

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ВЕБЗАСТОСУНКУ ІНТЕРНЕТ-МАГАЗИНУ ВЗУТТЯ**  
(тема)

Виконав:  
здобувач 4 року навчання,  
групи ІТІНФ-21-3  
Бочарніков І. В.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник ст. викл. Путятіна О.Є.  
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики \_\_\_\_\_  
(підпис)

Кобилін О. А.  
(прізвище, ініціали)

2025 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Бочарнікову Івану Вікторовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Розробка вебзастосунку інтернет-магазину взуття

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 24 травня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, інтернет джерела, мови програмування JavaScript, PHP, мова запитів SQL.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз вебзастосунків інтернет-магазинів.

2. Вибір програмних рішень вебзастосунку.

3. Розробка вебзастосунку.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми вебзастосунків, огляд існуючих вебзастосунків, постановка задачі, проектування та розробка вебзастосунку.

---



---



---



---



---



---



---



---

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка методу	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ ст. викл. Пуятіна О.Є.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 55 с., 38 рис.

ВЕБЗАСТОСУНОК, ВЗУТТЯ, МОВА ПРОГРАМУВАННЯ JAVASCRIPT, МОВА ГІПЕРТЕКСТОВОЇ РОЗМІТКИ HTML5, КАСКАДНІ ТАБЛИЦІ СТИЛІВ CSS, МОВА ПРОГРАМУВАННЯ PHP.

Об'єктом роботи є створення вебзастосунку інтернет-магазину взуття.

Метою роботи є розробка зручного та багатофункціонального вебзастосунку, завдяки якому можна обирати взуття не виходячи з дому.

Вебзастосунок призначений для користування як зареєстрованим клієнтам, так і незареєстрованим. Клієнт має змогу продивлятися варіанти взуття, але тільки зареєстрований клієнт може купити товар.

WEB APPLICATION, FOOTWEAR, JAVASCRIPT PROGRAMMING LANGUAGE, HTML5 HYPERTEXT MARKUP LANGUAGE, CASCADING STYLE SHEETS CSS, PHP PROGRAMMING LANGUAGE.

The object of the work is to create a shoe web application.

The purpose of the work is to develop a convenient web application that allows you to choose shoes without leaving your home.

The web application is intended for use by both registered and unregistered customers. Customers can browse through the shoe options, but only registered customers can buy the goods.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	6
Вступ.....	7
1 Аналіз предметної галузі і постановка задачі .....	8
1.1 Загальні відомості з предметної галузі.....	8
1.2 Переваги та недоліки вебзастосунку інтернет-магазину взуття.....	9
1.3 Приклади вебзастосунків інтернет-магазину взуття.....	10
1.4 Постановка задачі .....	18
2 Обґрунтування програмних рішень вебзастосунку.....	20
2.1 Вибір програмних рішень Front-end частини.....	20
2.1.1 HTML5 .....	20
2.1.2 CSS3.....	22
2.1.3 JavaScript .....	23
2.2 Вибір програмних рішень Back-end частини .....	24
2.2.1 PHP.....	24
2.2.2 SQL .....	27
2.2.3 PHPMyAdmin.....	29
2.3 Математичні та логічні основи роботи вебзастосунку .....	31
2.3.1 Обробка даних на сервері.....	31
2.3.2 Оптимізація SQL-запитів .....	32
2.3.3 Алгоритм обробки форми користувача .....	32
2.3.4 Шифрування паролів .....	33
3 Комп'ютерна модель вебзастосунку інтернет-магазину взуття .....	34
3.1 Обґрунтування вибору середовища програмної реалізації .....	34
3.2 Програмна реалізація клієнтської частини вебзастосунку .....	36
3.3 Реалізація авторизації та реєстрації користувача.....	42
Висновки .....	51
Перелік джерел посилання .....	52

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

HTML – HyperText Markup Language (мова розмітки гіпертексту)

CSS – Cascading Style Sheets (каскадні таблиці стилів)

SQL – Structured Query Language (структурована мова запитів)

API – Application Programming Interface (інтерфейс програмного застосунку)

UX – User Experience (досвід користування)

UI – User Interface (інтерфейс користувача)

## ВСТУП

Інтернет є невід'ємною частиною нашого життя. Кожень день у всесвітній паутині ми натикаємось на вебсайт або вебзастосунок, для пошуку важливої для нас інформації, така як погода перед виходом з дому або гарно провести час, дивлячись фільми або серіали. Багато хто полюбляє шопінг, та йому вкрай важливо слідкувати за новими товарами в магазинах через інтернет.

Створення вебзастосунку магазину взуття дає можливість людям зручно та швидко, не виходячи з дому або роботи продивлятися нові постачання в улюблений магазин та замовити нове взуття не витрачаючи часу на похід до магазину.

З урахуванням поширення тенденції до онлайн-покупок і зручності використання вебплатформ, створення зручного вебзастосунку магазину взуття стає важливим елементом для конкурентоздібності та зростання прибутку для власника магазину.

Актуальність роботи полягає у тому, що в багатьох людей немає часу або змоги відвідувати фізичні магазини взуття, та їм необхідна альтернатива. Вебзастосунок інтернет-магазину взуття дає таким людям змогу не витрачаючи час на походи до фізичного магазину замовити взуття через вебзастосунок швидко та зручно.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ

## 1.1 Загальні відомості з предметної галузі.

Вебзастосунок інтернет-магазину взуття – це магазин, який спеціалізується на продажі відомих або не дуже відомих брендів взуття. Основна функція такого магазину – зручність та швидкість вибору товару для клієнта та підвищення продажів.

Інтернет-магазин має різноманітні функції, як онлайн-допомога консультанта. Також особистий кабінет, де є інформація про поточне замовлення, система бонусів та знижок.

Можна зробити висновок, що дана галузь діяльності стала необхідною складовою життя людей. Багато магазинів робрили свої вебзастосунки для пошуку взуття, щоб спростити життя багатьом людям. Також власники цих магазинів володіють командами людей для підтримки роботи застосунку, таких як, технічних спеціалістів, які підтримують технічну роботу вебзастосунку та консультантів, які за пару хвилин зможуть відповісти Вам на всі необхідні питання.

Вебзастосунок дуже зручна річ в тому плані що, на відміну від класичних десктопнів застосунків, не потребують завантаження на пристрій. Тим самим економлять Ваш час та пам'ять пристрою. Вам нічого не потрібно, крім інтернету, яким володіє майже кожна людина в сучасному світі, та веббраузеру, який зазвичай вже встановлений за замовченням у будь-який пристрій.

Основна перевага вебзастосунку над десктопним застосунком також полягає в тому, що для оновлення вебзастосунку потрібно тільки оновити застосунок на вебсервері, після чого у користувача буде доступна остання версія.

