8. Перегляд товару

При спробі придбати товар без логіна нас перекидує на сторінку щоб ми залогінились(Рис. 2.9.).

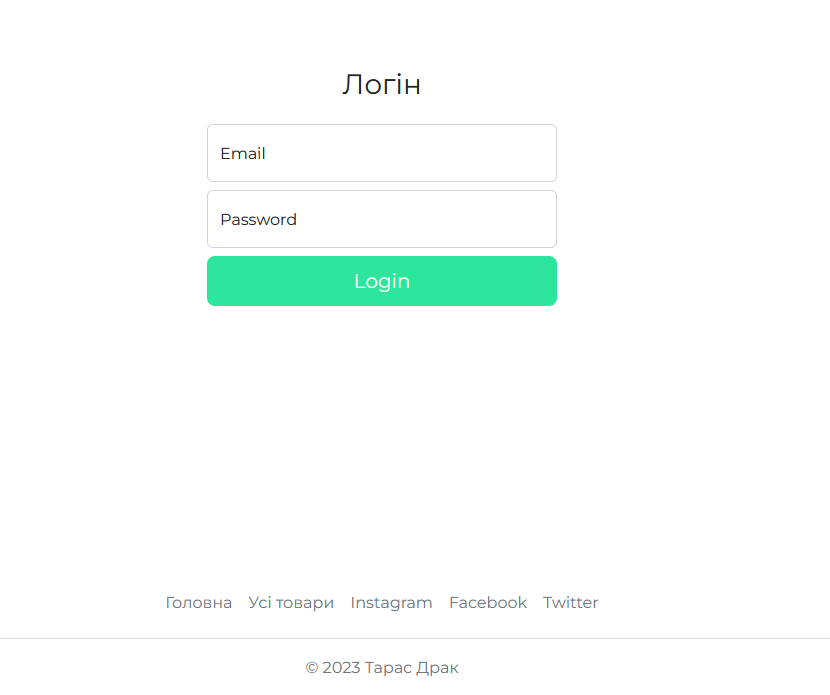


Рис. 2.9. Логін

Так як ми не зареєстрований користувач пройдемо реєстрацію використовуючи кнопку Зареєструватися в правому верхньому куту(Рис. 2.10.).

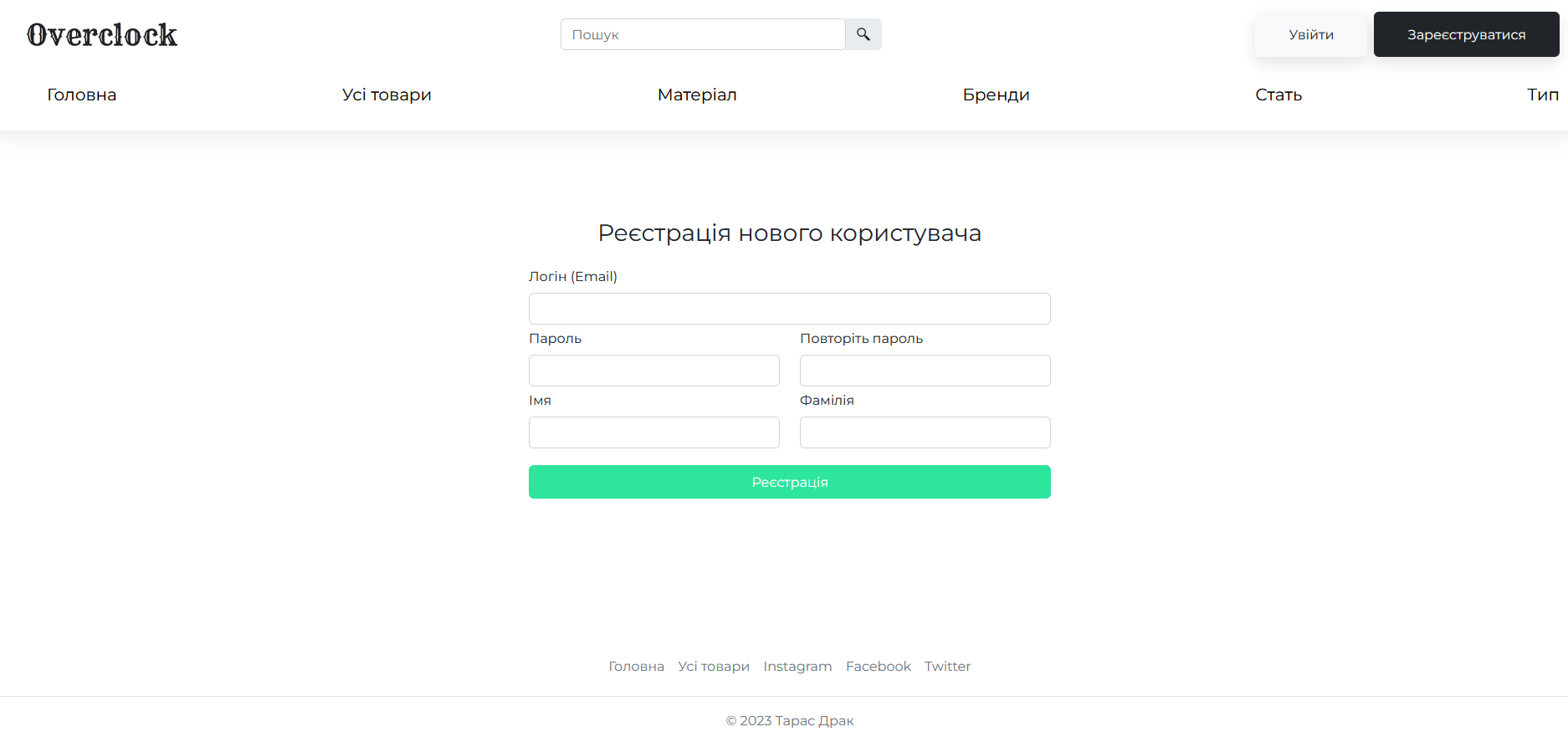


Рис. 2.10. Сторінка реєстрації

При коректному вводу даних проходить успішна реєстрація (Рис. 2.11.).

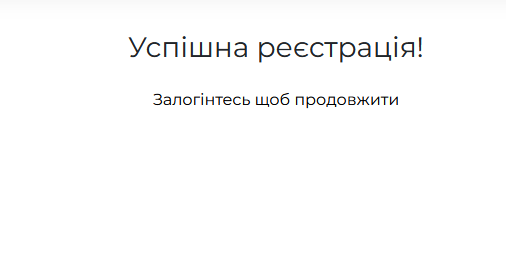


Рис.2.11. Успішна реєстрація

Тепір як зареєстрований користувач справа зверху ми можимо побачити кнопку Вийти та привітання яке перешле нас до списку замовлень зайду з адмін акаунту (Рис. 2.12.).

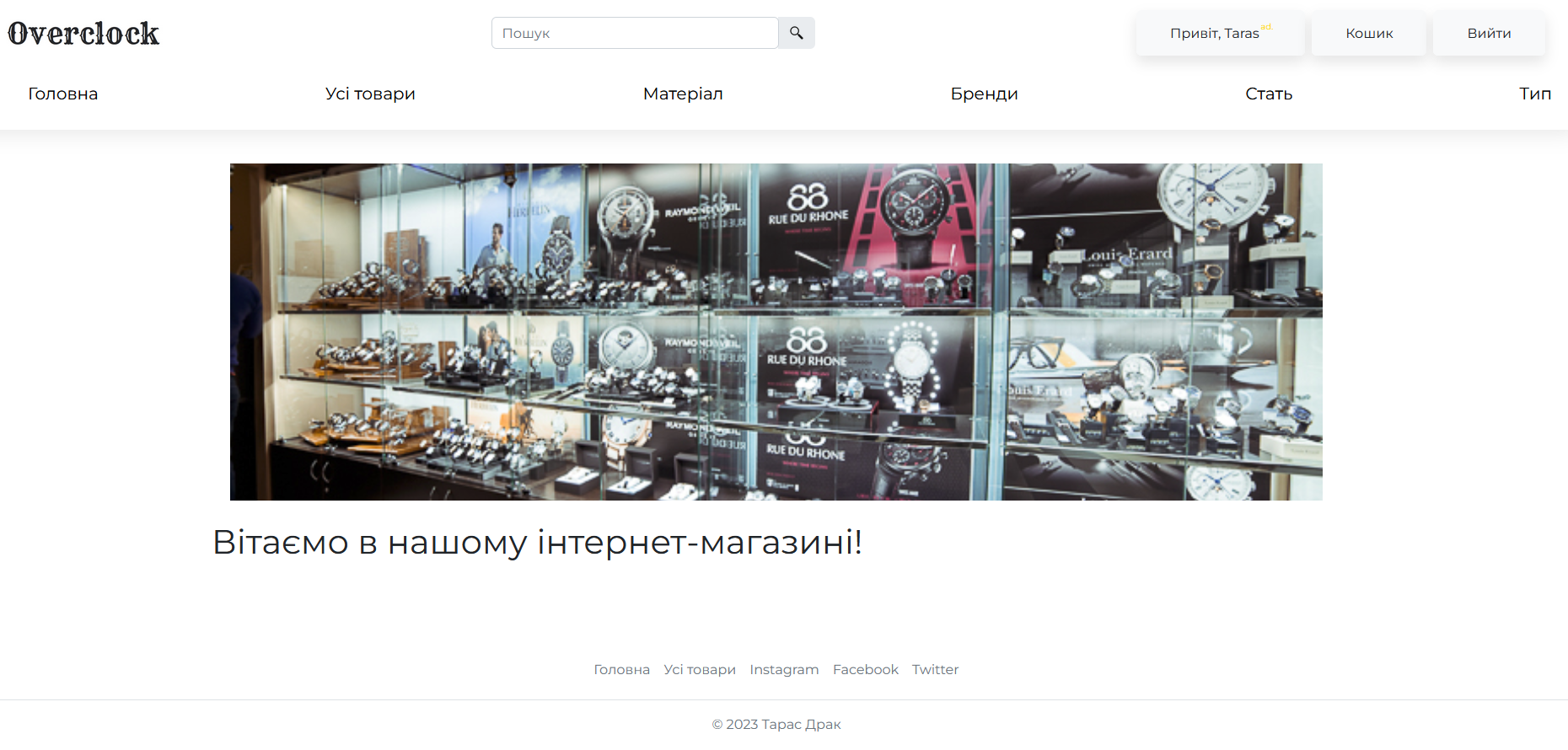


Рис. 2.12. Вікно зареєстрованого юзеру

Навідмінно від користувацького інтерфейсу тепер перегляд товарів сайту виглядає наступним чином.(Рис. 2.19.). З’явилася можливість редагувати, видаляти, та додавати продукти.

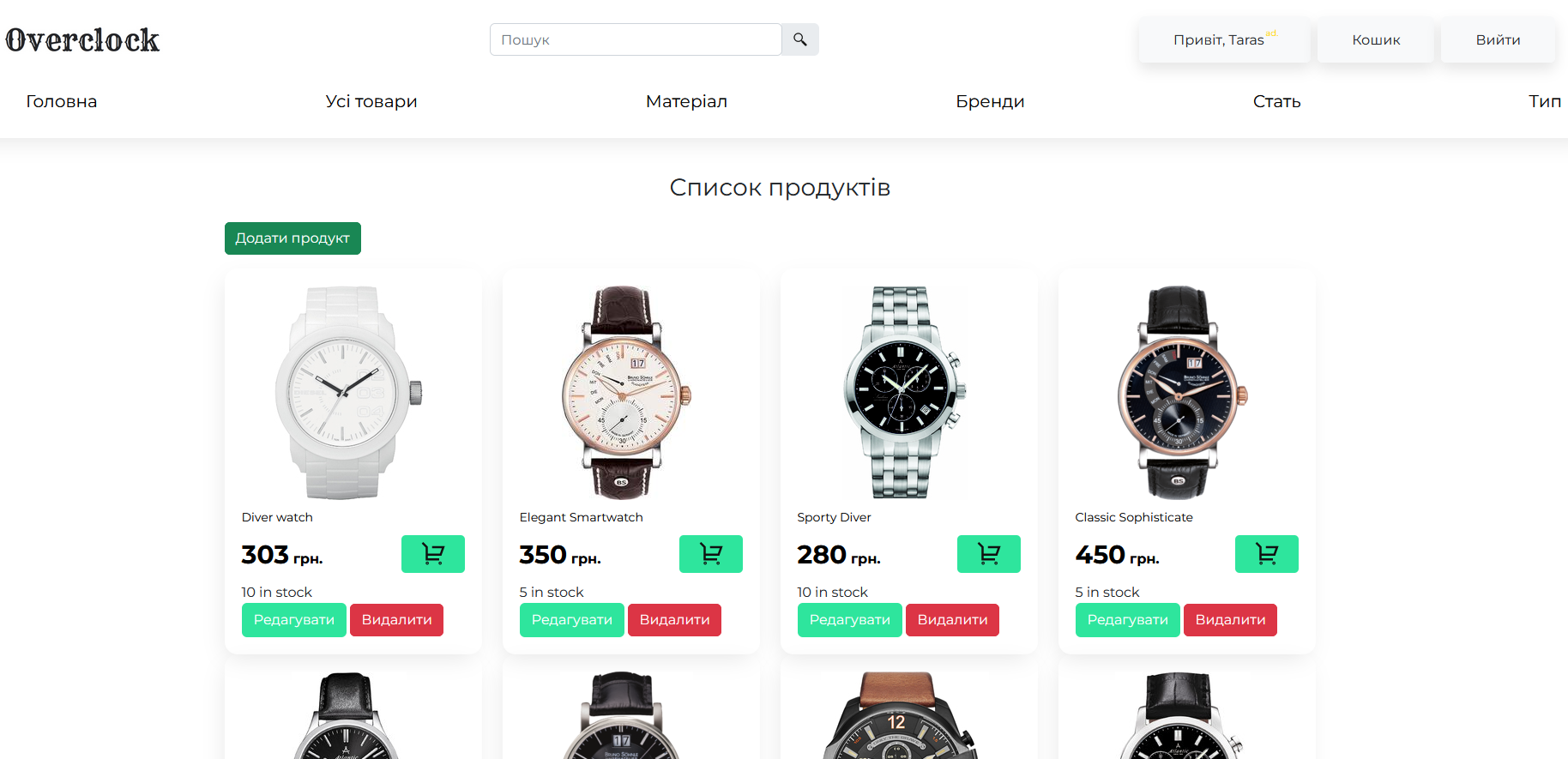


Рис. 2.13. Сторінка товарів адміністратора

Висновки: у ІС було оформлено інтерфейс, який містить поля вводу, кнопки, фотографії та ін. Також був задіяний JS та CSS з використанням фреймворка Bootstrap 5 та скриртів Fancybox.

# РОЗДІЛ 3 ОПИС РОБОТИ З WEB ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

# 3.1 Опис роботи з Web додатком (Опис інтерфейсу)

Запустити веб додаток можна за допомогою посилання «http:/store/»

Основна навігація по сайту здійснюється за допомогою меню зверху сторінки, завдяки ньому можна переходити на інші модулі.(Рис. 3.1.), та нижє меню сайта .(Рис. 3.2.)

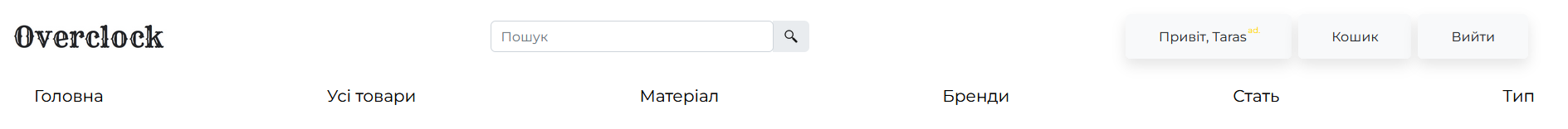


Рис. 3.1. Верхнє меню сайту

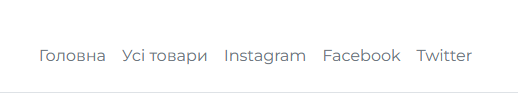


Рис. 3.2. Нижнє меню сайту

При натисканні в нижньому меня нас перекидує на інстаграм (Рис. 3.3.), фейсбук(Рис. 3.4.), та твітер (Рис. 3.5.).