При використанні вебзастосунку дані користувача будуть збережені на вебсервері, що дає можливість не втратити дані, при, наприклад, несправності пристрою користувача, де дані можуть зберігатися у локальній системі.

Серед переваг варто визначити і недоліки вебзастосунку. Одним з них є прив'язаність до мережі Інтернет. Якщо користувач має погане Інтернет-з'єднання, це може уповільнити процес завантаження вебзастосунку.

Сьогодні, коли навіть дитина має доступ до Інтернету, важко уявити глобальний ринок без сучасних технологій, з якими наше життя стає легшим.

## 1.2 Переваги та недоліки вебзастосунку інтернет-магазину взуття.

В даній сфері єдиним конкурентом вебзастосунку є фізичні магазини взуття, проте з поширенням можливостей доступу в Інтернет вони витісняються цією більш зручною та дешевою альтернативою.

Вебзастосунки мають деякі переваги перед фізичними магазинами, серед яких:

- доступність та зручність. Вебзастосунки можуть бути доступними для людей у будь-якій точці світу в будь-який час. Користувачі можуть шукати взуття, дивитися фотографії та відгуки, ціни тощо;
- широкий асортимент. На вебзастосунках можна знайти різні типи взуття, такі як кросівки, кеди, боксерське взуття тощо;
- зменшення витрат. Вебзастосунки не потребують великих будівель для магазину. Не потрібен гарний ремонт та багато яркого світла щоб покупець міг гарно все побачити та розгледіти товар. Потрібно тільки місце, де зберігати продукцію;
- швидкість для клієнта. Клієнту не потрібно витрачати час на похід до фізичного магазину, а лише зайти на потрібний сайт та обрати що б він хотів.

### 1.3 Приклади вебзастосунків інтернет-магазину взуття

Одним із вебзастосунків інтернет-магазину взуття є «lonza.shoes». На цьому сайті доступно понад 500 варіантів різного взуття на свій смак та колір.

При потраплянні на головну сторінку магазину (рис. 1.1), нас зустрічає приємний для зору інтерфейс, який не ріже зір та має доволі інформативний характер. Перше, що користувач бачить при потраплянні на сторінку – це новинки магазину. При наведенні на цікавий для нас товар з'являється ніби то слайдер (рис. 1.2) з новими фото товару, щоб можна було не переходити за посиланням для перегляду більшої кількості фото. На іконці товару є його назва, ціна, розміри, та можливість додати товар до кошику (рис. 1.3). Дуже зручним також є вибір взуття чоловічого або жіночого у верхньому лівому куті сторінки.