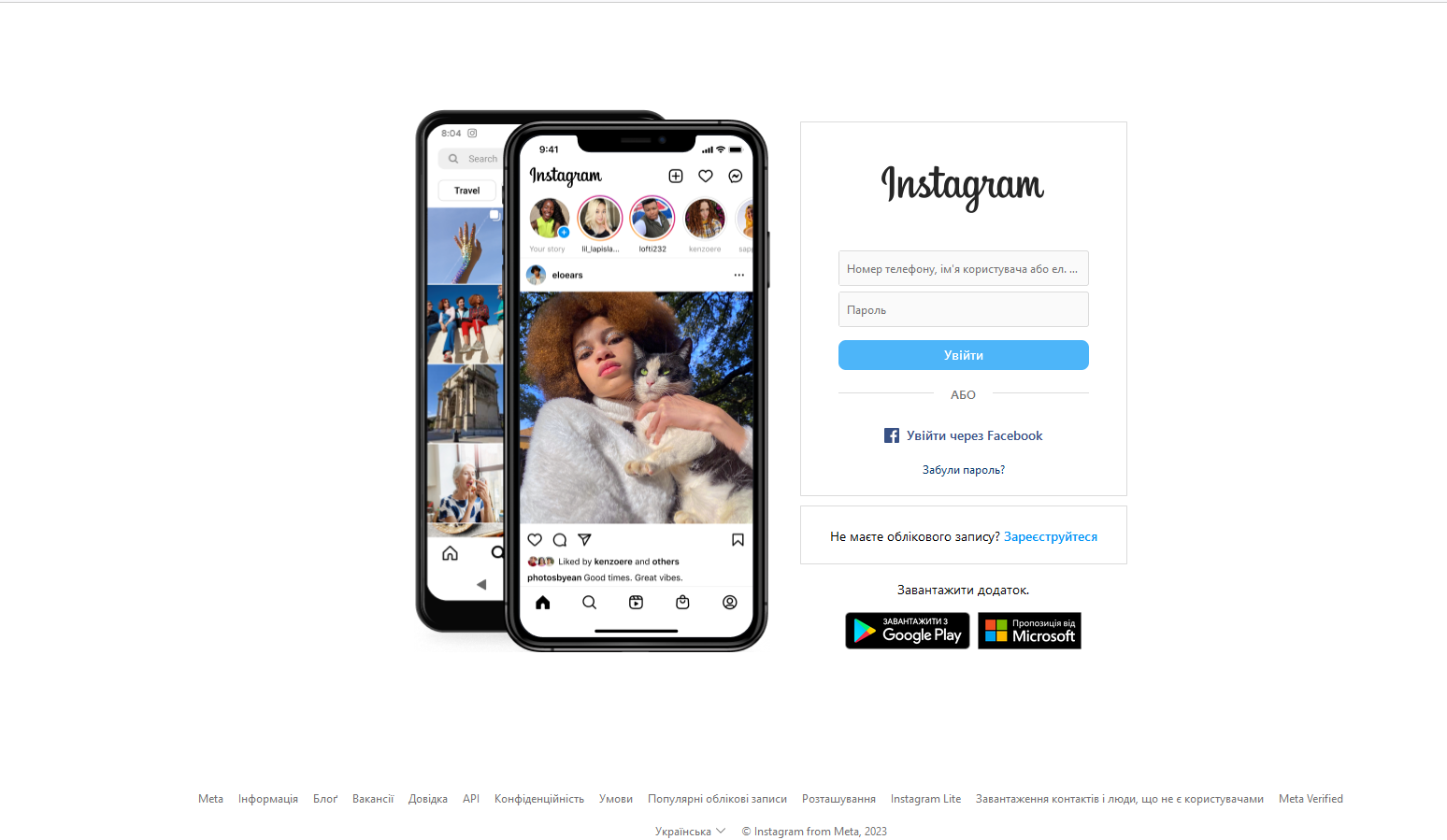


Рис. 3.3. Інстаграм

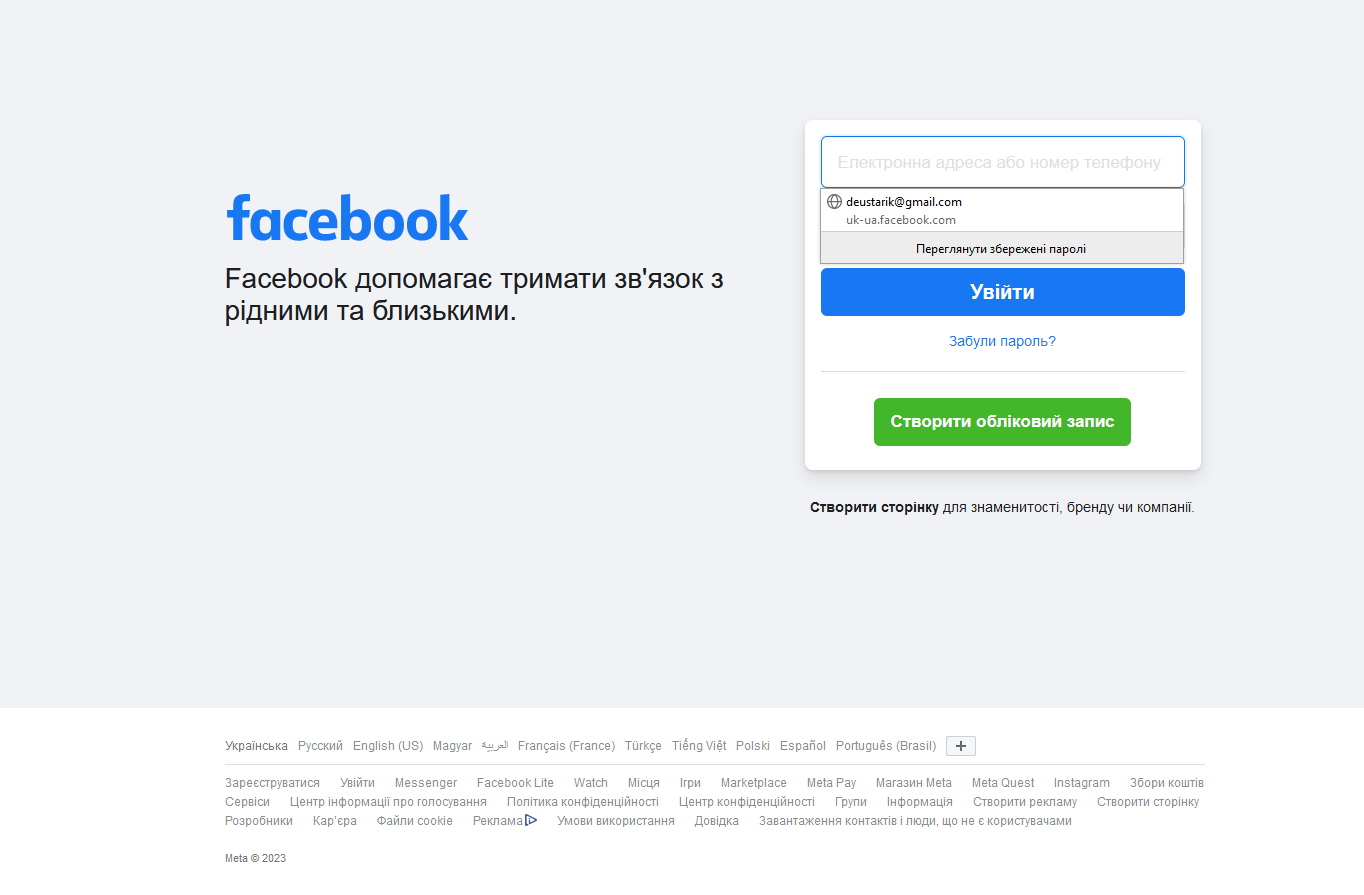


Рис. 3.4. Фейсбук

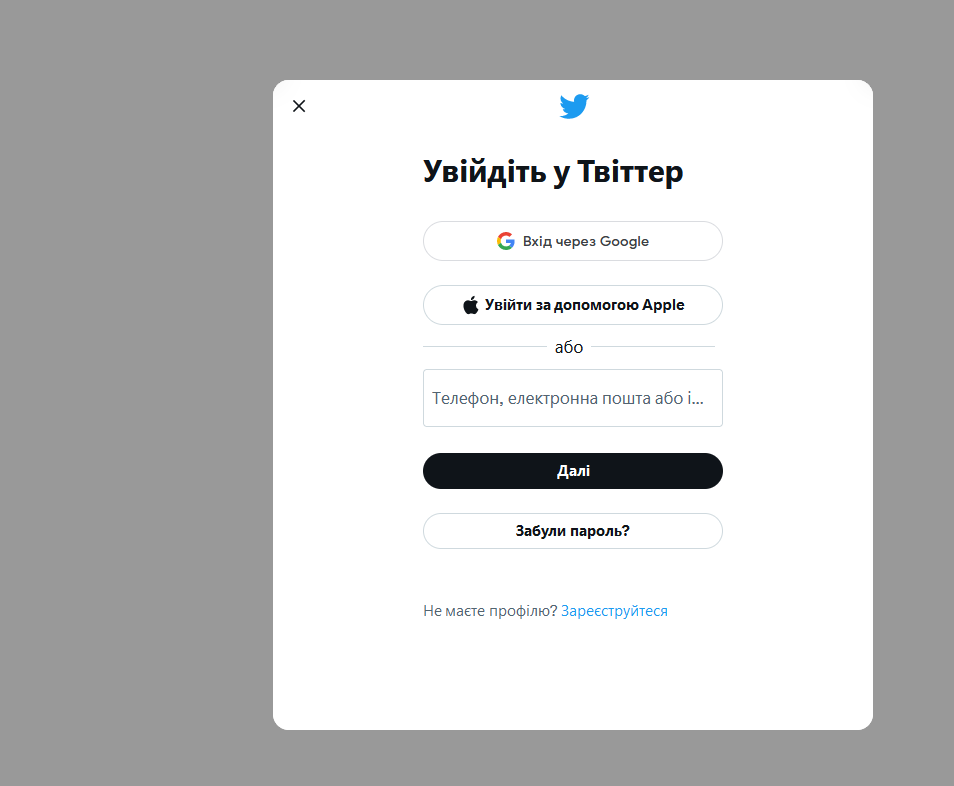


Рис. 3.5. Твітер

У верхньому меню та нижньому меню можна перейти на вкладку усіх товарів рисунок (Рис. 3.6.)

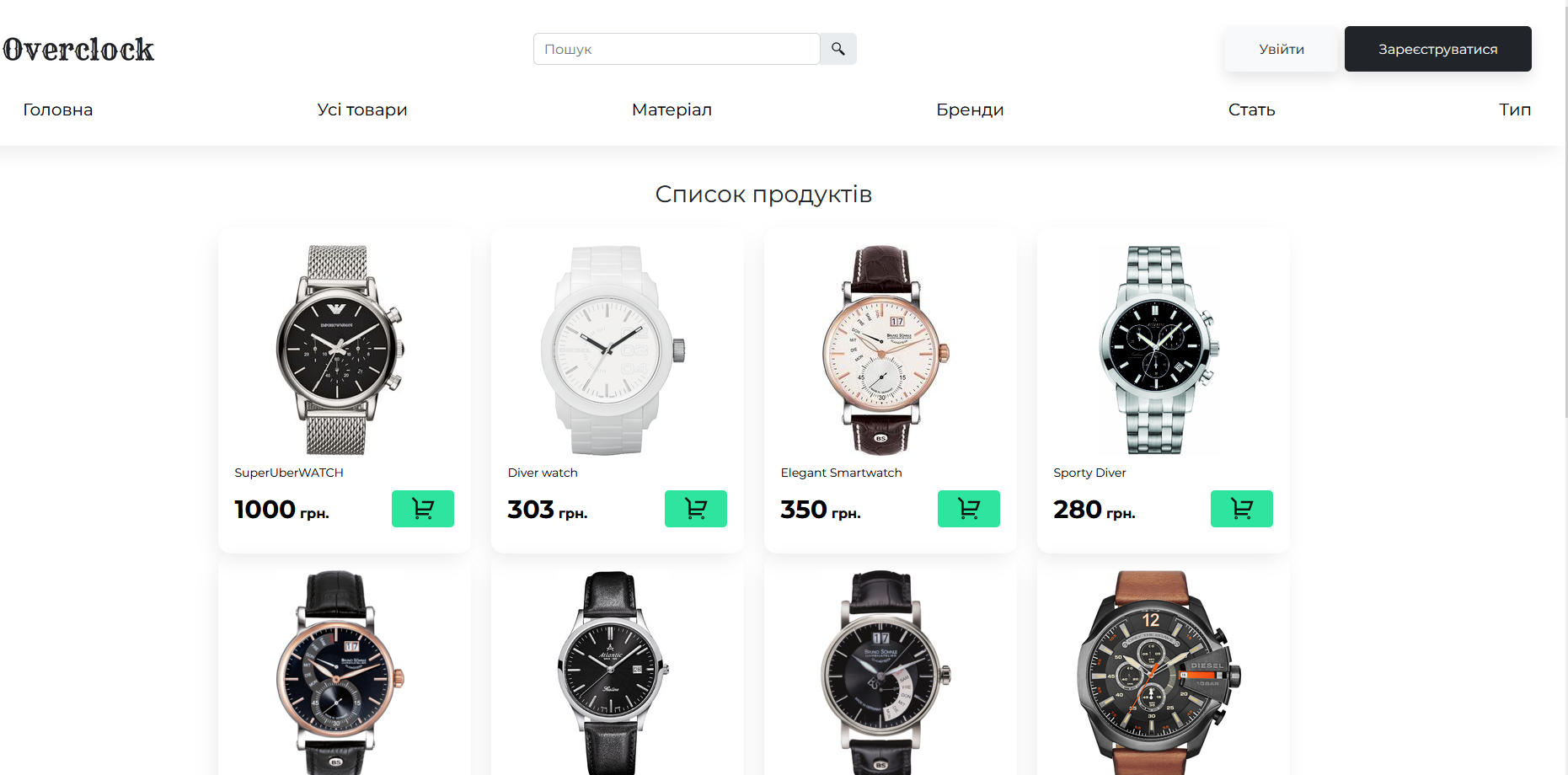


Рис. 3.6. Перелік усіх товарів

На верхній панелі є категорії пошуку які допоможуть направлено знайти необхідний продукт, або визначитися з бажаним товаром. Натиснувши на головну категорію у верхньому меню користувач побачить посилання на доступні підкатегорії

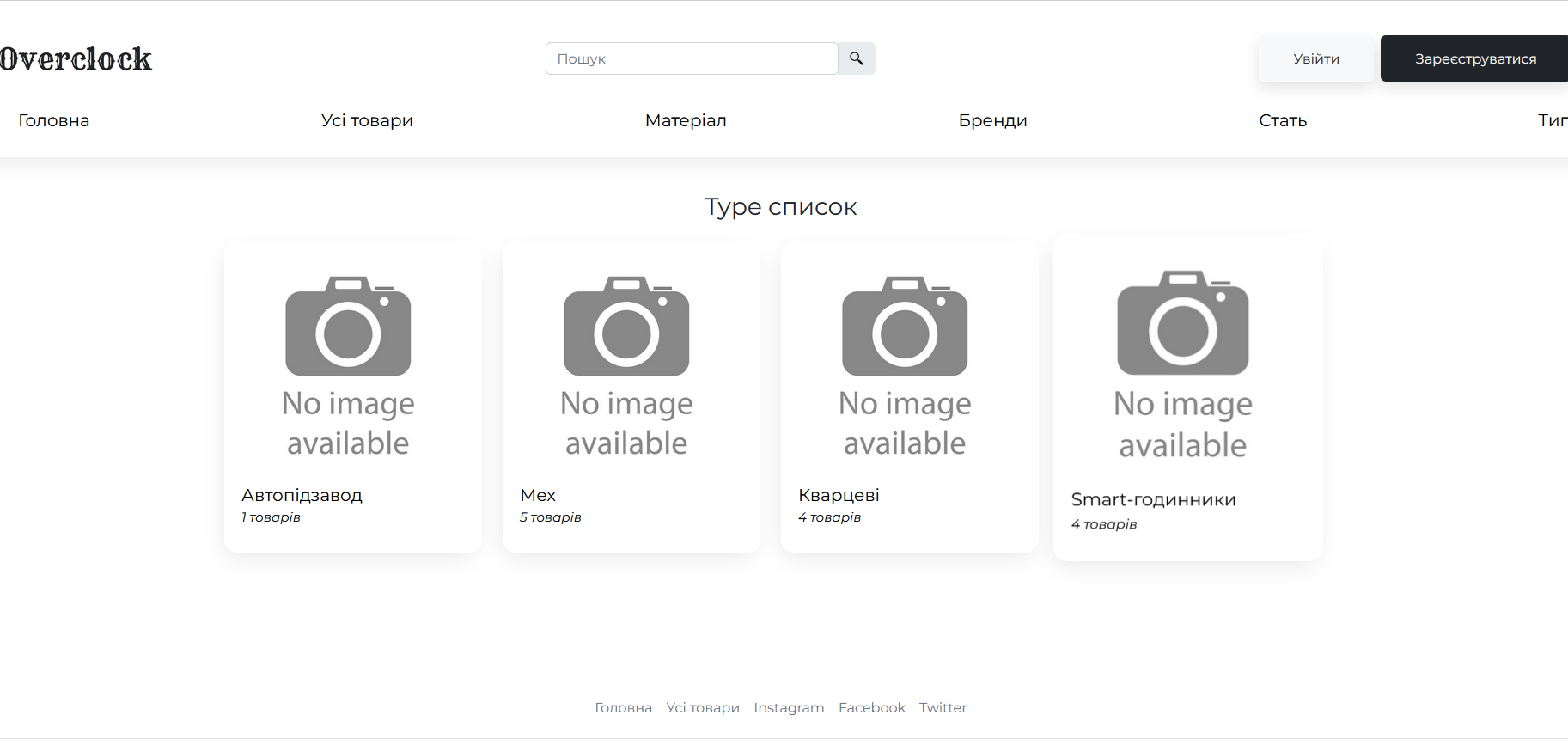


Рис. 3.7. Приклад взаємодії з типами

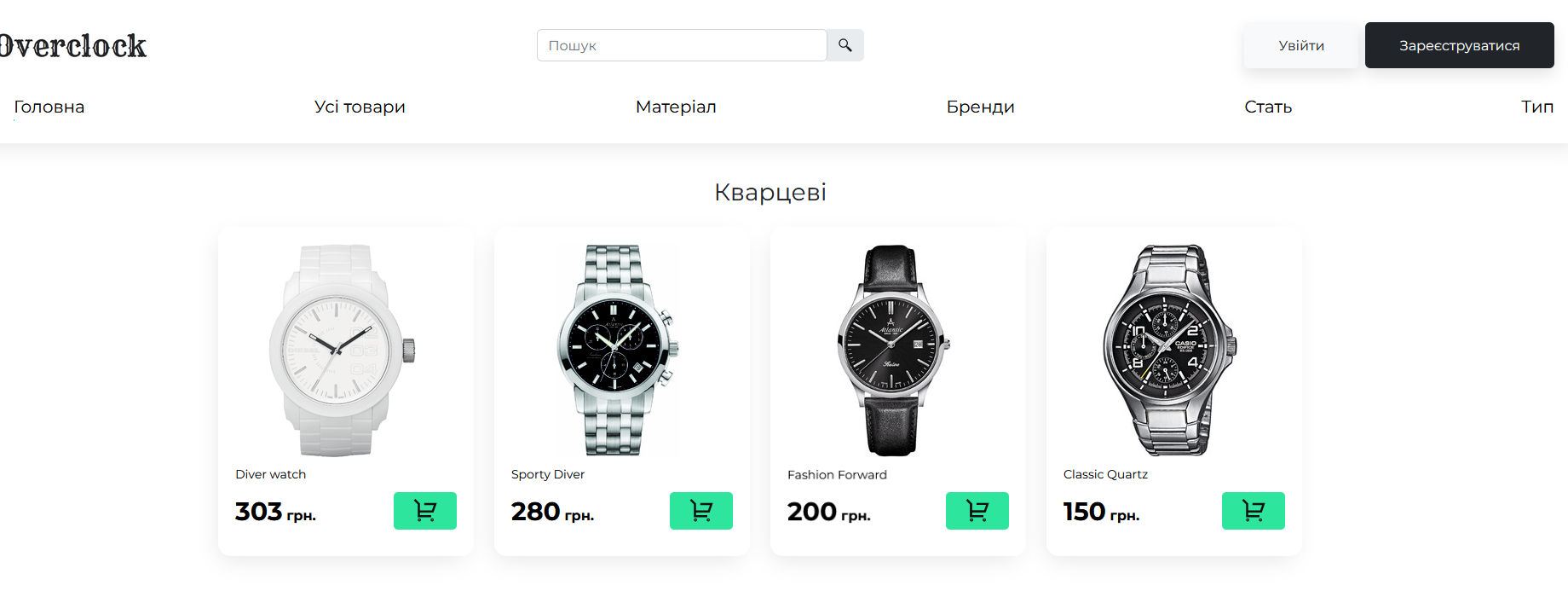


Рис. 3.8. Приклад взаємодії з типами

За допомогою пошуку можна знайти конкретний товар за повним чи частковим співпадінням (Рис. 3.9.).



Рис. 3.9. Сторінка пошуку товару

Як адміністратор добавимо новий бренд для годинників (Рис. 3.10.).

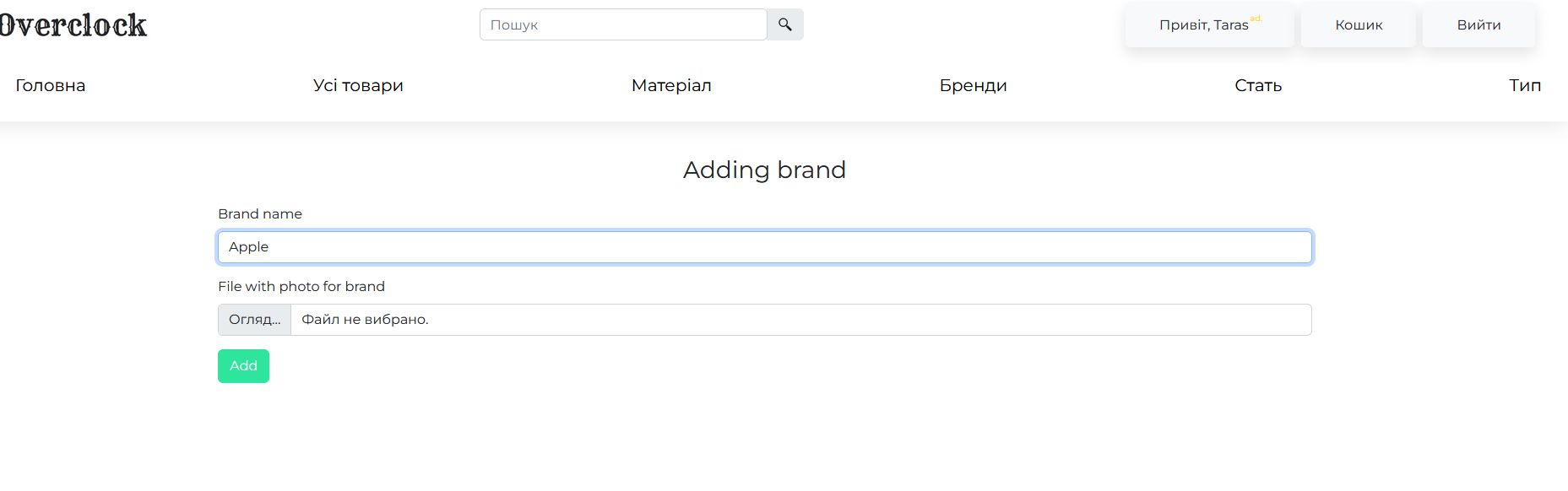


Рис. 3.11. Приклад додавання бренду

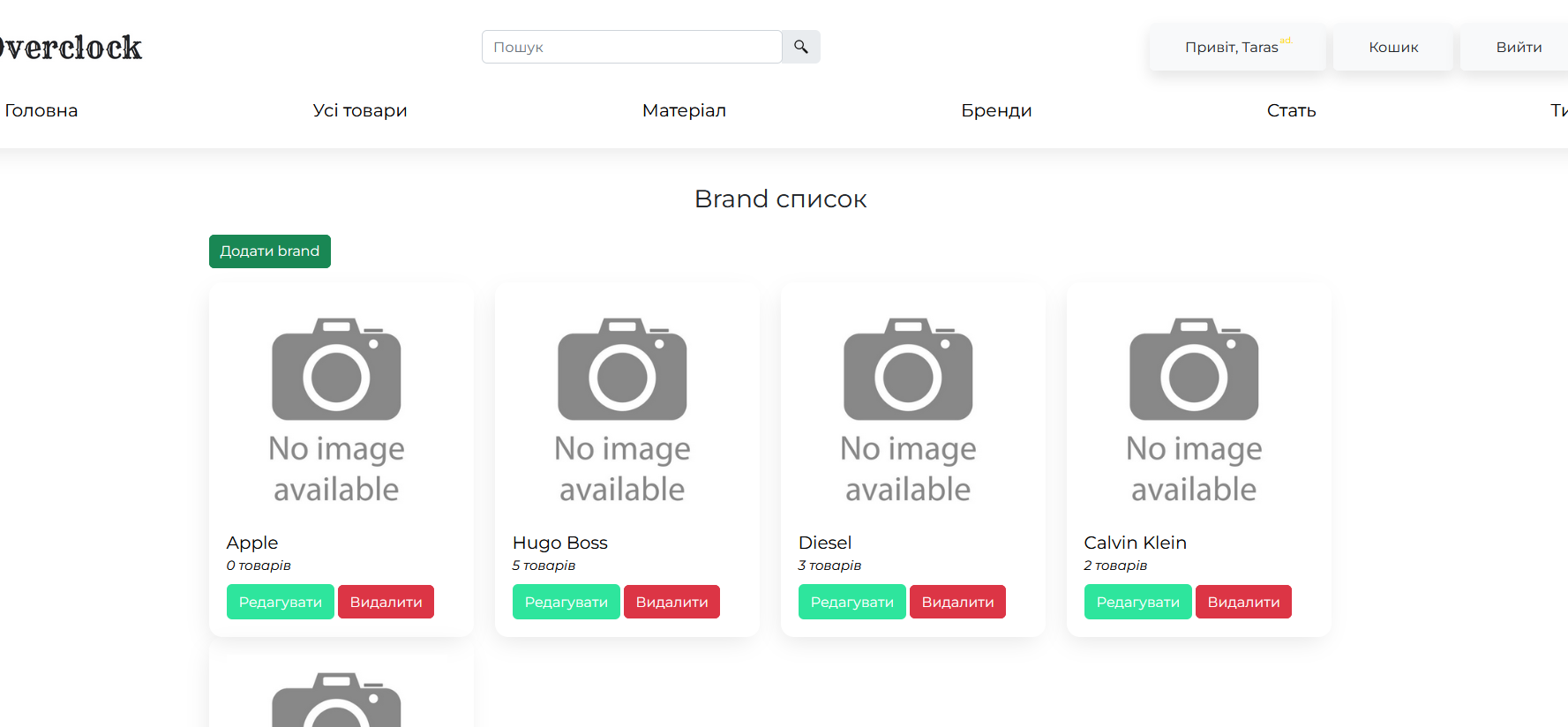


Рис. 3.11.2 Сторінка з новим брендом

Після додавання категорії "Apple" ми приступаємо до додавання товару (Рис. 3.12.).

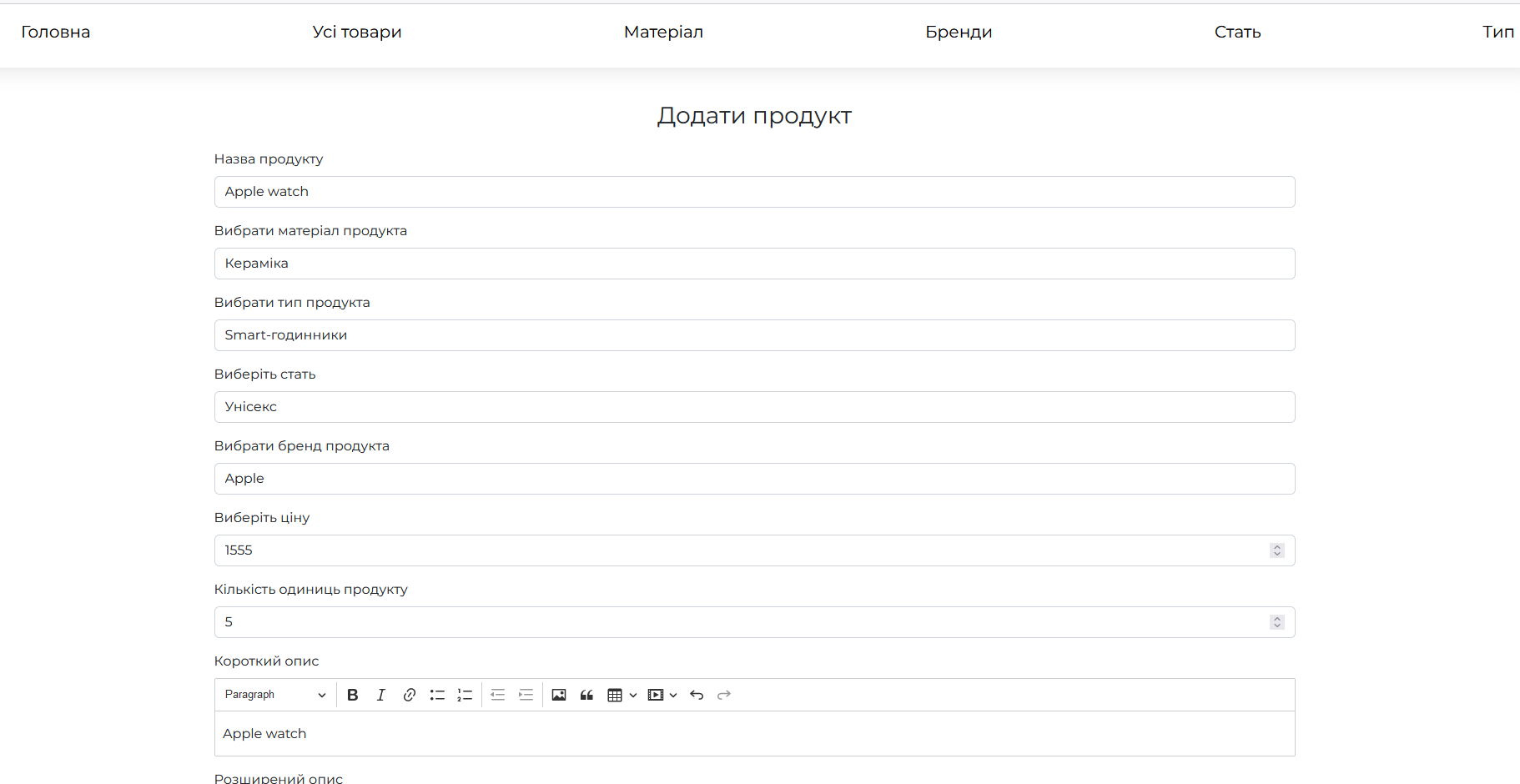


Рис. 3.13. Сторінка з додаванням нового продукту

Тепір у нас є нова категорія та в цій категорії знаходиться наш новий продукт (Рис. 3.14.).