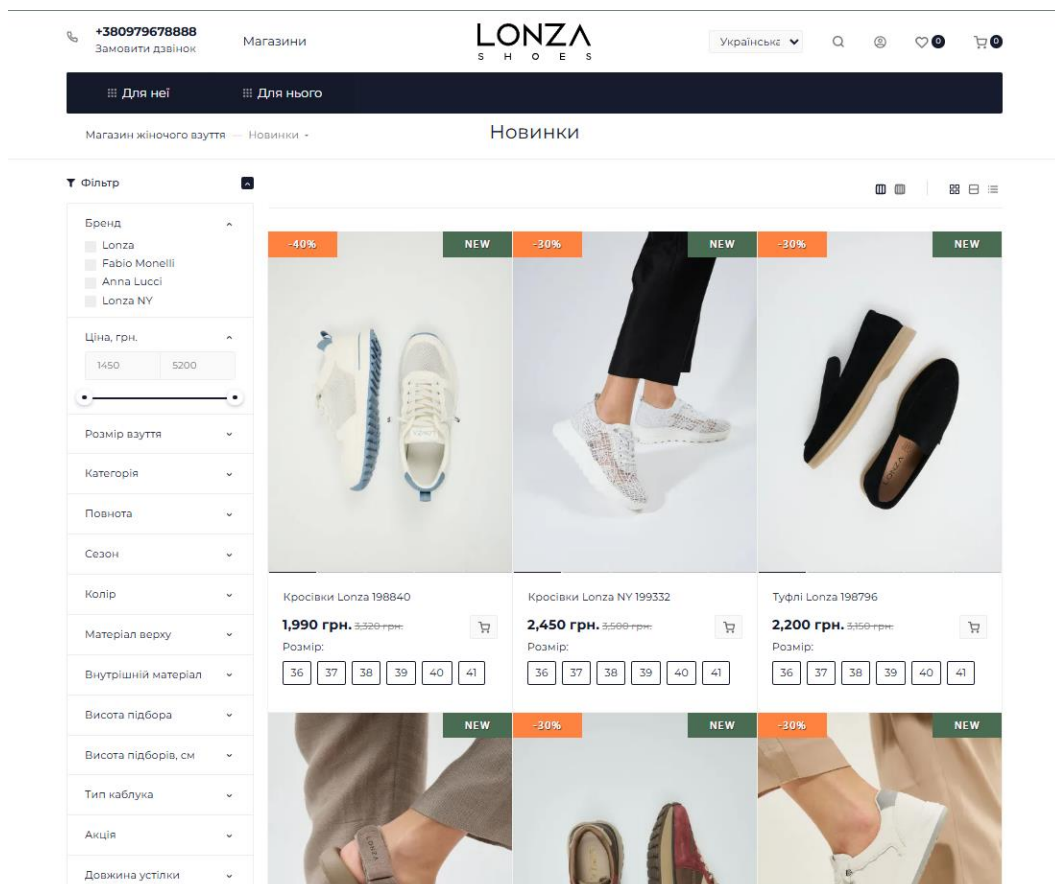


Рисунок 1.1 – Головна сторінка магазину LONZA

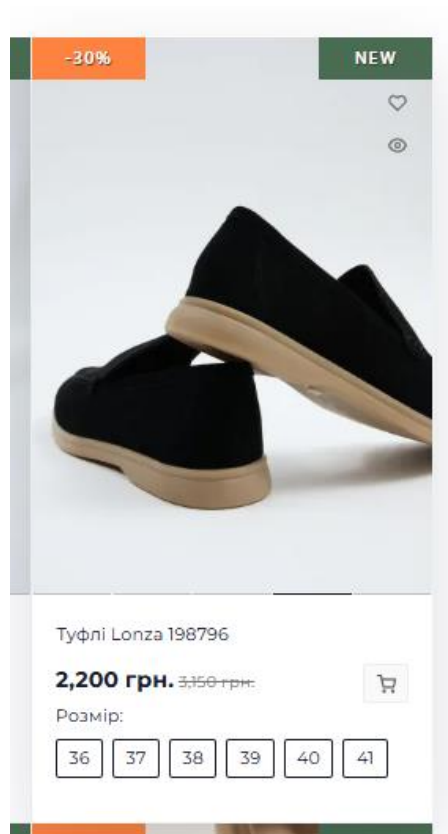


Рисунок 1.2 – Слайдер при наведенні на товар

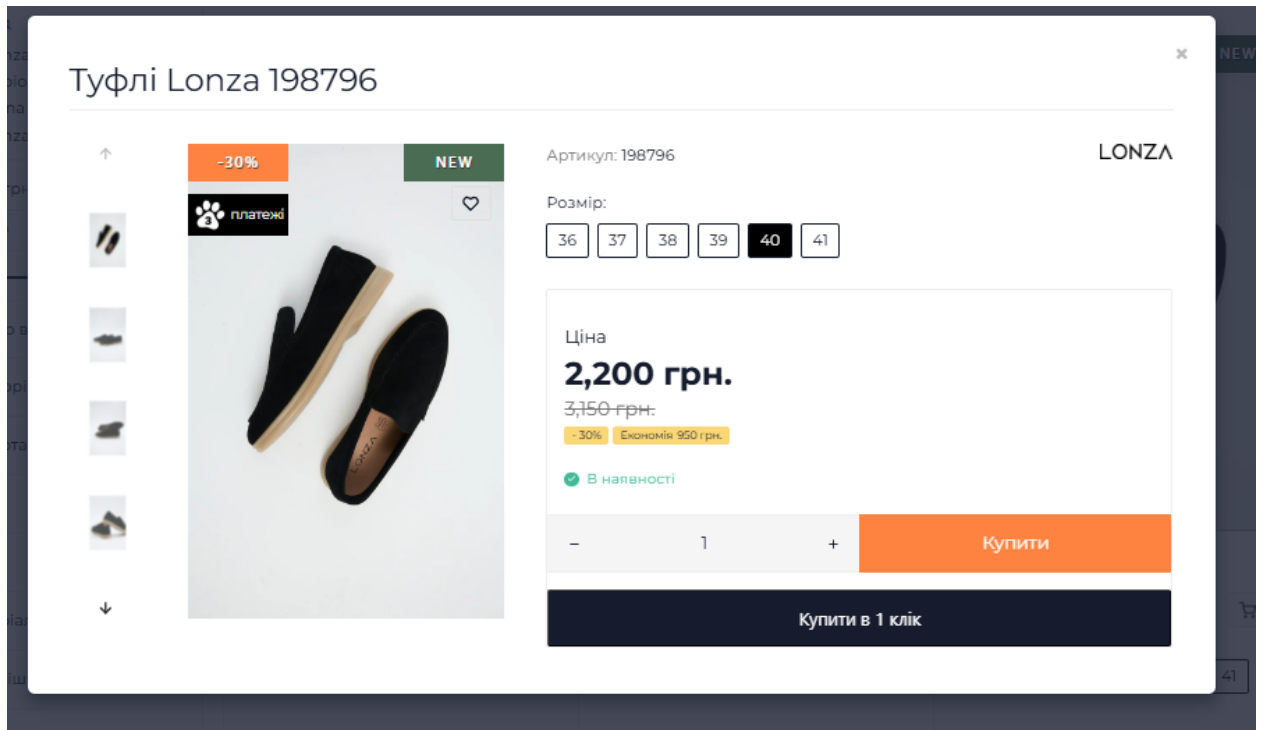


Рисунок 1.3 – Додавання товару у кошик

Якщо чоловік зайшов на сайт та наведе курсор на пункт «Для нього» (рис. 1.4), у нього з'явиться вікно, де він має уточнити, що саме він потребує

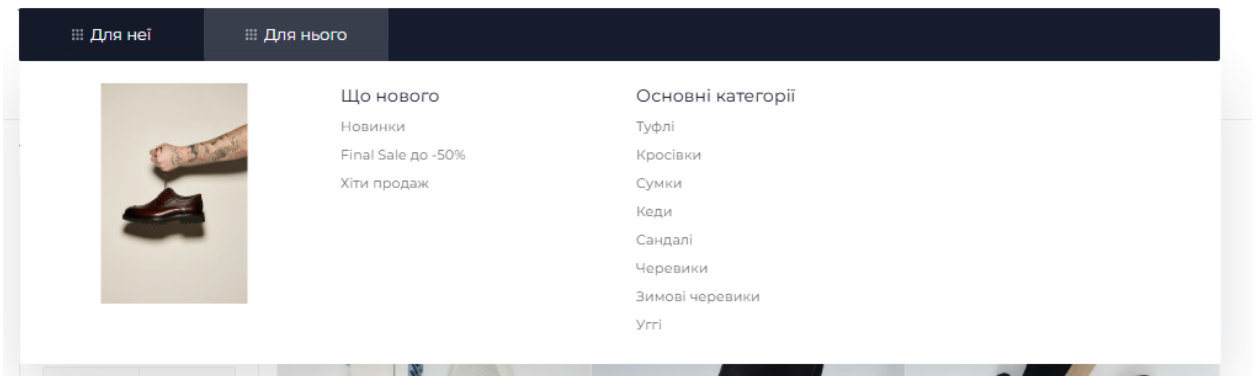


Рисунок 1.4 – Модальне вікно при наведенні

При виборі, наприклад чоловічих сандалей, нам відкриється сторінка із чоловічими сандалями (рис. 1.5).