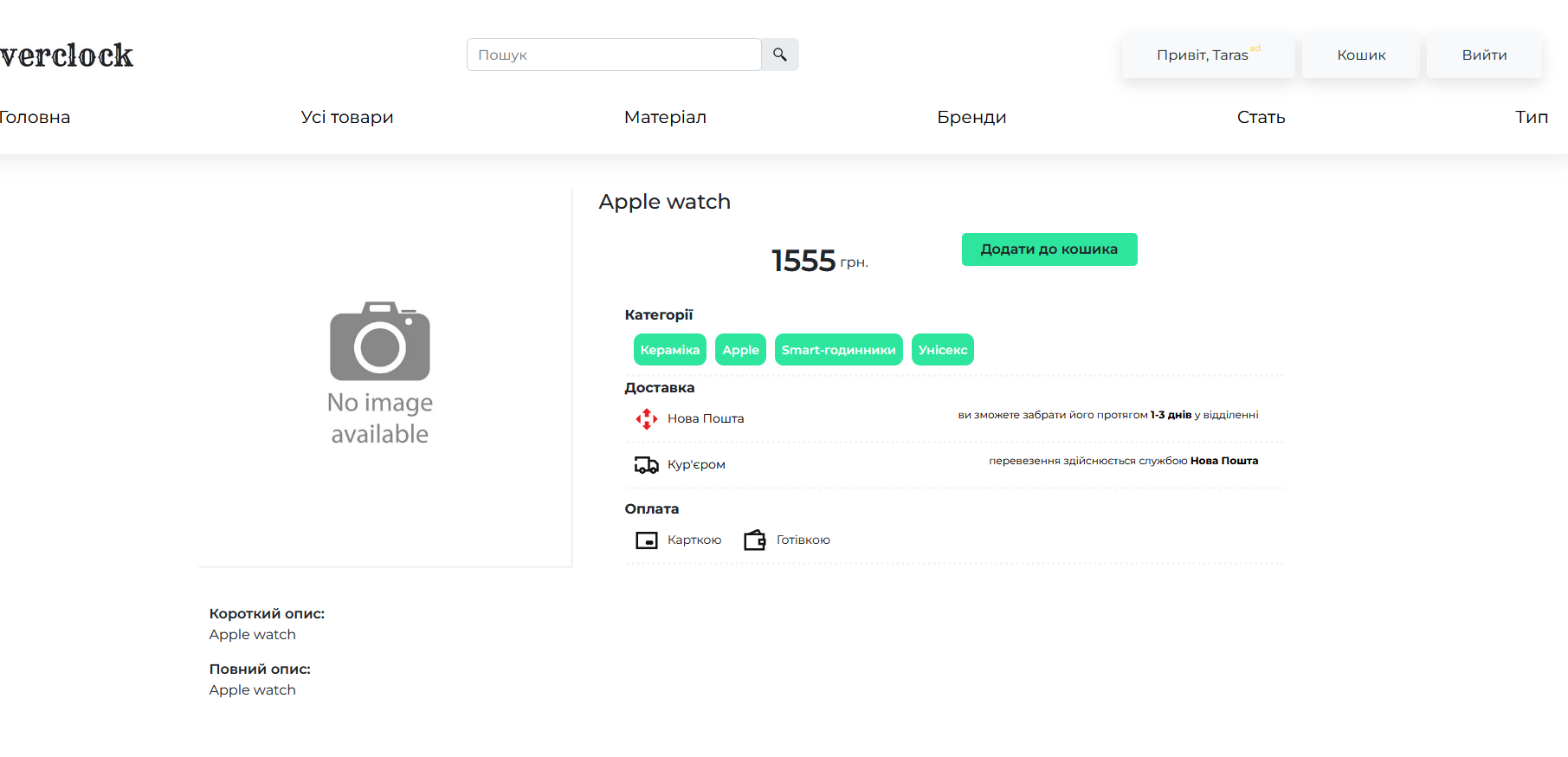


Рис. 3.15. Сторінка з новим продуктом

Тепір відредагуємо цей продукт та додамо до нього фотографію (Рис. 3.16.)

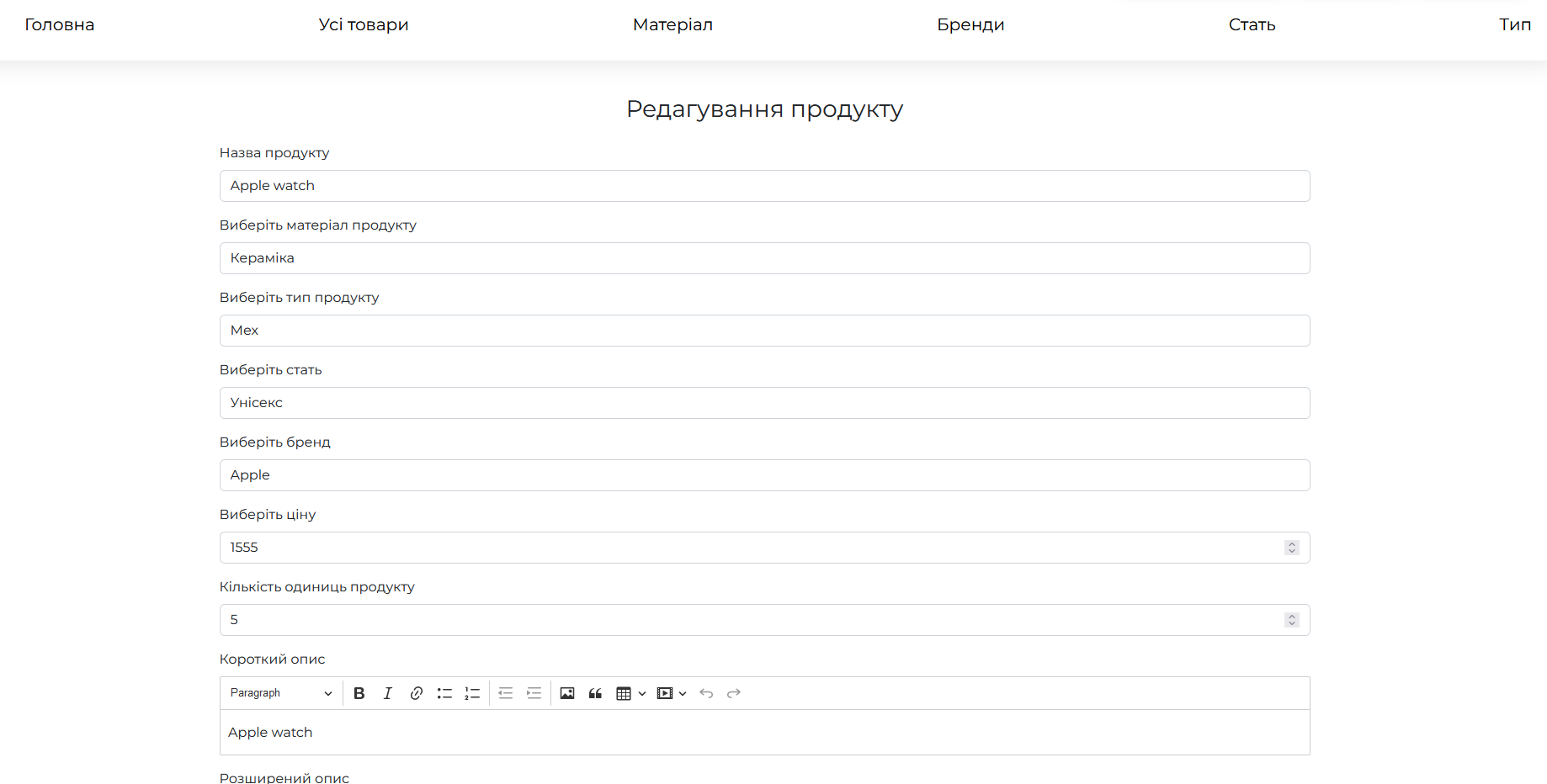


Рис. 3.17. Редагування продукту

Надалі ми можемо побачити добавлену фотографію та змінені данні на сторінці нового товару(Рис. 3.18).

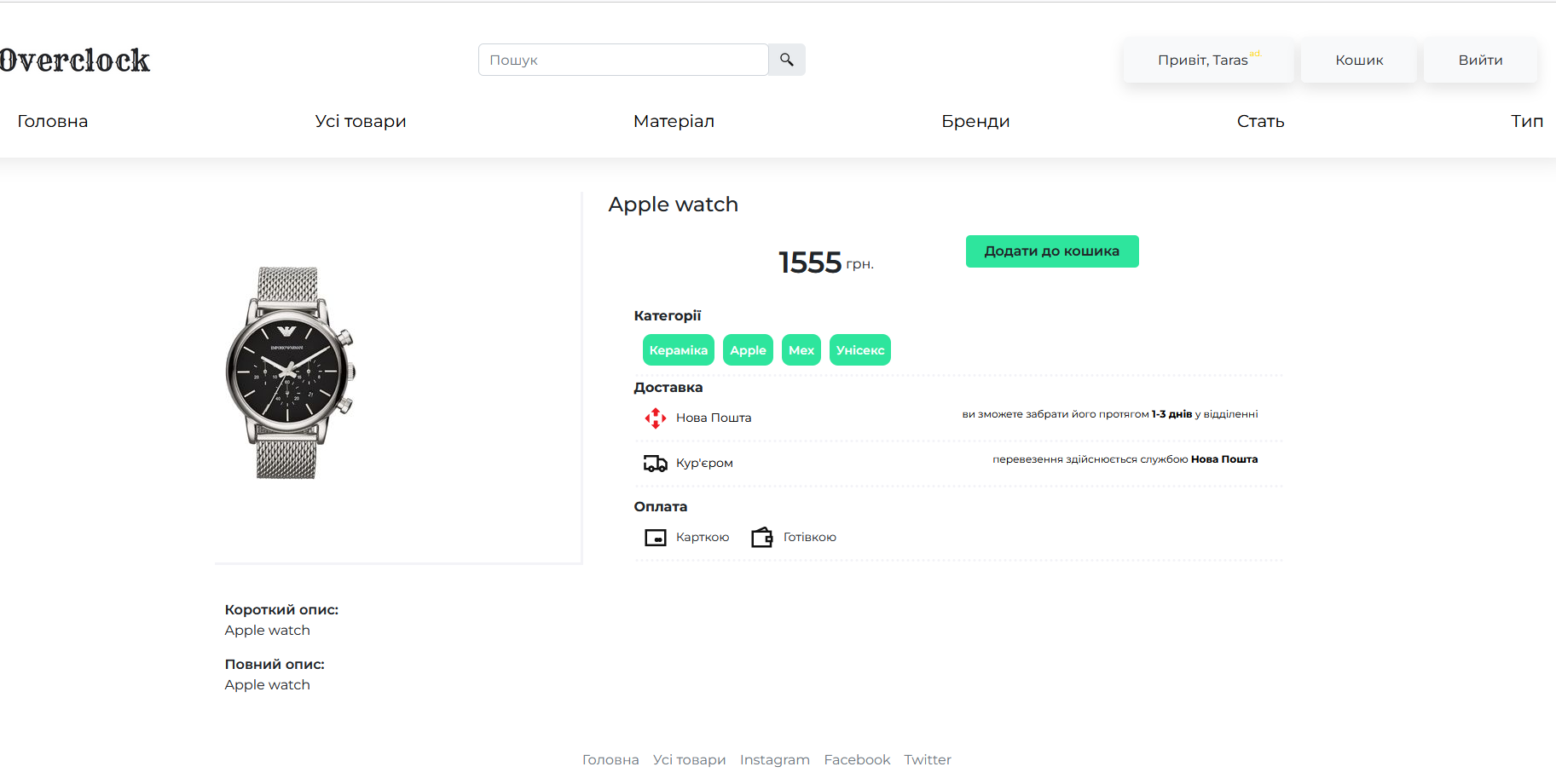


Рис 3.18. Відредагований товар

Далі перейдемо знайдемо наш товар у каталозі спробуємо його видалити та оновити сторінку(Рис. 3.19.).

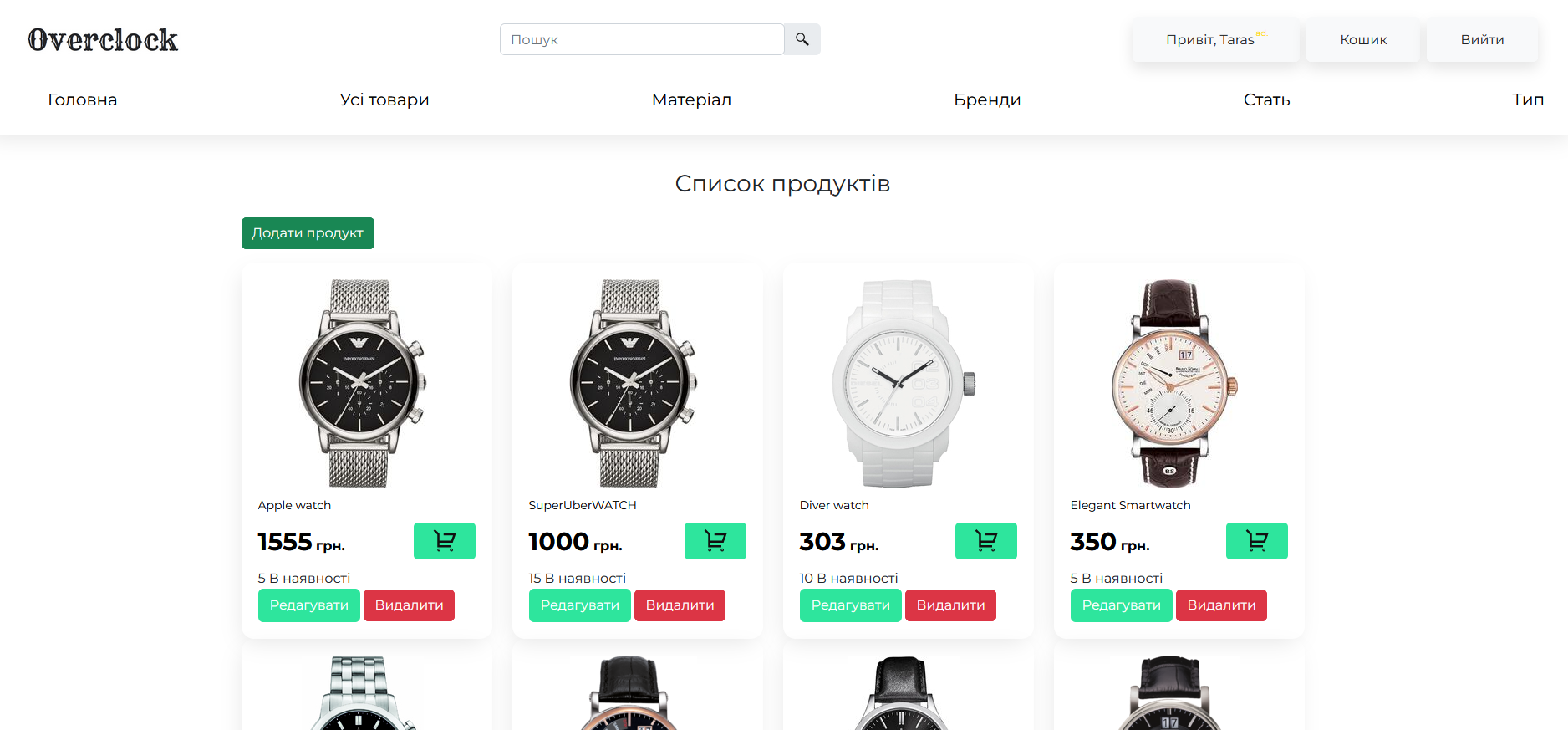


Рис. 3.19. Видалення нового товару

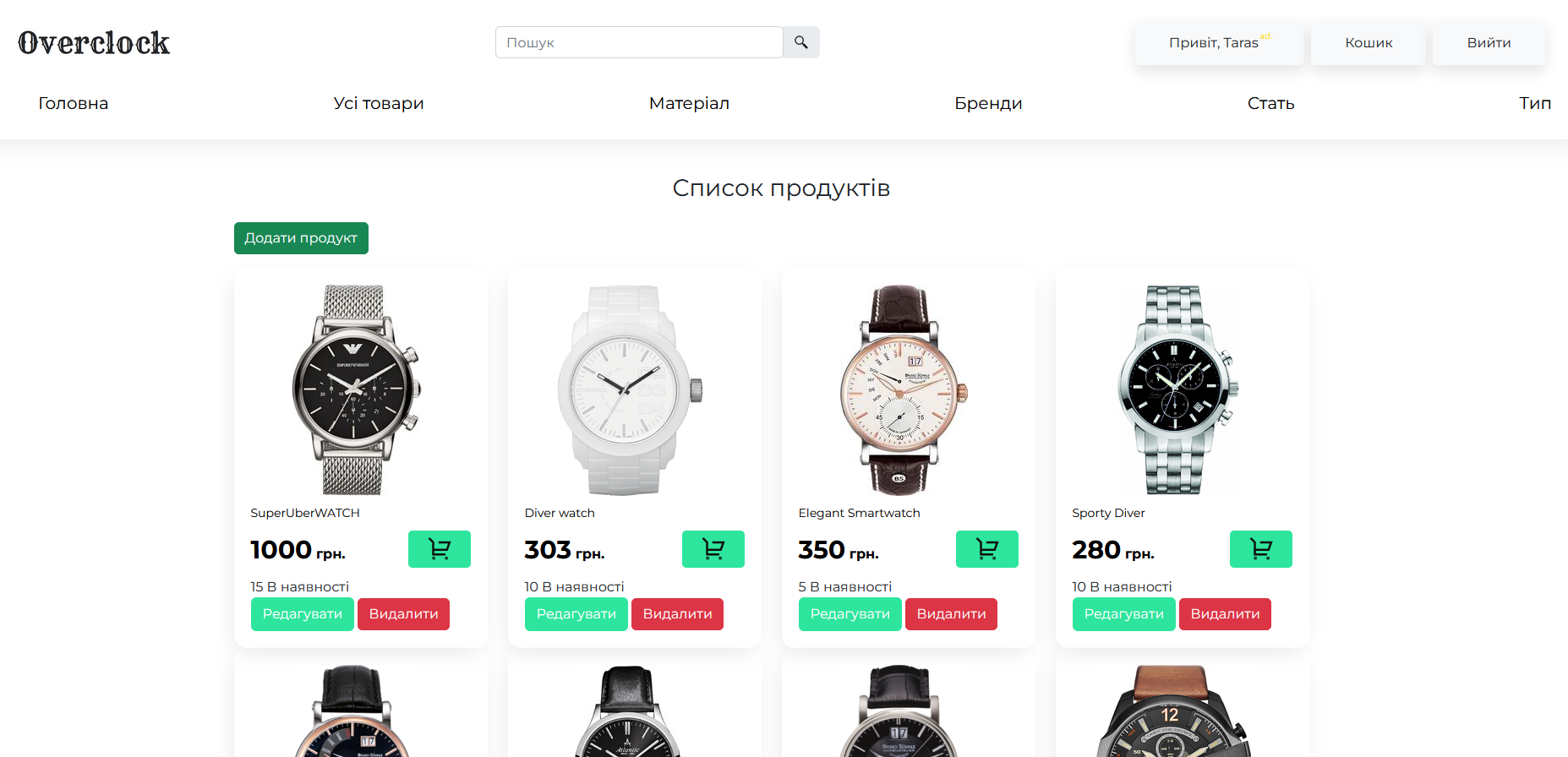


Рис. 3.19.2. Перезавантаження сторінки

# 3.2 Тестування роботи Web-додатку

У веб-додатку існують два механізми захисту, які забезпечують користувачеві повідомлення щодо некоректного вводу даних. Перший механізм виявляє, що введений користувачем логін вже використовується і повідомляє про це за допомогою повідомлення "Логін зайнятий". Другий механізм перевіряє, чи введений користувачем пароль співпадає з попередньо встановленим паролем і, у разі невідповідності, виводить повідомлення "Пароль не співпадає" (Рис. 3.20.).

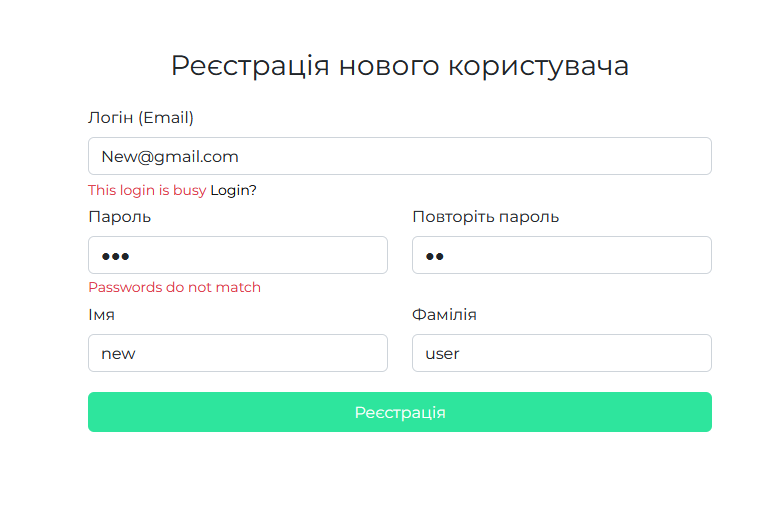


Рис. 3.20. Помилка вводу даних

WEB-додадок передбачає помилки при некоректних діях адміністратора. Виводяться попереджувальні сповіщення що ціна не може бути від’ємною.(Рис. 3.21.)

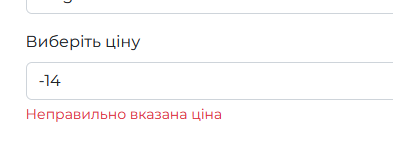


Рис. 3.21. Помилка вводу даних

Якщо ж користувач вирішить перейти на неіснуючу сторінку, він отримає помилку 404(Рис.3.22.)

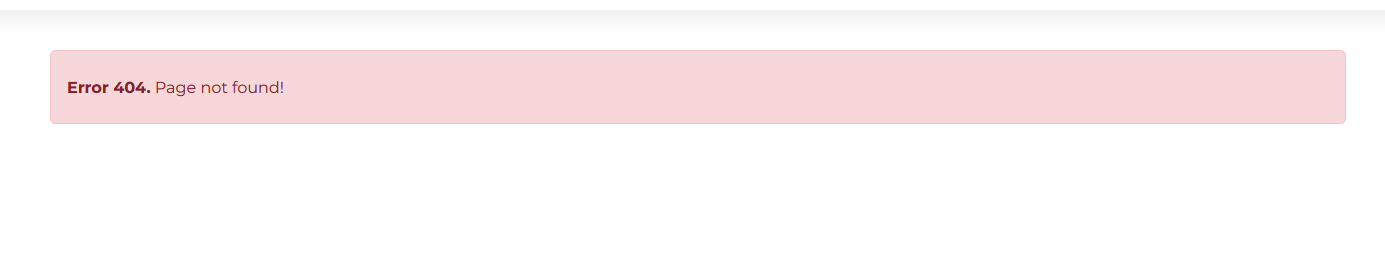


Рис. 3.22. Помилка вводу даних

Висновки: данний магазин готовий до користування іншими користувачами, проведена робота з помилками у додатку.

# ВИСНОВКИ

Отже, переддипломна практика проходила у ФОП Сидорчук протягом 4 тижнів, з 09.01 по 05.02. На початку проходження практики було отримано технічне завдання на розробку програмного продукту. Було виокремлено та проаналізовано вимоги до програмної системи.

Далі було розроблено модель системи веб-додатку. Було побудовано діаграму активності користувача і адміністратора для демонстрації алгоритму роботи додатку та можливих станів системи. Також була побудовані колекції для зображення зв’язків між моделями баз данних.

Далі була реалізована серверна частина інтернет магазину. Було протестовано коректність роботи системи і виправлено виявлені помилки.

Було створено фундаментальну частину інтернет магазину для подальшої розробки програмного продукту.

Отже, під час проходження переддипломної практики було успішно частина інтернет магазину з продажі годинників "Overclock".

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PHP: Hypertext Preprocessor [Електронний ресурс] – Режим доступу до ресурсу: [https://www.php.net/.](https://www.php.net/)
2. w3schools.com [Електронний ресурс] – Режим доступу до ресурсу: 4. <https://www.w3schools.com/php/default.asp>.
3. Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов / Давид Скляр. – Москва: Диалектика-Вильямс, 2020. – 464 с. – (ISBN; кн. 978).
4. Lockhart J. Modern PHP: New Features and Good Practices 1st Edition / Josh Lockhart., 2015. – 270 с.
5. Стефанов С. КНИГА JAVASCRIPT. ШАБЛОНЫ / Стоян Стефанов..
6. Янк К. PHP & MySQL: Novice to Ninja / К. Янк, Т. Батлер., 2017.
7. Пауерс Д. PHP 7 Solutions: Dynamic Web Design Made Easy / Девід Пауерс., 2019.
8. Haverbeke M. Виразний JavaScript: Сучасне введення в програмування / Marijn Haverbeke..

# ДОДАТОК

**Програмний код**

**Core.php**

<?php  
  
namespace core;  
  
use controllers\MainController;  
  
class Core  
{  
 private static *$instance* = null;  
 public $app;  
 public $pageParams;  
 public $db;  
 public $requestMethod;  
 private function \_\_construct() {  
 global $pageParams;  
 $this->app = [];  
 $this->pageParams = $pageParams;  
 }  
 public static function getInstance() {  
 if(empty(self::*$instance*)) {  
 self::*$instance* = new self();  
 }  
 return self::*$instance*;  
 }  
 public function Initialize() {  
 session\_start();  
 $this->db = new DB(*DATABASE\_HOST*, *DATABASE\_LOGIN*,  
 *DATABASE\_PASSWORD*, *DATABASE\_BASENAME*);  
 $this->requestMethod = $\_SERVER['REQUEST\_METHOD'];  
 }  
 public function Run() {  
 $route = $\_GET['route'];  
 $routeParts = explode('/', $route);  
 $moduleName = strtolower(array\_shift($routeParts));  
 $actionName = strtolower(array\_shift($routeParts));  
 if(empty($moduleName))  
 $moduleName = "main";  
 if(empty($actionName))  
 $actionName = "index";  
 $this->app['moduleName'] = $moduleName;  
 $this->app['actionName'] = $actionName;  
 $controllerName = '\\controllers\\'.ucfirst($moduleName).'Controller';  
 $controllerActionName = $actionName.'Action';  
 $statusCode = 200;  
 if(class\_exists($controllerName)) {  
 $controller = new $controllerName();  
  
 if (method\_exists($controller, $controllerActionName)) {  
 $actionResult = $controller->$controllerActionName($routeParts);  
 if($actionResult instanceof Error)  
 $statusCode = $actionResult->code;  
 $this->pageParams['content'] = $actionResult;  
 }  
 else {  
 $statusCode = 404;  
 }  
 } else {  
 $statusCode = 404;  
 }  
 $statusCodeType = intval($statusCode / 100);  
 if($statusCodeType == 4 || $statusCodeType == 5) {  
 $mainController = new MainController();  
 $this->pageParams['content'] = $mainController->errorAction($statusCode);  
 }  
 }  
 public function Done() {  
 $pathToLayout = 'themes/light/layout.php';  
 $tpl = new Template($pathToLayout);  
 $tpl->setParams($this->pageParams);  
 $html = $tpl->getHTML();  
 echo $html;  
 }  
}

**Controller.php**

<?php  
  
  
namespace core;  
  
  
class Controller  
{  
 protected $viewPath;  
 protected $moduleName;  
 protected $actionName;  
 public function \_\_construct() {  
 $this->moduleName = \core\Core::*getInstance*()->app['moduleName'];  
 $this->actionName = \core\Core::*getInstance*()->app['actionName'];  
 $this->viewPath = "views/{$this->moduleName}/{$this->actionName}.php";  
 }  
 public function render($viewPath = null, $params = null) {  
 if(empty($viewPath))  
 $viewPath = $this->viewPath;  
 $tpl = new Template($viewPath);  
 if(!empty($params))  
 $tpl->setParams($params);  
 return $tpl->getHTML();  
 }  
 public function error($code, $message = null) {  
 return new Error($code, $message);  
 }  
 public function redirect($url) {  
 header("Location: {$url}");  
 die;  
 }  
 public function renderView($viewName) {  
 $path = "views/{$this->moduleName}/{$viewName}.php";  
 $tpl = new Template($path);  
 if(!empty($params))  
 $tpl->setParams($params);  
 return $tpl->getHTML();  
 }  
}

**Template.php**

<?php  
  
  
namespace core;  
  
  
class Template  
{  
 protected $path;  
 protected $params;  
 public function \_\_construct($path)  
 {  
 $this->path = $path;  
 $this->params = [];  
 }  
 public function setParam($name, $value) {  
 $this->params[$name] = $value;  
 }  
 public function setParams($params){  
 foreach ($params as $name => $value)  
 $this->setParam($name, $value);  
 }  
 public function getHTML()  
 {  
 ob\_start();  
 extract($this->params);  
 include($this->path);  
 $html = ob\_get\_contents();  
 ob\_end\_clean();  
 return $html;  
 }  
}

**Db.php**

<?php  
  
  
namespace core;  
  
*/\*\*  
 \* Клас для виконання запитів до БД  
 \*/*class DB  
{  
 protected $pdo;  
 public function \_\_construct($hostname, $login, $password, $database){  
 $this->pdo = new \PDO("mysql: host={$hostname};dbname={$database}", $login, $password);  
 }  
  
 public function select($tableName, $fieldsList = "\*", $conditionArray = null, $operatorName = 'LIKE') {  
 if(is\_string($fieldsList))  
 $fieldsListString = $fieldsList;  
 if(is\_array($fieldsList))  
 $fieldsListString = implode(', ', $fieldsList);  
 $wherePartString = "";  
 if(is\_array($conditionArray)) {  
 $parts = [];  
 foreach ($conditionArray as $key => $value) {  
 $parts[]="{$key} {$operatorName} :{$key}";  
 }  
 if(is\_string($wherePartString))  
 $wherePartString = "WHERE ".implode(' AND ', $parts);  
 }  
 $res = $this->pdo->prepare(  
 "SELECT {$fieldsListString} FROM {$tableName} {$wherePartString}");  
 $res->execute($conditionArray);  
 return $res->fetchAll(\PDO::*FETCH\_ASSOC*);  
 }  
  
 public function update($tableName, $newValuesArray, $conditionArray){  
 $setParts = [];  
 $paramsArray = [];  
 foreach ($newValuesArray as $key => $value) {  
 $setParts [] = "{$key} = :set{$key}";  
 $paramsArray['set'.$key] = $value;  
 }  
 $setPartString = implode(', ', $setParts);  
  
 $whereParts = [];  
 foreach ($conditionArray as $key => $value) {  
 $whereParts []="{$key} = :{$key}";  
 $paramsArray[$key] = $value;  
 }  
 $wherePartString = "WHERE ".implode(' AND ', $whereParts);  
 $res = $this->pdo->prepare("UPDATE {$tableName} SET {$setPartString} {$wherePartString}");  
 return $res->execute($paramsArray);  
 }  
  
 public function insert($tableName, $newRowArray) {  
 $fieldsArray = array\_keys($newRowArray);  
 $fieldsListString = implode(', ', $fieldsArray);  
 $paramsArray = [];  
 foreach ($newRowArray as $key => $value) {  
 $paramsArray [] = ':'.$key;  
 }  
 $valuesListString = implode(', ', $paramsArray);  
 $res = $this->pdo->prepare("INSERT INTO {$tableName} ($fieldsListString) VALUES($valuesListString)");  
 $res->execute($newRowArray);  
 }  
  
 public function delete($tableName, $conditionArray) {  
 $whereParts = [];  
 foreach ($conditionArray as $key => $value) {  
 $whereParts []="{$key} = :{$key}";  
 $paramsArray[$key] = $value;  
 }  
 $wherePartString = "WHERE ".implode(' AND ', $whereParts);  
 $res = $this->pdo->prepare("DELETE FROM {$tableName} {$wherePartString}");  
 $res->execute($conditionArray);  
 }  
  
}

**Product.php**

<?php  
namespace models;  
  
use core\Core;  
use utils\Filter;  
  
class Product  
{  
 protected static *$tableName* = 'product';  
 public static function add($row, $filePath) {  
 $fieldsList = ['name', 'photo', 'gender\_id', 'type\_id', 'material\_id',  
 'brand\_id', 'price', 'count', 'short\_description', 'description', 'visible'];  
 $row = Filter::*filterArray*($row, $fieldsList);  
 $row += ['photo' => $filePath];  
 Core::*getInstance*()->db->insert(self::*$tableName*, $row);  
 }  
  
 public static function delete($id) {  
 Core::*getInstance*()->db->delete(self::*$tableName*, [  
 'id' => $id  
 ]);  
 }  
 public static function update($id, $row) {  
 $fieldaslist = ['name', 'photo', 'gender\_id', 'type\_id', 'material\_id',  
 'brand\_id', 'price', 'count', 'short\_description', 'description', 'visible'];  
 $row = Filter::*filterArray*($row, $fieldaslist);  
 Core::*getInstance*()->db->update(self::*$tableName*, $row, [  
 'id' => $id  
 ]);  
 }  
 public static function getAllProducts() {  
 $rows = Core::*getInstance*()->db->select(self::*$tableName*, '\*');  
 return array\_reverse($rows);  
 }  
 public static function count() {  
 return count(self::*getAllProducts*());  
 }  
 public static function getOnePage($rows, $currentPage, $count) {  
 $startItem = ($currentPage - 1) \* $count;  
 $pageItems = array\_slice($rows, $startItem, $count);  
 return $pageItems;  
 }  
 public static function getById($id) {  
 $row = Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 'id' => $id  
 ]);  
 if(!empty($row))  
 return $row[0];  
 else  
 return null;  
 }  
 public static function getProductsIn($tableName, $id) {  
 $rows = Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 $tableName.'\_id' => $id  
 ]);  
 return array\_reverse($rows);  
 }  
 public static function setDefaultForNull($column, $id) {  
 $row = self::*getById*($id);  
 $row["{$column}\_id"] = 1;  
 self::*update*($row['id'], $row);  
 }  
 public static function search($searchString, $all) {  
 $conditionsArray = [  
 'name' => "%{$searchString}%"  
 ];  
 if(!$all)  
 $conditionsArray += ['visible' => 1];  
 $rows = Core::*getInstance*()->db->select(self::*$tableName*, '\*', $conditionsArray, 'LIKE');  
 return array\_reverse($rows);  
 }  
}