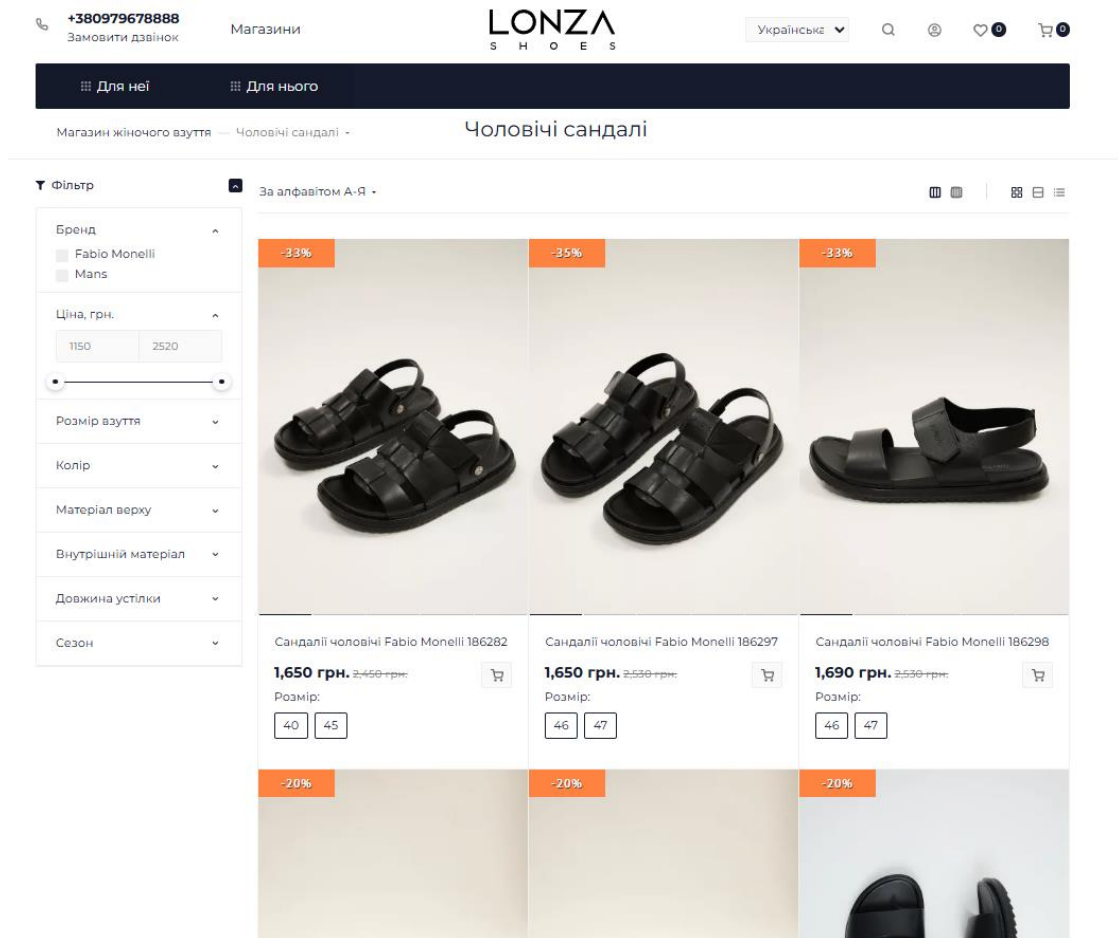


Рисунок 1.5 – Нова сторінка із взуттям

На сторінці можна побачити список товару з фото, цінами та розмірами. У лівому боці маємо вибір фільтрів для пошуку потрібного нам взуття, серед яких бренд, діапазон цін, розмір взуття, колір, матеріал верху, внутрішній матеріал, довжина устілки, сезон на який ви бажаєте носити взуття. При виборі фільтра, наприклад «матеріал верху» – шкіра натуральна (рис. 1.6) сторінка автоматично оновиться та покажуть тільки ті товари, які відповідають фільтру.

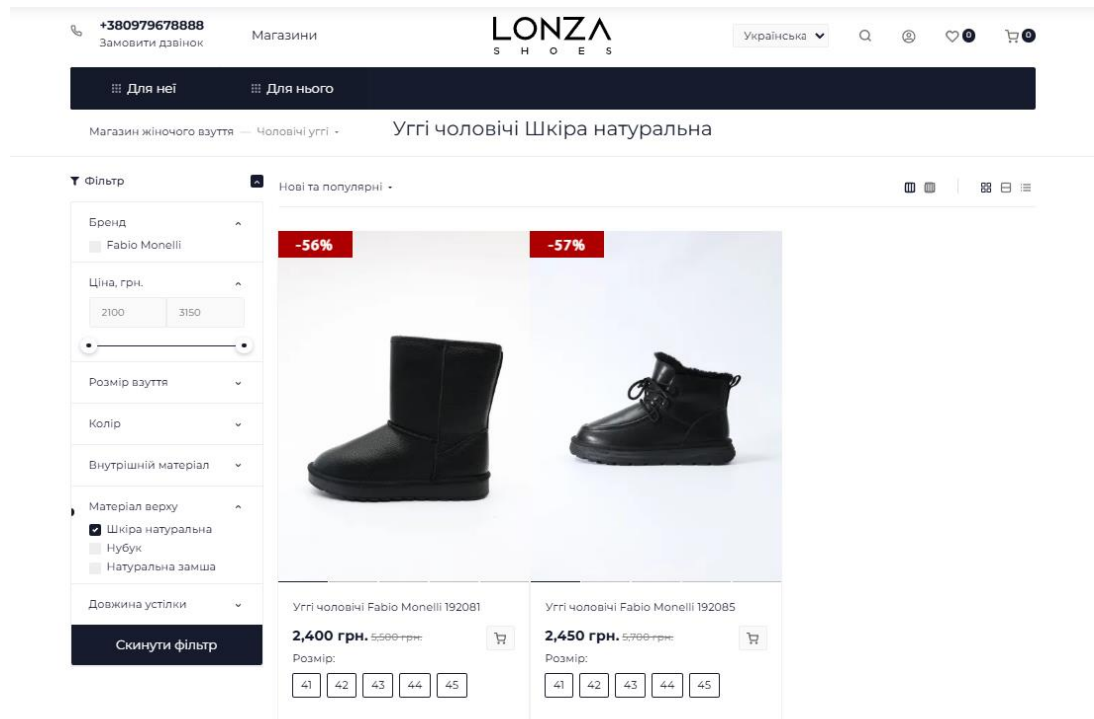


Рисунок 1.6 – Вибір фільтру

Переваги вебзустосунку інтернет-магазину взуття «lonza.shoes»:

- три варіанти відображення товару на сторінці: плитка, таблиця, список;
- можливість додати в обране або кошик;
- онлайн чат з підтримкою;
- багато різноманітнів фільтрів;
- приємний інтерфейс;
- можливість зміни мови магазину;
- особистий кабінет.

Інший приклад вебзастосунку інтернет-магазину взуття є «vittorossi.ua». При переході на головну сторінку (рис. 1.7) користувач може побачити великий блок з інформацією про сезонні знижки, кнопкою зв'язку, та у верхній частині можна обрати для кого взуття, догляд за взуттям (що може бути важливим для багатьох людей які піклуються за своє взуття), місто де є магазини, мова сайту, особистий кабінет. Якщо навести курсор на один з цих пунктів (рис. 1.8), з'явиться меню де можна вибрати підкатегорію яка потрібна користувачу.

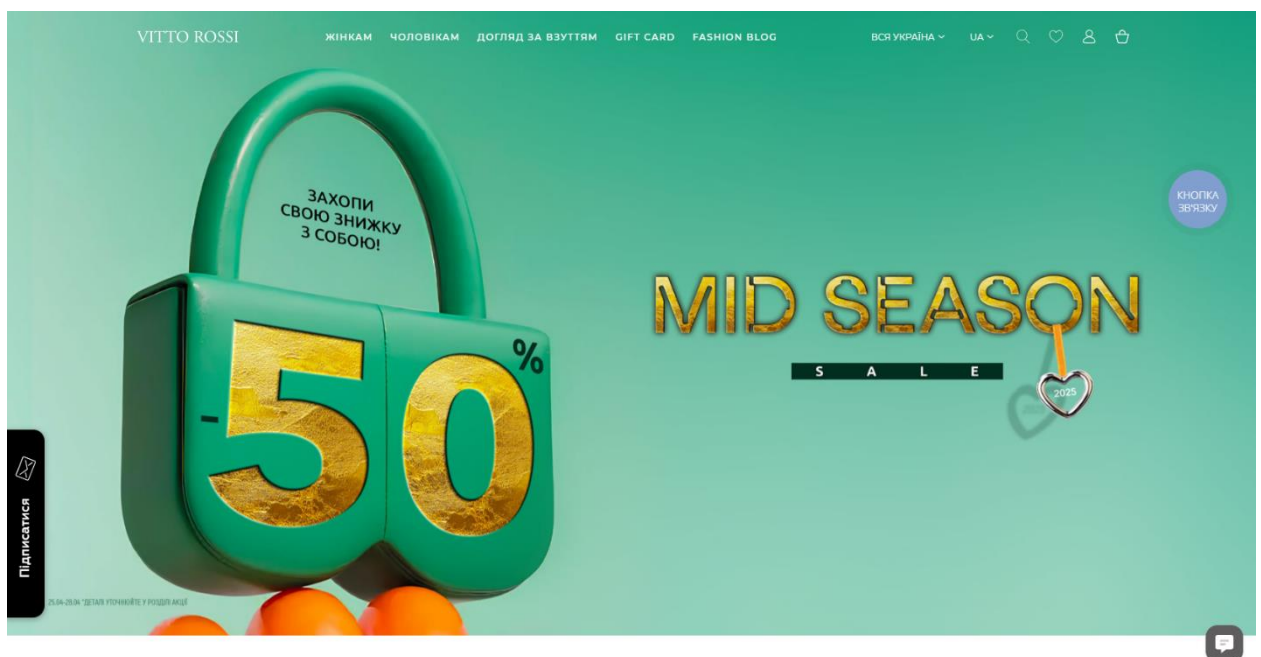


Рисунок 1.7 – Головна сторінка магазину «Vitto Rossi»

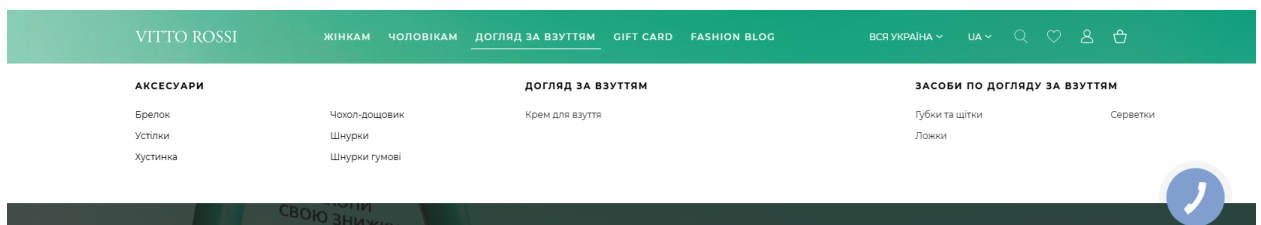


Рисунок 1.8 – Випадаюче меню при наведенні на головній сторінці

Якщо перейти на сторінку чоловічого взуття, наприклад макасини, то користувач перейде на сторінку з цим взуттям (рис. 1.9). На цій сторінці все дуже схоже на магазин «lonza.shoes». Тут є фільтри на любий смак та блоки із

взуттям, який має в собі фото товару, ціну, розміри. Також товари можна відсортувати за новизною, за знижкою, або за ціною. Та неможна змінити розташування плиток взуття, тобто таблицю, плиткою, списком.

Фільтрів товару дуже багато (рис. 1.10), є навіть можливість вибору кольору взуття, але є інші дуже важливі фільтри, як довжина та ширина взуття, бо розмір взуття який носить людина може не співпадати по параметрам ширини або довжини іншого взуття, та буде відчувати дискомфорт при носінні. Також інтересним фільтром є «підходить для» (рис. 1.11), в якому є можливість вибрати для чого саме вам потрібно взуття: для школи, активного відпочинку, весілля тощо. Такий фільтр дуже прискорить час людині, яка не хоче дивитись усі варіанти взуття в магазині, бо вони не підходять для місця, на яке він збирається.

При переході на потрібне взуття, яке клієнт хоче придбати, він переходить на сторінку саме цього взуття (рис. 1.12). Тут можна побачити необхідну інформацію про товар, а саме фото товару, назву, ціну, розміри, характеристики товару. Одразу на цій сторінці є змога «купити в 1 клік», де користувач заповнює свій номер телефону та дуже швидко йому передзвонить менеджер для уточнення замовлення.

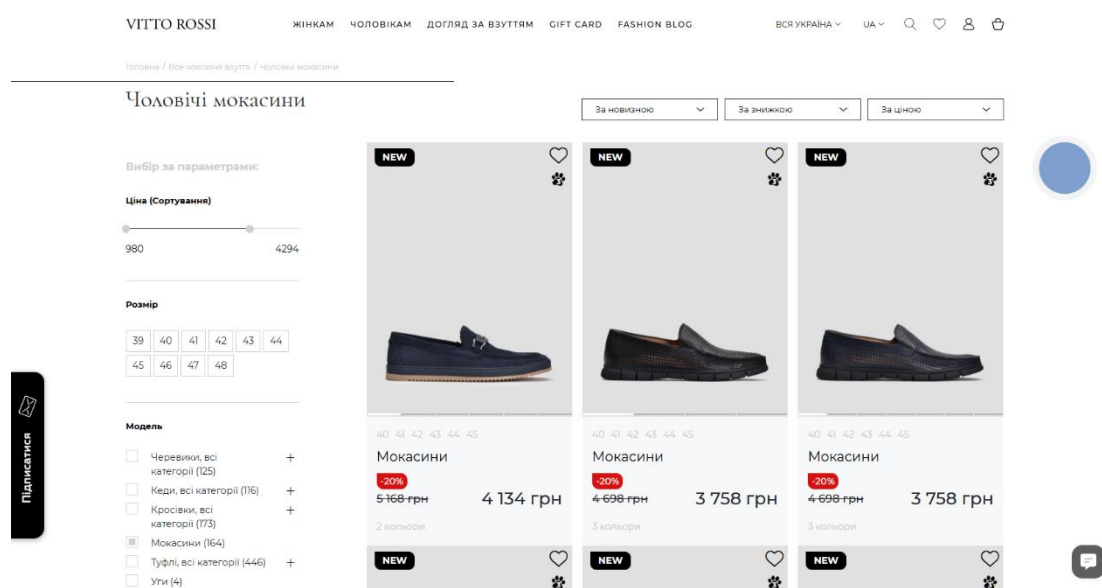


Рисунок 1.9 – Варіації взуття

**Ціна (Середня)**

0 4294

---

**Висота**

39	40	41	42	43
44	45	46	47	48

---

**Модель**

- Червоний, всі категорії (22) +
- Наві, всі категорії (18) +
- Кросівки, всі категорії (12) +
- Мокасени (16)
- Туфлі, всі категорії (1-48) +
- Уліт (4)
- Сандали (8)
- Топсайдери (2)
- Шльопанці (2)