**User.php**

<?php  
namespace models;  
  
use core\Utils;  
  
class User  
{  
 protected static *$tableName* = 'user';  
 public static function addUser($login, $password, $firstname, $lastname) {  
 \core\Core::*getInstance*()->db->insert(  
 self::*$tableName*, [  
 'login' => $login,  
 'password' => self::*hashPassword*($password),  
 'firstname' => $firstname,  
 'lastname' => $lastname  
 ]  
 );  
 }  
 public static function hashPassword($password) {  
 return md5($password);  
 }  
 public static function isLoginExists($login) {  
 $user = \core\Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 'login' => $login  
 ]);  
 return !empty($user);  
 }  
 public static function verifyUser($login, $password) {  
 $user = \core\Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 'login' => $login,  
 'password' => $password  
 ]);  
 return !empty($user);  
 }  
 public static function getUserByLoginAndPassword($login, $password) {  
 $user = \core\Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 'login' => $login,  
 'password' => self::*hashPassword*($password)  
 ]);  
 if(!empty($user))  
 return $user[0];  
 return null;  
 }  
 public static function authenticateUser($user) {  
 $\_SESSION['user'] = $user;  
 }  
 public static function logoutUser() {  
 unset($\_SESSION['user']);  
 }  
 public static function isUserAuthenticated() {  
 return isset($\_SESSION['user']);  
 }  
 public static function getCurrentAuthenticatedUser() {  
 return $\_SESSION['user'];  
 }  
 public static function isAdmin() {  
 $user = self::*getCurrentAuthenticatedUser*();  
 return $user['access\_level'] == 10;  
 }  
}

**Order.php**

<?php  
  
  
namespace models;  
  
  
use core\Core;  
use utils\Filter;  
  
class Order  
{  
 protected static *$tableName* = 'user\_order';  
 public static function addOrder($row) {  
 $products = $row['products'];  
  
 $fieldsList = ['date', 'user\_id', 'name',  
 'phone', 'city', 'post\_number', 'total\_price'];  
 $rowOrder = Filter::*filterArray*($row, $fieldsList);  
 Core::*getInstance*()->db->insert(self::*$tableName*, $rowOrder);  
  
 $orderId = self::*getOrderId*($rowOrder['user\_id'], $rowOrder['date']);  
 foreach ($products as $product) {  
 if($product['product']['count'] != 0) {  
 $product['product']['count'] -= $product['count'];  
 Product::*update*($product['product']['id'], $product['product']);  
 Core::*getInstance*()->db->insert('order\_item', [  
 'order\_id' => $orderId,  
 'product\_id' => $product['product']['id'],  
 'count' => $product['count']  
 ]);  
 }  
 }  
 }  
 public static function getOrderId($userId, $dateTime) {  
 $ids = Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 'user\_id' => $userId,  
 'date' => $dateTime  
 ]);  
 if(isset($ids))  
 return $ids[0]['id'];  
 else  
 return null;  
 }  
 public static function getUserOrdersById($userId) {  
 $orders = Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 'user\_id' => $userId  
 ]);  
 return array\_reverse($orders);  
 }  
 public static function getOrdersByStatus($statusName) {  
 $orders = Core::*getInstance*()->db->select(self::*$tableName*, '\*', [  
 'status' => $statusName  
 ]);  
 return array\_reverse($orders);  
 }  
 public static function approve($id) {  
 Core::*getInstance*()->db->update(self::*$tableName*, [  
 'status' => 'approved'  
 ],[  
 'id' => $id  
 ]);  
 }  
 public static function getOrderItems($order\_id) {  
 $rows = Core::*getInstance*()->db->select('order\_item', '\*', [  
 'order\_id' => $order\_id  
 ]);  
 $products = [];  
 foreach ($rows as $row) {  
 $product = Product::*getById*($row['product\_id']);  
 $products [] = ['product' => $product, 'count\_in\_order' => $row['count']];  
 }  
 return $products;  
 }  
}

**DataTable.php**

<?php  
namespace models;  
  
use core\Core;  
use utils\Photo;  
  
class DataTable {  
 public static function addItem($tableName, $name, $fileName) {  
 Core::*getInstance*()->db->insert($tableName, [  
 'name' => $name,  
 'photo' => $fileName  
 ]);  
 }  
 public static function getById($tableName, $id) {  
 $rows = Core::*getInstance*()->db->select($tableName, '\*', [  
 'id' => $id  
 ]);  
 if(!empty($rows))  
 return $rows[0];  
 else  
 return null;  
 }  
 public static function getNameById($tableName, $id) {  
 $row = self::*getById*($tableName, $id);  
 if(!empty($row))  
 return $row['name'];  
 else  
 return null;  
 }  
 public static function deleteItem($tableName, $id) {  
 $rows = Core::*getInstance*()->db->delete($tableName, [  
 'id' => $id  
 ]);  
 }  
 public static function updateItem($tableName, $id, $newName) {  
 $rows = Core::*getInstance*()->db->update($tableName, [  
 'name' => $newName  
 ], [  
 'id' => $id  
 ]);  
 }  
 public static function getItems($tableName) {  
 $rows = Core::*getInstance*()->db->select($tableName);  
 $rows = array\_reverse($rows);  
 return $rows;  
 }  
 public static function getSortedItems($tableName) {  
 $rows = self::*getItems*($tableName);  
 $name = array\_column($rows, 'name');  
 array\_multisort($name, *SORT\_ASC*, $rows);  
 // Move 'Default' item to the first place  
 $i = 0;  
 foreach ($rows as $key => $value) {  
 if($value['name'] === 'Default') {  
 $default = $rows[$i];  
 unset($rows[$key]);  
 $rows = array($default) + $rows;  
 break;  
 }  
 $i++;  
 }  
 return $rows;  
 }  
}

**Basket.php**

<?php  
  
  
namespace models;  
  
  
use core\Core;  
  
class Basket  
{  
 protected static $tableName = 'basket';  
 public static function addProduct($product\_id, $count = 1) {  
 $rowProduct = Product::getById($product\_id);  
 $user\_id = User::getCurrentAuthenticatedUser()['id'];  
 if(isset($rowProduct)) {  
 $rowBasket = Core::getInstance()->db->select(self::$tableName, '\*', [  
 'user\_id' => $user\_id,  
 'product\_id' => $product\_id  
 ])[0];  
 if(isset($rowBasket)) {  
 Core::getInstance()->db->update(self::$tableName, [  
 'count' => $rowBasket['count'] + $count  
 ], [  
 'id' => $rowBasket['id']  
 ]);  
 }  
 else {  
 Core::getInstance()->db->insert(self::$tableName, [  
 'user\_id' => $user\_id,  
 'product\_id' => $rowProduct['id'],  
 'count' => $count  
 ]);  
 }  
 }  
 }  
 public static function getProducts() {  
 $rows = Core::getInstance()->db->select(self::$tableName, '\*', [  
 'user\_id' => User::getCurrentAuthenticatedUser()['id']  
 ]);  
 $result = [];  
 $products = [];  
 $totalPrice = 0;  
 foreach ($rows as $row) {  
 $product = Product::getById($row['product\_id']);  
 $products [] = ['product' => $product, 'count' => $row['count']];  
 if($row['count'] > $product['count'])  
 continue;  
 $totalPrice += $product['price'] \* $row['count'];  
 }  
 $result['products'] = $products;  
 $result['total\_price'] = $totalPrice;  
 return $result;  
 }  
 public static function getCountOfId($product\_id) {  
 $rows = Core::getInstance()->db->select(self::$tableName, '\*', [  
 'user\_id' => User::getCurrentAuthenticatedUser()['id'],  
 'product\_id' => $product\_id  
 ]);  
 if(isset($rows))  
 return $rows[0]['count'];  
 }  
 public static function clear() {  
 Core::getInstance()->db->delete(self::$tableName, [  
 'user\_id' => User::getCurrentAuthenticatedUser()['id']  
 ]);  
 }  
 public static function removeItem($product\_id) {  
 Core::getInstance()->db->delete(self::$tableName, [  
 'user\_id' => User::getCurrentAuthenticatedUser()['id'],  
 'product\_id' => $product\_id  
 ]);  
 }  
 public static function getCountItems() {  
 $rows = Core::getInstance()->db->select(self::$tableName, '\*', [  
 'user\_id' => User::getCurrentAuthenticatedUser()['id']  
 ]);  
 if(isset($rows))  
 return array\_sum(array\_column($rows, 'count'));  
 else  
 return 0;  
 }  
 public static function reduceOverflowItems() {  
 $rows = Core::getInstance()->db->select(self::$tableName, '\*', [  
 'user\_id' => User::getCurrentAuthenticatedUser()['id']  
 ]);  
 foreach ($rows as $rowBasket) {  
 $rowProduct = Product::getById($rowBasket['product\_id']);  
 if($rowBasket['count'] > $rowProduct['count']) {  
 Core::getInstance()->db->update(self::$tableName, [  
 'count' => $rowProduct['count']  
 ], [  
 'id' => $rowBasket['id']  
 ]);  
 }  
 }  
 }  
 public static function isEnoughStock() {  
 $rows = Core::getInstance()->db->select(self::$tableName, '\*', [  
 'user\_id' => User::getCurrentAuthenticatedUser()['id']  
 ]);  
 foreach ($rows as $rowBasket) {  
 $rowProduct = Product::getById($rowBasket['product\_id']);  
 if($rowBasket['count'] > $rowProduct['count']) {  
 return false;  
 }  
 }  
 return true;  
 }  
}

**ProductController.php**

<?php  
  
  
namespace controllers;  
  
  
use core\Controller;  
use core\Core;  
use models\DataTable;  
use models\Product;  
use models\User;  
use utils\Photo;  
  
class ProductController extends Controller  
{  
 public function indexAction() {  
 $searchString = $\_GET['search'];  
 if(isset($\_GET['page']))  
 $page = intval($\_GET['page']);  
 else  
 $page = 1;  
 $countPerPage = 12;  
 $rows = Product::*search*($searchString, User::*isAdmin*());  
 $products = Product::*getOnePage*($rows, $page, $countPerPage);  
 $totalNumber = count($rows);  
 if(!isset($products))  
 return $this->render('views/main/exception-no-content.php');  
 if(($page - 1) \* $countPerPage > $totalNumber || $page <= 0)  
 return $this->error(404);  
 return $this->render(null, [  
 'products' => $products,  
 'pagination' => [  
 'page' => $page,  
 'count' => $countPerPage,  
 'totalNumber' => $totalNumber  
 ],  
 'searchString' => $searchString  
 ]);  
 return $this->render();  
 }  
 public function getInputErrors() {  
 $errors = [];  
 $\_POST['name'] = trim($\_POST['name']);  
 if (empty($\_POST['name']))  
 $errors['name'] = 'Не вказано назву продукту';  
 if ($\_POST['price'] <= 0)  
 $errors['price'] = 'Неправильно вказана ціна';  
 if ($\_POST['count'] < 0)  
 $errors['count'] = 'Неправильно вказана кількість продуктів';  
 return $errors;  
 }  
 public function addAction($params) {  
 $moduleName = $params[0];  
 $module\_id = intval($params[1]);  
 if (empty($module\_id))  
 $module\_id = null;  
  
 $materialsList = DataTable::*getSortedItems*('material');  
 $typesList = DataTable::*getSortedItems*('type');  
 $gendersList = DataTable::*getSortedItems*('gender');  
 $brandsList = DataTable::*getSortedItems*('brand');  
 if (Core::*getInstance*()->requestMethod === 'POST') {  
 $errors = $this->getInputErrors();  
  
 if (empty($errors)) {  
 $fileName = Photo::*loadPhoto*('product', $\_FILES['file']['tmp\_name']);  
 Product::*add*($\_POST, $fileName);  
 $this->redirect('/product');  
 } else {  
 $model = $\_POST;  
 return $this->render(null, [  
 'errors' => $errors,  
 'model' => $model,  
 'materials' => $materialsList,  
 'types' => $typesList,  
 'genders' => $gendersList,  
 'brands' => $brandsList,  
 $moduleName.'\_id' => $module\_id  
 ]);  
 }  
 }  
 return $this->render(null, [  
 'materials' => $materialsList,  
 'types' => $typesList,  
 'genders' => $gendersList,  
 'brands' => $brandsList,  
 $moduleName.'\_id' => $module\_id  
 ]);  
 }  
 public function editAction($params) {  
 $id = $params[0];  
 if(!User::*isAdmin*())  
 return $this->error(403);  
 if($id > 0) {  
 $row = Product::*getById*($id);  
 $materialsList = DataTable::*getSortedItems*('material');  
 $typesList = DataTable::*getSortedItems*('type');  
 $gendersList = DataTable::*getSortedItems*('gender');  
 $brandsList = DataTable::*getSortedItems*('brand');  
 if(Core::*getInstance*()->requestMethod === 'POST') {  
 $errors = [];  
 $\_POST['name'] = trim($\_POST['name']);  
 $errors = $this->getInputErrors();  
  
 if (empty($errors)) {  
 Product::*update*($id, $\_POST);  
 if(!empty($\_FILES['file']['tmp\_name']))  
 Photo::*changePhoto*('product', $id, $\_FILES['file']['tmp\_name']);  
 $this->redirect("/product/view/{$id}");  
 } else {  
 $model = $\_POST;  
 return $this->render(null, [  
 'errors' => $errors,  
 'model' => $model,  
 'row' => $row,  
 'materials' => $materialsList,  
 'types' => $typesList,  
 'genders' => $gendersList,  
 'brands' => $brandsList  
 ]);  
 }  
 }  
 return $this->render(null, [  
 'row' => $row,  
 'materials' => $materialsList,  
 'types' => $typesList,  
 'genders' => $gendersList,  
 'brands' => $brandsList  
 ]);  
 } else  
 return $this->error(403);  
 }  
 public function deleteAction($params) {  
 $id = intval($params[0]);  
 if(!User::*isAdmin*())  
 return $this->error(403);  
 if($id > 0) {  
 Photo::*deletePhoto*('product', $id);  
 Product::*delete*($id);  
 $this->redirect('/product/index');  
 } else  
 return $this->error(403);  
 }  
 public function viewAction($params) {  
 $id = intval($params[0]);  
 $product = Product::*getById*($id);  
 return $this->render(null, [  
 'product' => $product  
 ]);  
 }  
 public function searchAction($params) {  
 $searchString = $params[0];  
 $rows = Product::*search*($searchString);  
 }  
}

**UserController.php**

<?php  
  
  
namespace controllers;  
  
use core\Controller;  
use core\Core;  
use models\Order;  
use models\User;  
  
class UserController extends Controller  
{  
 public function registerAction() {  
 if(User::*isUserAuthenticated*())  
 $this->redirect('/');  
 if (Core::*getInstance*()->requestMethod === 'POST') {  
 $errors = [];  
  
 if(!filter\_var($\_POST['login'], *FILTER\_VALIDATE\_EMAIL*))  
 $errors['login'] = 'Error entering email';  
 if(User::*isLoginExists*($\_POST['login']))  
 $errors['login'] = 'This login is busy <a href="/user/login">Login?</a>';  
  
 if($\_POST['password'] != $\_POST['password2']) {  
 $errors['password'] = 'Passwords do not match';  
 $errors['password'] = 'Passwords do not match';  
 }  
  
 foreach ($\_POST as $key => $value) {  
 if(trim($value) == null) {  
 $errors[$key] = ucfirst($key).' is not set';  
 }  
 }  
  
 if(count($errors) > 0) {  
 $model = $\_POST;  
 return $this->render(null, [  
 'errors' => $errors,  
 'model' => $model  
 ]);  
 } else {  
 User::*addUser*($\_POST['login'], $\_POST['password'], $\_POST['firstname'], $\_POST['lastname']);  
 return $this->renderView('register-success');  
 }  
 } else  
 return $this->render();  
 }  
  
 public function loginAction() {  
 if(User::*isUserAuthenticated*())  
 $this->redirect('/');  
 if (Core::*getInstance*()->requestMethod === 'POST') {  
 $user = User::*getUserByLoginAndPassword*($\_POST['login'], $\_POST['password']);  
 $error = null;  
 if(empty($user)) {  
 $error = 'Incorrect login or password';  
 } else {  
 User::*authenticateUser*($user);  
 $this->redirect('/');  
 }  
 }  
 return $this->render(null, [  
 'error' => $error  
 ]);  
 }  
  
 public function logoutAction() {  
 if(!User::*isUserAuthenticated*())  
 $this->redirect('/user/login');  
  