---

**Колір**

Хлоп  Молочний  Червоний  
 Блакитний  Чорний  Коричневий...  
 Білий  Червоний  Синій

[Показати всі](#)

---

**Матеріал об'язування**

- Натуральна (2)
- Натуральна (2)
- Шовк (2)
- Текстиль (1)
- Без матеріалу (28)
- Інше (2)

---

**Матеріал**

- Натуральна (42)
- Текстиль (8)
- Нубук (2)
- Лак (2)
- Замша (4)
- Сеткан (2)
- Нубук/Замша (2)

---

**Висота**

0 см 28 см

---

**Довжина**

0 см 2 см

---

**Ширина**

0 см 8 см

---

**Стан**

---

**Сезонність**

---

**Висказати для**

---

[Показати](#)

Рисунок 1.10 – Фільтри взуття

## Підходить для

- Туризму (0)
- Школи (35)
- Спорту (0)
- Активного відпочинку (4)
- Повсякденного використання (49)
- Випускного (1)
- Заходу (11)
- Весілля (0)

## Показати ще

Показати

Рисунок 1.11 – Фільтр «підходить для»

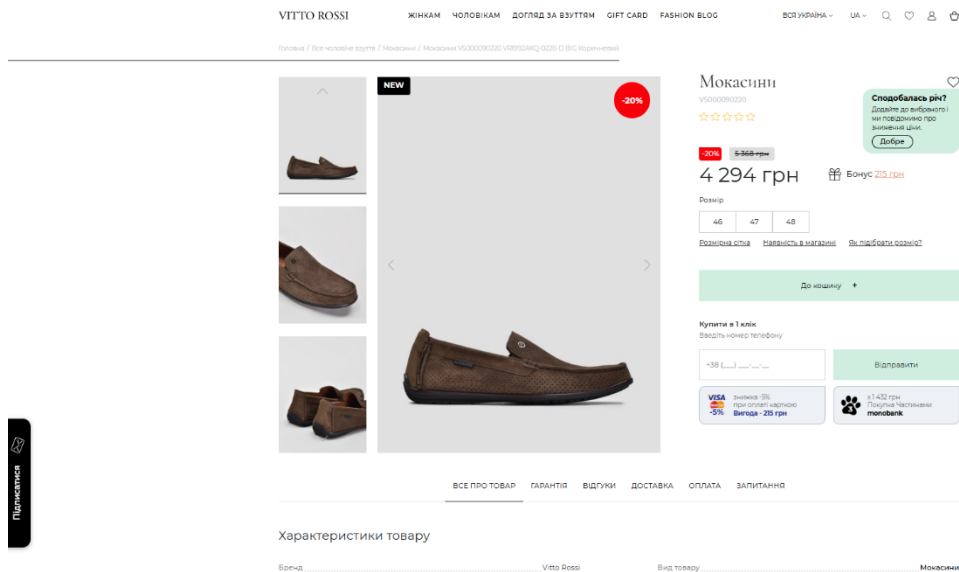


Рисунок 1.12 – Сторінка взуття

Після аналізу інтернет-магазину «vittorossi.ua» можна визначити його переваги:

- можливість швидкого замовлення товару одразу на його сторінці;
- чат або номер телефону для зв'язку з підтримкою або менеджером;
- багато різноманітних фільтрів;
- можливість зміни мови магазину;
- особистий кабінет;
- категорія «догляд за взуттям» з інформацією про догляд або можливістю придбати товари для догляду.

Після перегляду двох прикладів вебзастосунків інтернет-магазинів взуття можна сказати, що вони інтуїтивно зрозуміли на дуже схожі друг на друга. Є відмінність у візуалі, та деяких функціях, щоб підкреслити свою унікальність.

#### 1.4 Постановка задачі

Таким чином, основним завданням є розробити зручний та багатофункціональний вебзастосунок для магазину взуття, який дає можливість швидко та якісно обрати клієнту взуття.

Об'єктом роботи є створення вебзастосунку інтернет-магазину взуття.

Метою роботи є розробка зручного та багатофункціонального вебзастосунку, завдяки якому можна обирати взуття не виходячи з дому.

Головним завданням є створення зручного та багатофункціонального вебзастосунку інтернет-магазину взуття, де клієнт зможе створити особистих кабінет, переглядати список товарів та інформацію про них. Для розробки застосунку буде використано HTML5, CSS3, JavaScript, PHP, SQL.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих вебзастосунків інтернет-магазинів взуття;
- створення UX та UI застосунку;

- розробити технічне завдання;
- розробити вебзастосунок з використанням сучасних технологій;
- провести тестування.

## 2 ОБГРУНТУВАННЯ ПРОГРАМНИХ РІШЕНЬ ВЕБЗАСТОСУНКУ

### 2.1 Вибір програмних рішень Front-end частини

#### 2.1.1 HTML5

Основним програмним рішенням створення вебзастосунку буде мова гіпертекстової розмітки HTML5. Це «скелет» будь-якого сайту. Структура сайту робиться завдяки цій мові.

Цифра 5 у назві це версія. Як будь яка мова програмування, HTML також має версії. Що саме додали у 5 версію? Одним із нововведень була підтримка кросбраузерності (роботи застосунку у різних браузерах). Зазвичай браузери зберігали різні плагіни для програвання мультимедіа. У новій версії HTML додали вбудовану підтримку аудіо та відео, що усуває проблеми із сумісністю. Також були такі оновлення:

- було видалено застарілі елементи, як-от center, font і strike;
- поліпшення правил парсингу зробило його гнучкішим і суміснішим;
- з'явилися нові елементи video, time, nav, section, progress, meter, aside і canvas;
- нові атрибути для інпутів, зокрема email, URL, dates та times;
- нові атрибути, зокрема charset, async та ping;
- нові API з офлайн-кешуванням та підтримкою drag-and-drop тощо;
- підтримка векторної графіки без сторонніх програм типу Silverlight, Flash;
- підтримка MathML поліпшила відображення математичних позначок;
- завдяки JS Web worker API JS тепер може працювати у фоновому режимі;
- підтримка MathML поліпшила відображення математичних позначень;

- завдяки JS Web worker API, JS тепер може працювати у фоновому режимі;
- глобальні атрибути tabindex, repeat та id тепер можна застосовувати до всіх елементів.

Ось деякі переваги HTML5 над HTML:

- частину даних можна зберігати на пристрої користувача, тобто додатки можуть працювати без інтернет-з'єднання;
- вебсторінки можуть відображати більше шрифтів із ширшим діапазоном кольорів, тіней та інших ефектів;
- об'єкти на сторінці можуть рухатися разом із курсором;
- інтерактивні медіа ігор можна запускати просто в браузері без під'єднання додаткового програмного забезпечення або плагінів;
- Для аудіо та відео тепер також не потрібні плагіни;
- браузери можуть відображати інтерактивну 3D графіку за допомогою графічного процесора комп'ютера;
- відкинувши необхідність у сторонніх плагінах, HTML5 прискорює доставку більш динамічного контенту.

Також нова версія HTML5 має важливе значення у забезпеченні семантичної структури вебсторінки. Завдяки новим тегам, як <article>, <section>, <header>, <footer>, <main>, вебсторінки стали більш логічно структурованими. Це покращує доступність контенту для користувачів із порушенням зору. Голосові помічники можуть чіткіше усвідомлювати інформацію з вебсторінки та більш ясно розповісти користувачу зміст сторінки. Також завдяки новим тегам покращилась індексація сайтів пошуковими системами. Нові теги допомагають пошуковим роботам краще розуміти структуру вебсторінки та покращує позиції у результаті пошуку. Це збільшує імовірність того, що при запиті користувача у пошуковій системі першими будуть показані більш точні результати до того, що шукає користувач.

HTML5 підтримує адаптивну верстку, що дає змогу створювати вебсторінки які будуть коректно відображатись на пристроях з різними розмірами екранів, від маленьких смартфонів до великих телевізорів. У поєднанні з CSS та медіа-запитами це забезпечує зручність користування незалежно від типу пристрою.

Завдяки простому синтаксису, HTML5 легко вивчати, що робить його чудовим стартом для початківців у веброзробці. У той же час, з розвитком стандартів і розширенням функціональності HTML5, ця мова залишається актуальною та ефективною для створення сучасних і складних вебсторінок.

### 2.1.2 CSS3

CSS – це мова стилів, що використовується для представлення зовнішнього вигляду документа, написаного на HTML або XML (включно з різними мовами XML, як-от SVG і XHTML). CSS описує, яким чином елемент має відображатися на екрані, на папері, голосом або з використанням інших медіа засобів.

CSS є однією з основних мов вільного веб-розроблення, яка стандартизована специфікацією W3C (The World Wide Web Consortium). Стандарт CSS ділиться на рівні: CSS1 наразі застарів, CSS2.1 – рекомендований для застосування, а CSS3, розбитий на більш дрібні модулі, розвивається на шляху стандартизації.

Якщо HTML5 це «скелет» сайту, то CSS3 це його «одяг» або «стиль». Він відповідає за зовнішній вигляд сайту: кольори, шрифти, відступи, положення елементів, анімації. Також завдяки CSS можна робити адаптивний дизайн вебзастосунку. Це означає, що вебзастосунок буде коректно відображений на пристроях різних розмірів, будь то екран маленького телефону або великий телевізор. Використовуючи класи, ідентифікатори та селектори,

CSS здійснює «каскадне» застосування стилів, що робить його дуже потужним інструментом для веброзробників.

Нова версія CSS3 додала розробникам такі функції як селектори, різні варіації роботи з кольором, скруглення меж блоку, форматування тексту, градієнти, переходи та анімації. Сьогодні найчастіше використовується саме версія CSS3.

### 2.1.3 JavaScript

JavaScript – це високорівнева мова програмування, основною метою якого є створення інтерактивних сторінок вебзастосунків. Якщо HTML «скелет», а CSS «одяг» сайту, то JavaScript це його мозок. Зараз він використовується не тільки в створенні динамічних сторінок, які дозволяють користувачам взаємодіяти із сайтом без зайвих перезавантажень сторінки, а також для розробки серверної сторони застосунку (Back-end), наприклад з використанням Node.js.

Основні особливості JavaScript:

- інтерпретованість: код JavaScript виконується браузером користувача без необхідності попередньої компіляції;
- динамічна типізація: змінні не мають фіксованого типу даних – тип визначається автоматично в момент виконання;
- об’єктна модель: більшість сутностей у JavaScript є об’єктами, що спрощує роботу з даними та структурування програм;
- подієво-орієнтована модель: мова широко використовує обробку подій для створення інтерактивних інтерфейсів;
- асинхронність: JavaScript підтримує неблокувальні операції завдяки колбекам, промісам і конструкції `async/await`;

– прототипне успадкування: об'єкти можуть успадковувати властивості один від одного через ланцюжок прототипів, що відрізняє його від класичного ООП.

Роль JavaScript у веброботці на стороні клієнта (у браузері):

- перевірки введених даних форм;
- зміни структури документа без перезавантаження сторінки (DOM-маніпуляції);
- надсилання та отримання даних із сервера без перезавантаження (AJAX, Fetch API);
- створення анімацій, динамічних меню, слайдерів, спливаючих вікон та інших інтерфейсних рішень;
- реалізації SPA (Single Page Application) за допомогою фреймворків і бібліотек (наприклад, React, Vue, Angular).

На стороні сервера (за допомогою Node.js) JavaScript дає змогу:

- створювати серверні додатки;
- працювати з базами даних;
- обробляти запити клієнтів;
- будувати повноцінні API.

Тож JavaScript є невід'ємною частиною при розробці вебзастосунку. Він забезпечує динамічну взаємодію користувача на сайті, розширює можливості HTML та CSS та відіграє ключову роль у створенні вебзастосунку.

## 2.2 Вибір програмних рішень Back-end частини

### 2.2.1 PHP

PHP (Hypertext Preprocessor) – це скриптова мова програмування, який може бути вбудованим в HTML. Ця мова допомагає розробникам легко створювати динамічно генеруємі сторінки. Автором початкової версії цієї мови є Рasmus Лерддорф, ідея якого полягала у розробці набору інструментів

спрощення процесу створення динамічних вебсторінок. Незважаючи на те, що зараз PHP є мовою загального призначення, найчастіше його використовують як серверний інструмент для генерації HTML коду, який потім інтерпретується веббраузером.

Синтаксис PHP схожий на Java та C++. Ця мова програмування постійно розвивається, на даний момент актуальною версією є 8.3. За статистикою, кожен шостий проєкт створений за допомогою PHP.