 User::*logoutUser*();  
 $this->redirect('/');  
 }  
  
 public function indexAction() {  
 if(!User::*isUserAuthenticated*())  
 $this->redirect('/user/login');  
  
 if(User::*isAdmin*())  
 $orders = Order::*getOrdersByStatus*('in processing');  
 else  
 $orders = Order::*getUserOrdersById*(User::*getCurrentAuthenticatedUser*()['id']);  
  
 return $this->render(null, [  
 'orders' => $orders  
 ]);  
 }  
  
 public function approveAction($params) {  
 if (!User::*isAdmin*())  
 return $this->error(403);  
  
 $orderId = intval($params[0]);  
 Order::*approve*($orderId);  
 $this->redirect("/user");  
 }  
}

**BasketController.php**

<?php  
  
  
namespace controllers;  
  
  
use core\Controller;  
use models\Basket;  
use models\Product;  
use models\User;  
  
class BasketController extends Controller  
{  
 public function indexAction($params) {  
 if(!User::*isUserAuthenticated*())  
 $this->redirect('/user/login');  
  
 $id = intval($params[0]);  
  
 $basket = Basket::*getProducts*();  
 return $this->render(null, [  
 'basket' => $basket  
 ]);  
 }  
 public function addAction($params) {  
 if(!User::*isUserAuthenticated*())  
 $this->redirect('/user/login');  
  
 $id = intval($params[0]);  
 $product = Product::*getById*($id);  
 if(isset($product)) {  
 Basket::*addProduct*($id);  
 $this->redirect('/basket');  
 }  
 }  
 public function deleteAction($params) {  
 if(!User::*isUserAuthenticated*())  
 $this->redirect('/user/login');  
  
 $id = intval($params[0]);  
 Basket::*removeItem*($id);  
 $this->redirect('/basket');  
 }  
 public function reduceAction() {  
 Basket::*reduceOverflowItems*();  
 $this->redirect('/basket');  
 }  
}

**DataController.php**

<?php  
  
  
namespace controllers;  
use \core\Controller;  
use core\Core;  
use models\DataTable;  
use models\Product;  
use models\User;  
use utils\Photo;  
  
  
class DataController extends Controller  
{  
 public static function indexAction($controller, $moduleName) {  
 $rows = DataTable::*getItems*($moduleName);  
 if(!isset($rows))  
 return $controller->render('views/main/exception-no-content.php');  
  
 foreach ($rows as &$row)  
 $row += array('products\_count' => count(self::*getProductsIn*($moduleName, $row['id'])));  
  
 return $controller->render(null, [  
 'moduleName' => $moduleName,  
 'rows' => $rows  
 ]);  
 }  
 public static function addAction($controller, $moduleName)  
 {  
 if (!User::*isAdmin*())  
 return $controller->error(403);  
 if (Core::*getInstance*()->requestMethod === 'POST') {  
 $errors = [];  
 $\_POST['name'] = trim($\_POST['name']);  
 if (empty($\_POST['name']))  
 $errors['name'] = "The name of the {$moduleName} is not specified";  
 if (empty($errors)) {  
 $fileName = Photo::*loadPhoto*($moduleName, $\_FILES['file']['tmp\_name']);  
 DataTable::*addItem*($moduleName, $\_POST['name'], $fileName);  
 return $controller->redirect("/{$moduleName}/index");  
 } else {  
 $model = $\_POST;  
 return $controller->render("views/$moduleName/add.php", [  
 'moduleName' => $moduleName,  
 'errors' => $errors,  
 'model' => $model  
 ]);  
 }  
 }  
 return $controller->render(null, [  
 'moduleName' => $moduleName  
 ]);  
 }  
 public static function deleteAction($controller, $moduleName, $id, $answer) {  
 if(!User::*isAdmin*())  
 return $controller->error(403);  
 if($id > 0) {  
 $row = DataTable::*getById*($moduleName, $id);  
 if($answer) {  
 $filePath = "files/{$moduleName}/{$row['photo']}";  
 if(is\_file($filePath)) {  
 unlink($filePath);  
 }  
 $products = Product::*getProductsIn*($moduleName, $id);  
 foreach ($products as $product) {  
 Product::*setDefaultForNull*($moduleName, $product['id']);  
 }  
 Photo::*deletePhoto*($moduleName, $id);  
 DataTable::*deleteItem*($moduleName, $id);  
 return $controller->redirect("/{$moduleName}/index");  
   
 }  
 return $controller->render(null, [  
 'moduleName' => $moduleName,  
 'row' => $row  
 ]);  
 } else  
 return $controller->error(403);  
 }  
 public static function editAction($controller, $moduleName, $id) {  
 if(!User::*isAdmin*())  
 return $controller->error(403);  
 if($id > 0) {  
 $row = DataTable::*getById*($moduleName, $id);  
 if(Core::*getInstance*()->requestMethod === 'POST') {  
 $errors = [];  
 $\_POST['name'] = trim($\_POST['name']);  
 if (empty($\_POST['name']))  
 $errors['name'] = "The name of the $moduleName is not specified";  
  
 if (empty($errors)) {  
 DataTable::*updateItem*($moduleName, $id, $\_POST['name']);  
 if(!empty($\_FILES['file']['tmp\_name']))  
 Photo::*changePhoto*($moduleName, $id, $\_FILES['file']['tmp\_name']);  
 return $controller->redirect("/{$moduleName}/index");  
 } else {  
 $model = $\_POST;  
 return $controller->render(null, [  
 'moduleName' => $moduleName,  
 'errors' => $errors,  
 'model' => $model,  
 'row' => $row  
 ]);  
 }  
 }  
 return $controller->render(null, [  
 'moduleName' => $moduleName,  
 'row' => $row  
 ]);  
 } else  
 return $controller->error(403);  
 }  
 public static function viewAction($controller, $moduleName, $id) {  
 $row = DataTable::*getById*($moduleName, $id);  
 $products = self::*getProductsIn*($moduleName, $id);  
 return $controller->render(null, [  
 'moduleName' => $moduleName,  
 'row' => $row,  
 'products' => $products  
 ]);  
 }  
  
 public static function getProductsIn($moduleName, $id) {  
 $className = 'Product';  
 $methodName = 'getProductsIn';  
 $products = call\_user\_func('\\models\\'.$className.'::'.$methodName, $moduleName, $id);  
 return $products;  
 }  
}

**basket/index.php**

<?php  
*/\*\** ***@var*** *array $basket \*/*use models\Basket;  
  
?>  
 <h1 class="h3 mb-4 fw-normal">Кошик</h1>  
<?php if(count($basket['products']) == 0) : ?>  
 <div class="empty-block">  
 <h3 class="h3">Кошик порожній :(</h3>  
 <span>Але ви можете замовити товари <a href="/product">тут</a>.</span>  
 </div>  
<?php else : ?>  
 <table class="table">  
 <thead>  
 <tr>  
 <th></th>  
 <th>№</th>  
 <th>Зображення</th>  
 <th>Назва</th>  
 <th>Ціна за одиницю</th>  
 <th>Кількість</th>  
 <th>Вартість</th>  
 <th>Наявність</th>  
 </tr>  
 </thead>  
 <?php  
 $index = 1;  
 $reduceMarker = false;  
 foreach ($basket['products'] as $row) : ?>  
 <tr class="table-row-dyn <?php  
 if($row['count'] > $row['product']['count'])  
 echo 'table-danger';  
 else if($row['count'] == 0)  
 echo 'table-secondary';  
 ?>" onclick="  
 ***window***.location='/product/view/<?=$row['product']['id']?>';  
 ">  
 <td onclick="  
 if (!***e***) var ***e*** = ***window***.event;  
 ***e***.cancelBubble = true;  
 if (***e***.stopPropagation) ***e***.stopPropagation();  
 ***window***.location='/basket/delete/<?=$row['product']['id']?>';  
 ">  
 <svg xmlns="http://www.w3.org/2000/svg" width="32" height="32" fill="currentColor" class="bi bi-x" viewBox="0 0 16 16">  
 <path d="M4.646 4.646a.5.5 0 0 1 .708 0L8 7.293l2.646-2.647a.5.5 0 0 1 .708.708L8.707 8l2.647 2.646a.5.5 0 0 1-.708.708L8 8.707l-2.646 2.647a.5.5 0 0 1-.708-.708L7.293 8 4.646 5.354a.5.5 0 0 1 0-.708z"/>  
 </svg>  
 </td>  
 <td><?=$index ?></td>  
 <td>  
 <?php $filePath = 'files/product/'.$row['product']['photo']; ?>  
 <?php  
 if (is\_file($filePath)) :?>  
 <img src="/<?=$filePath?>" width="100">  
 <?php else : ?>  
 <img src="/static/images/no-image.jpg" width="100">  
 <?php endif; ?>  
 </td>  
 <td><?=$row['product']['name'] ?></td>  
 <td><?=$row['product']['price'] ?> грн.</td>  
 <td><?=$row['count']?>  
 </td>  
 <?php  
 if($row['count'] > $row['product']['count']) {  
 $reduceMarker = true;  
 }  
 ?>  
 <td><?=$row['product']['price'] \* $row['count'] ?> грн.</td>  
 <td><?php  
 if($row['count'] > $row['product']['count'])  
 echo 'Недостатньо на складі';  
 else if ($row['product']['count'] == 0)  
 echo 'Не доступний';  
 else  
 echo 'Є в наявності';  
 ?></td>  
 </tr>  
 <?php  
 $index++;  
 endforeach; ?>  
 <tfoot>  
 <tr>  
 <th>Загальна вартість</th>  
 <th></th>  
 <th></th>  
 <th></th>  
 <th></th>  
 <th></th>  
 <th><?=$basket['total\_price'] ?> грн.</th>  
 <th>  
 <?php if($reduceMarker) : ?>  
 <a class="btn btn-danger" href="/basket/reduce">Зменшити до доступної кількості</a>  
 <?php elseif(Basket::*getCountItems*() == 0) : ?>  
 <a class="btn btn-primary disabled" href="/order">Замовити</a>  
 <?php else : ?>  
 <a class="btn btn-primary" href="/order">Замовити</a>  
 <?php endif; ?>  
 </th>  
 </tr>  
 </tfoot>  
 </table>  
<?php endif; ?>

**product/add.php**

<?php  
*/\*\** ***@var*** *array $model \*/  
/\*\** ***@var*** *array $errors \*/  
/\*\** ***@var*** *array $materials \*/  
/\*\** ***@var*** *int|null $material\_id \*/  
/\*\** ***@var*** *array $types \*/  
/\*\** ***@var*** *int|null $type\_id \*/  
/\*\** ***@var*** *array $genders \*/  
/\*\** ***@var*** *int|null $gender\_id \*/  
/\*\** ***@var*** *array $brands \*/  
/\*\** ***@var*** *int|null $brand\_id \*/*\core\Core::*getInstance*()->pageParams['title'] = 'Додавання продукту';  
?>  
  
<h1 class="h3 mb-4 fw-normal text-center">Додати продукт</h1>  
<form action="" method="post" enctype="multipart/form-data">  
 <div class="mb-3">  
 <label for="name" class="form-label">Назва продукту</label>  
 <input type="text" class="form-control" id="name" name="name">  
 <?php if (!empty($errors['name'])): ?>  
 <div class="form-text text-danger"><?=$errors['name'] ?></div>  
 <?php endif; ?>  
 </div>  
  
 <div class="mb-3">  
 <label for="material\_id" class="form-label">Вибрати матеріал продукта</label>  
 <select class="form-control" id="material\_id" name="material\_id">  
 <?php foreach ($materials as $row) : ?>  
 <option <?php if($row['id'] == $material\_id) echo 'selected'; ?> value="<?=$row['id']?>"><?=$row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['material\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['material\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
  
 <div class="mb-3">  
 <label for="type\_id" class="form-label">Вибрати тип продукта</label>  
 <select class="form-control" id="type\_id" name="type\_id">  
 <?php foreach ($types as $row) : ?>  
 <option <?php if($row['id'] == $type\_id) echo 'selected'; ?> value="<?=$row['id']?>"><?=$row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['type\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['type\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
  
 <div class="mb-3">  
 <label for="gender\_id" class="form-label">Виберіть стать </label>  
 <select class="form-control" id="gender\_id" name="gender\_id">  
 <?php foreach ($genders as $row) : ?>  
 <option <?php if($row['id'] == $gender\_id) echo 'selected'; ?> value="<?=$row['id']?>"><?=$row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['gender\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['gender\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
  
 <div class="mb-3">  
 <label for="brand\_id" class="form-label">Вибрати бренд продукта</label>  
 <select class="form-control" id="brand\_id" name="brand\_id">  
 <?php foreach ($brands as $row) : ?>  
 <option <?php if($row['id'] == $brand\_id) echo 'selected'; ?> value="<?=$row['id']?>"><?=$row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['brand\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['brand\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
  
 <div class="mb-3">  
 <label for="price" class="form-label">Виберіть ціну</label>  
 <input type="number" class="form-control" id="price" name="price">  
 <?php if (!empty($errors['price'])): ?>  
 <div class="form-text text-danger"><?=$errors['price'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="count" class="form-label">Кількість одиниць продукту</label>  
 <input type="number" class="form-control" id="count" name="count">  
 <?php if (!empty($errors['count'])): ?>  
 <div class="form-text text-danger"><?=$errors['count'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="short\_description" class="form-label">Короткий опис</label>  
 <textarea class="ck-editor form-control" name="short\_description" id="short\_description"></textarea>  
 <?php if (!empty($errors['short\_description'])): ?>  
 <div class="form-text text-danger"><?=$errors['short\_description'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="description" class="form-label">Розширений опис</label>  
 <textarea class="ck-editor form-control" name="description" id="description"></textarea>  
 <?php if (!empty($errors['description'])): ?>  
 <div class="form-text text-danger"><?=$errors['description'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="file" class="form-label">Файл з фото</label>  
 <input type="file" class="form-control" name="file" id="file" accept="image/jpeg"/>  
 <?php if (!empty($errors['file'])): ?>  
 <div class="form-text text-danger"><?=$errors['file'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="visible" class="form-label">Показати товар?</label>  
 <select class="form-control" id="visible" name="visible">  
 <option value="1">Так</option>  
 <option value="0">Ні</option>  
 </select>  
 </div>  
 <div>  
 <button class="btn btn-primary">Додати</button>  
 </div>  
</form>  
<script src="https://cdn.ckeditor.com/ckeditor5/35.4.0/classic/ckeditor.js"></script>  
<script>  
 let editors = document.querySelectorAll('.ck-editor');  
 for(let editor of editors) {  
 ClassicEditor  
 .create(editor)  
 .catch( error => {  
 console.error(error);  
 } );  
 }  
</script>

**product/edit.php**

<?php  
*/\*\** ***@var*** *array $row \*/  
/\*\** ***@var*** *array $modelАле не забувайте \*/  
/\*\** ***@var*** *array $errors \*/  
/\*\** ***@var*** *array $materials \*/  
/\*\** ***@var*** *array $types \*/  
/\*\** ***@var*** *array $genders \*/  
/\*\** ***@var*** *array $brands \*/*\core\Core::*getInstance*()->pageParams['title'] = 'Редагування продукту';  
  
if(isset($model)) $row = $model;  
?>  
  
<h1 class="h3 mb-4 fw-normal text-center">Редагування продукту</h1>  
<form action="" method="post" enctype="multipart/form-data">  
 <div class="mb-3">  
 <label for="name" class="form-label">Назва продукту</label>  
 <input type="text" class="form-control" id="name" name="name" value="<?=$row['name']?>">  
 <?php if (!empty($errors['name'])): ?>  
 <div class="form-text text-danger"><?=$errors['name'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="material\_id" class="form-label">Виберіть матеріал продукту</label>  
 <select class="form-control" id="material\_id" name="material\_id">  
 <?php foreach ($materials as $material\_row) :  
 ?>  
 <option <?php if($material\_row['id'] == $row['material\_id']) echo 'selected'; ?> value="<?=$material\_row['id']?>"><?=$material\_row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['material\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['material\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="type\_id" class="form-label">Виберіть тип продукту</label>  
 <select class="form-control" id="type\_id" name="type\_id">  
 <?php foreach ($types as $type\_row) : ?>  
 <option <?php if($type\_row['id'] == $row['type\_id']) echo 'selected'; ?> value="<?=$type\_row['id']?>"><?=$type\_row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['type\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['type\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="gender\_id" class="form-label">Виберіть стать </label>  
 <select class="form-control" id="gender\_id" name="gender\_id">  
 <?php foreach ($genders as $gender\_row) : ?>  
 <option <?php if($gender\_row['id'] == $row['gender\_id']) echo 'selected'; ?> value="<?=$gender\_row['id']?>"><?=$gender\_row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['gender\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['gender\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="brand\_id" class="form-label">Виберіть бренд</label>  
 <select class="form-control" id="brand\_id" name="brand\_id">  
 <?php foreach ($brands as $brand\_row) : ?>  
 <option <?php if($brand\_row['id'] == $row['brand\_id']) echo 'selected'; ?> value="<?=$brand\_row['id']?>"><?=$brand\_row['name']?></option>  
 <?php endforeach; ?>  
 </select>  
 <?php if (!empty($errors['brand\_id'])): ?>  
 <div class="form-text text-danger"><?=$errors['brand\_id'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="price" class="form-label">Виберіть ціну</label>  
 <input type="number" class="form-control" id="price" name="price" value="<?=$row['price']?>">  
 <?php if (!empty($errors['price'])): ?>  
 <div class="form-text text-danger"><?=$errors['price'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="count" class="form-label">Кількість одиниць продукту</label>  
 <input type="number" class="form-control" id="count" name="count" value="<?=$row['count']?>">  
 <?php if (!empty($errors['count'])): ?>  
 <div class="form-text text-danger"><?=$errors['count'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="short\_description" class="form-label">Короткий опис</label>  
 <textarea class="ck-editor form-control" name="short\_description" id="short\_description"><?=$row['short\_description']?></textarea>  
 <?php if (!empty($errors['short\_description'])): ?>  
 <div class="form-text text-danger"><?=$errors['short\_description'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="description" class="form-label">Розширений опис</label>  
 <textarea class="ck-editor form-control" name="description" id="description"><?=$row['description']?></textarea>  
 <?php if (!empty($errors['description'])): ?>  
 <div class="form-text text-danger"><?=$errors['description'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="col-3">  
 <?php $filePath = 'files/product/'.$row['photo'];?>  
 <?php if (is\_file($filePath)) : ?>  
 <img src="/<?=$filePath?>" class="card-img-top img-thumbnail">  
 <?php else : ?>  
 <img src="/static/images/no-image.jpg" class="card-img-top img-thumbnail">  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="file" class="form-label">Файл з фото</label>  
 <input type="file" class="form-control" name="file" id="file" accept="image/jpeg" value=""/>  
 <?php if (!empty($errors['file'])): ?>  
 <div class="form-text text-danger"><?=$errors['file'] ?></div>  
 <?php endif; ?>  
 </div>  
 <div class="mb-3">  
 <label for="visible" class="form-label">Показати товар?</label>  
 <select class="form-control" id="visible" name="visible">  
 <option value="1" <?php if($row['visible']) echo 'selected'; ?>>Так</option>  
 <option value="0" <?php if(!$row['visible']) echo 'selected'; ?>>Ні</option>  
 </select>  
 </div>  
 <div>  
 <button class="btn btn-primary">Редагувати</button>  
 </div>  
</form>  
<script src="https://cdn.ckeditor.com/ckeditor5/35.4.0/classic/ckeditor.js"></script>  
<script>  
 let ***editors*** = ***document***.querySelectorAll('.ck-editor');  
 for(let editor of ***editors***) {  
 ClassicEditor  
 .create(editor)  
 .catch( error => {  
 ***console***.error(error);  
 } );  
 }  
</script>

**product/index.php**

<?php  
  
*/\*\** ***@var*** *array $products \*/  
/\*\** ***@var*** *array $pagination \*/  
/\*\** ***@var*** *string $searchString \*/*use models\User;  
  
\core\Core::*getInstance*()->pageParams['title'] = 'Список продуктів';  
?>  
  
<h1 class="h3 mb-4 fw-normal text-center">Список продуктів</h1>  
<?php if(User::*isAdmin*()) : ?>  
 <div class="mb-3">  
 <a href="/product/add" class="btn btn-success">Додати продукт</a>  
 </div>  
<?php endif; ?>  
  
<?php require 'views/parts/products-plates.php'?>  
  
<?php  
$paginationBar = \utils\Pagination::*GetStylingArray*(  
 $pagination['page'],  
 $pagination['count'],  
 $pagination['totalNumber']  
);  
$pagination = $paginationBar['paginationNums'];  
?>  
<nav class="nav-pagination">  
 <ul class="pagination">  
 <li class="page-item">  
 <a class="page-link <?=$paginationBar['previous']?>" href="/product/index?search=<?=$searchString?>&page=<?=$paginationBar['previous']?>" aria-label="Попередня">  
 <span aria-hidden="true">&laquo;</span>  
 <span class="sr-only">Попередня</span>  
 </a>  
 </li>  
 <?php foreach ($pagination as $key => $value) :?>  
 <li class="page-item">  
 <a  
 class="page-link <?=$value?>"  
 href="/product/index?search=<?=$searchString?>&page=<?=$key?>"  
 >  
 <?=$key?>  
 </a>  
 </li>  
 <?php endforeach; ?>  
 <li class="page-item">  
 <a class="page-link <?=$paginationBar['next']?>" href="/product/index?search=<?=$searchString?>&page=<?=$paginationBar['next']?>" aria-label="Наступна">  
 <span aria-hidden="true">&raquo;</span>  
 <span class="sr-only">Наступна</span>  
 </a>  
 </li>  
 </ul>  
</nav>

**product/view.php**

<?php  
*/\*\** ***@var*** *array $product \*/*use models\DataTable;  
use models\User;  
  
\core\Core::*getInstance*()->pageParams['title'] = 'Перегляд товару';  
?>  
<div class="container view-container">  
 <div class="row">  
 <div class="col-4 col-img">  
 <?php $filePath = 'files/product/' . $product['photo']; ?>  
 <?php if (is\_file($filePath)) : ?>  
 <a data-fancybox data-src="/<?= $filePath ?>">  
 <img src="/<?= $filePath ?>" class="product-view-avatar">  
 </a>  
 <?php else : ?>  
 <img src="/static/images/no-image.jpg" class="product-view-avatar">  
 <?php endif; ?>  
 </div>  
 <div class="col-8 col-info">  
 <div class="row mb-1">  
 <h3 class="h4"><?= $product['name'] ?></h3>  
 </div>  
 <div class="buy-block">  
 <div>  
 <span class="price"><?=$product['price']?></span> <span class="cur">грн.</span>  
 </div>  
 <?php  
 $countInBasket = \models\Basket::*getCountOfId*($product['id']);  
 $classBtn = "";  
 $disableLink = "";  
  
 if(User::*isUserAuthenticated*() && $countInBasket != 0) {  
 if ($countInBasket < $product['count'])  
 $outputBtn = "Додати ({$countInBasket} у кошику)";  
 else if ($countInBasket == $product['count']) {  
 $outputBtn = "Все у кошику ({$countInBasket})";  
 $classBtn = "disabled";  
 }  
 }  
 else if($product["count"] == 0) {  
 $outputBtn = "Немає в наявності";  
 $classBtn = "disabled";  
 $disableLink = "href-dis";  
 }  
 else {  
 $outputBtn = "Додати до кошика";  
 }  
  
 ?>  
 <a href="/basket/add/<?=$product['id']?>" class="<?=$disableLink?>">  
 <div class="btn cart-btn <?=$classBtn?>">  
 <?=$outputBtn?>  
 </div>  
 </a>  
 </div>  
 <div class="row mb-1">  
 <section>  
 <div class="row-info-mg">  
 <span class="fw-bold">Категорії</span>  
 </div>  
 <div class="row-info row-info-mg material">  
 <a href="/material/view/<?=$product['material\_id']?>" class="material">  
 <?=DataTable::*getNameById*('material', $product['material\_id'])?>  
 </a>  
 <a href="/brand/view/<?=$product['brand\_id']?>" class="material">  
 <?=DataTable::*getNameById*('brand', $product['brand\_id'])?>  
 </a>  
 <a href="/type/view/<?=$product['type\_id']?>" class="material">  
 <?=DataTable::*getNameById*('type', $product['type\_id'])?>  
 </a>  
 <a href="/gender/view/<?=$product['gender\_id']?>" class="material">  
 <?=DataTable::*getNameById*('gender', $product['gender\_id'])?>  
 </a>  
 </div>  
 <div class="row-info-mg">  
 <span class="fw-bold">Доставка</span>  
 </div>  
 <div class="row-info row-info-mg">  
 <a href="https://novaposhta.ua/">  
 <img src="../../static/layout/nova\_poshta.svg" class="icon"/>  
 <span class="name">Нова Пошта</span>  
 <span class="description">ви зможете забрати його протягом <b>1-3 днів</b> у відділенні</span>  
 </a>  
 </div>  
 <div class="row-info row-info-mg">  
 <a href="https://novaposhta.ua/dostavka\_courier">  
 <img src="../../static/layout/car.svg" class="icon"/>  
 <span class="name">Кур'єром</span>  
 <span class="description">перевезення здійснюється службою <b>Нова Пошта</b></span>  
 </a>  
 </div>  
 </section>  
 <section>  
 <div class="row-info-mg">  
 <span class="fw-bold">Оплата</span>  
 </div>  
 <div class="row-info row-info-mg">  
 <a>  
 <img src="../../static/layout/card.svg" class="icon"/>  
 <span class="name">Карткою</span>  
 </a>  
 <a>  
 <img src="../../static/layout/cash.svg" class="icon"/>  
 <span class="name">Готівкою</span>  
 </a>  
 </div>  
 </section>  
 </div>  
 </div>  
 </div>  
 <div class="row descriptions">  
 <div class="col-12">  
 <div class="brief-description">  
 <?php if(isset($product['short\_description'])) : ?>  
 <span class="title">Короткий опис:</span>  
 <span class="description"><?=$product['short\_description']?></span>  
 <?php endif; ?>  
 <?php if(isset($product['description'])) : ?>  
 <span class="title">Повний опис:</span>  
 <span class="description"><?=$product['description']?></span>  
 <?php endif; ?>  
 </div>  
 </div>  
 </div>  
</div>

**parts/products-plates.php**

<?php  
*/\*\** ***@var*** *array $products \*/*use models\Basket;  
use models\User;  
?>  
  
<div class="item-cards">  
 <?php foreach ($products as $row) : ?>  
 <?php if (!$row['visible'] && !User::*isAdmin*()) continue; ?>  
 <div class="item-card">  
 <a href="/product/view/<?=$row['id'] ?>" class="card-link">  
 <?php $filePath = 'files/product/'.$row['photo'];?>  
 <?php if (is\_file($filePath)) : ?>  
 <img src="/<?=$filePath?>" class="avatar product-avatar <?php if(!$row['visible']) echo 'black-white'?>">  
 <?php else : ?>  
 <img src="/static/images/no-image.jpg" class="avatar <?php if(!$row['visible']) echo 'black-white'?>">  
 <?php endif; ?>  
 <?php if (!$row['visible']) : ?>  
 <img src="../../static/layout/unvisible.png" class="invis">  
 <?php endif; ?>  
 <div class="product-name name">  
 <?=$row['name']?>  
 </div>  
 <div class="card-row">  
 <div class="price">  
 <span class="number"><?=$row['price']?></span> <span class="cur">грн.</span>  
 </div>  
 <?php  
 $classBtn = "";  
 $disableLink = "";  
 if ($row['count'] == Basket::*getCountOfId*($row['id']) || $row['count'] == 0) {  
 $classBtn = "disabled";  
 $disableLink = "href-dis";  
 }  
 ?>  
 <a href="/basket/add/<?=$row['id']?>" class="<?=$disableLink?>">  
 <div class="btn cart-btn <?=$classBtn?>">  
 <img src="../../static/layout/cart.svg">  
 </div>  
 </a>  
 </div>  
 <div class="card-body">  
 <?php if(User::*isAdmin*()) : ?>  
 <div class="card-row <?php if($row['count'] == 0) echo 'text-danger fw-bold' ?>">  
 <?=$row['count']?> В наявності  
 </div>  
 <a href="/product/edit/<?=$row['id'] ?>" class="btn btn-primary">Редагувати</a>  
 <a href="/product/delete/<?=$row['id'] ?>" class="btn btn-danger">Видалити</a>  
 <?php endif; ?>  
 </div>  
 </a>  
 </div>  
 <?php endforeach; ?>  
</div>

**themes/layout.php**

<?php  
*/\*\** ***@var*** *string $title \*/  
/\*\** ***@var*** *string $content \*/  
/\*\** ***@var*** *string $siteName \*/*use models\User;  
use models\DataTable;  
if (User::*isUserAuthenticated*())  
 $user = User::*getCurrentAuthenticatedUser*();  
else  
 $user = null;  
?>  
<!doctype html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <link rel="icon" href="../../static/layout/icon.png">  
 <title><?=$siteName ?> | <?=$title ?></title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">  
 <link rel="stylesheet" href="/themes/light/css/style.css" />  
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fancyapps/ui/dist/fancybox.css"/>  
 <link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-bundle.min.css" />  
</head>  
<body>  
<div class="wrapper">  
 <header>  
 <div class="top-row">  
 <div class="logo">  
 <a href="/">  
 <img src="../../static/layout/logotipe.png" width="200"/>  
 </a>  
 </div>  
 <div class="search">  
 <form action="" method="post">  
 <div class="search-module input-group rounded">  
 <input type="text" name="search" class="form-control rounded" value="<?=$\_GET['search']?>" placeholder="Пошук" />  
 <button type="submit" name="btn-search" class="input-group-text border-0" id="search">  
 <img src="../../static/layout/search.png" width="20">  
 </button>  
 </div>  
 </form>  
 </div>  
 <?php  
 if(isset($\_POST['btn-search'])) {  
 header("Location: /product/index?search={$\_POST['search']}");  
 }  
 ?>  
 <div class="user">  
 <?php if (User::*isUserAuthenticated*()) : ?>  
 <a href="/user" class="btn btn-light text-dark me-2 <?php if (User::*isAdmin*()) echo 'admin-label'?>">Привіт, <?=$user['firstname']?></a>  
 <a href="/basket" class="btn btn-light text-dark me-2">  
 Кошик  
 <span class="count-in-basket">  
 <?php  
 $count = \models\Basket::*getCountItems*();  
 if($count > 0) echo "({$count})";  
 ?>  
 </span>  
 </a>  
 <a href="/user/logout" class="btn btn-light text-dark me-2">Вийти</a>  
 <?php else : ?>  
 <a href="/user/login" class="btn btn-light text-dark me-2 ">Увійти</a>  
 <a href="/user/register" class="btn btn-dark">Зареєструватися</a>  
 <?php endif; ?>  
 </div>  
 </div>  
 <div class="bottom-row">  
 <ul class="modules">  
 <li><a href="/">Головна</a></li>  
 <li><a href="/product">Усі товари</a></li>  
 <li><a href="/material">Матеріал</a></li>  
 <li><a href="/brand">Бренди</a></li>  
 <li><a href="/gender">Стать</a></li>  
 <li><a href="/type">Тип</a></li>  
 </ul>  
 </div>  
 </header>  
  
 <main>  
 <div class="container">  
 <?=$content ?>  
 </div>  
 </main>  
  
 <footer class="py-3 my-4">  
 <ul class="nav justify-content-center border-bottom pb-3 mb-3">  
 <li class="nav-item"><a href="/" class="nav-link px-2 text-muted">Головна</a></li>  
 <li class="nav-item"><a href="/product" class="nav-link px-2 text-muted">Усі товари</a></li>  
 <li class="nav-item"><a href="https://www.instagram.com/" class="nav-link px-2 text-muted">Instagram</a></li>  
 <li class="nav-item"><a href="https://www.facebook.com/" class="nav-link px-2 text-muted">Facebook</a></li>  
 <li class="nav-item"><a href="https://twitter.com/home" class="nav-link px-2 text-muted">Twitter</a></li>  
  
 </ul>  
 <p class="text-center text-muted">© 2023 Тарас Драк</p>  
 </footer>  
</div>  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>  
<script src="https://cdn.jsdelivr.net/npm/@fancyapps/ui@4.0/dist/fancybox.umd.js"></script>  
<script src="https://unpkg.com/swiper@7/swiper-bundle.min.js"></script>  
  
</body>  
</html>