Сьогодні виділяють 3 основні області використання мови програмування PHP:

- для написання скриптів та повноцінних вебзастосунків, що виконуються на серверній стороні. Це найпопулярніша сфера застосування, оскільки мова спочатку створювалася саме для веброзробок. Для повноцінної роботи вебзастосунку, написаного на PHP, необхідні сервер, парсер (CGI-додаток) і клієнтське програмне забезпечення (браузер), яке відображає результат виконання коду;

- для створення сценаріїв, що виконуються у командному рядку. Такі міні-програми можуть працювати на будь-якому ПК. Для їх виконання потрібно лише парсер. Оскільки PHP містить потужні інструменти для роботи з рядками, такі сценарії найчастіше створюють для обробки текстових даних;

- для написання графічних інтерфейсів. PHP має багато відгалужень, створених для реалізації різних завдань. Одним із таких відгалужень є PHP-GTK. Його зазвичай використовують програмісти, які звикли до синтаксису PHP.

Ключові переваги PHP:

- простий та інтуїтивно зрозумілий синтаксис. PHP швидко освоюють навіть програмісти-новачки. Він увібрав усі найкращі особливості таких популярних мов, як C, Java та Perl. PHP-код легко читається незалежно від способу використання (будь то для написання невеликих скриптів або створення потужних додатків з використанням ООП (об'єктно-орієнтованого підходу) до реалізації програми);

- кросплатформність та гнучкість. PHP сумісний із усіма популярними платформами (Linux, Windows, MacOS). Написані на ньому програми успішно працюють на різному серверному ПЗ (IIS, Nginx, Apache та багатьох інших);
- відмінна масштабованість. PHP дозволяє досягти максимальної продуктивності додатків, написаних на ньому, зі зростанням апаратних ресурсів. Вебпрограми, розподілені на кілька серверів, здатні справлятися із суттєвими навантаженнями (великим трафіком);
- вбудовуваність у HTML-документи. На просту HTML-сторінку можна легко додати контент, що динамічно змінюється шляхом вставки блоків коду PHP. Вони додаються подібно до HTML-тегів, не порушують структуру документа;
- активний розвиток та вдосконалення. Спільнота розробників постійно працює над впровадженням додаткового функціоналу, що розширює можливості мови, спрощенням синтаксису та покращенням захисту від можливих атак;
- простий пошук рішень проблем, що виникають. В інтернеті існує безліч форумів, присвячених програмуванню на PHP;
- широкі перспективи її подальшого розвитку. Більшість CMS були створені на чистому PHP та фреймворках. Цим зумовлені популярність та затребуваність PHP програмістів.

В багатьох мовах програмування є недоліки, PHP не став виключенням. У цієї мови програмування незважаючи на її переваги є недоліки:

- наявність «дір» у системі безпеки. Багато проблем було вирішено в 8-й версії, проте деякі вразливості, як і раніше, присутні;
- націленість на веброботку. Незважаючи на наявність інших сфер застосування, мова програмування PHP продовжує асоціюватися з Інтернетом;
- деякі протиріччя синтаксису. Оскільки над розвитком PHP працюють програмісти-ентузіасти, деякі їхні дії виявилися неузгодженими. Код містить елементи, характерні для різних мов, а також різні назви вбудованих функцій, у зв'язку з чим може бути плутанина.

Все ж таки, незважаючи на недоліки цієї мови програмування, вона залишається дуже популярною та багато розробників продовжують використовувати її у своїх проєктах.

### 2.2.2 SQL

SQL – це мова запитів для зберігання та обробки інформації в реляційній базі даних. Реляційна база даних зберігає інформацію в табличній формі з рядками і стовпчиками, що представляють різні атрибути даних і різні зв'язки між значеннями даних. Інструкції SQL можна використовувати для зберігання, оновлення, видалення, пошуку та вилучення інформації з бази даних. Можна також використовувати SQL для підтримки та оптимізації продуктивності бази даних.

Рішення SQL було винайдено в 1970-х роках на основі реляційної моделі даних. Спочатку воно було відоме як структурована англійська мова запитів (SEQUEL). Пізніше цей термін було скорочено до SQL. Компанія Oracle, раніше – Relational Software, стала першим постачальником, який запропонував комерційну систему управління реляційними базами даних SQL.

Чому SQL так важливий? Це дуже популярна мова запитів, яка використовується у всіх типах додатків. Аналітики даних і розробники вивчають і використовують SQL тому що це рішення добре інтегрується з різними мовами програмування. Наприклад, вони можуть впроваджувати SQL-запити з мовою програмування Java для створення високопродуктивних додатків обробки даних з основними системами баз даних SQL, такими як Oracle або MS SQL Server.

Одна з головних переваг SQL полягає в його стандартизації та універсальності – ця мова підтримується практично всіма популярними системами управління базами даних, зокрема MySQL, PostgreSQL, Oracle,

Microsoft SQL Server, SQLite та іншими. Це забезпечує зручність переходу між різними СУБД без потреби повного переписування коду.

У мові SQL використовуються різні типи запитів:

- DML (Data Manipulation Language) для роботи з даними (SELECT, INSERT, UPDATE, DELETE);
- DDL (Data Definition Language) для створення й зміни структури таблиць (CREATE, ALTER, DROP);
- DCL (Data Control Language) для управління доступом до даних (GRANT, REVOKE);
- TCL (Transaction Control Language) для керування транзакціями (COMMIT, ROLLBACK).

SQL також підтримує різні типи даних, зокрема числові, текстові, дати, логічні значення тощо. Це дозволяє гнучно моделювати структуру даних у базі даних відповідно для потреб застосунку.

Сьогодні, у більшості сучасних вебзастосунків SQL використовується в парі з сучасними мовами програмування, наприклад, PHP. Зв'язок між PHP та базою даних відбувається через спеціальні функції або бібліотеки, такі як `mysqli` або `PDO`, які дозволяються виконувати SQL-запити безпосередньо з коду. Така інтеграція є ключовою для реалізації динамічних сайтів, де контент формується на основі даних, які є в базі даних.

Крім того, у SQL важливу роль відіграють механізми індексації, нормалізації бази даних і забезпечення цілісності даних (наприклад, через первинні та зовнішні ключі). Такі підходи дозволяють оптимізувати запити, підвищити швидкість доступу до інформації та уникати дублювання даних.

Не менш важливою є й безпека роботи з SQL. Захист від SQL-ін'єкцій – критичний аспект при розробці вебзастосунків. Це досягається шляхом використання параметризованих запитів та перевірки вхідних даних користувача.

Завдяки гнучкості, потужності та широкій підтримці, SQL залишається основним інструментом для збереження та обробки даних у більшості сучасних інформаційних систем.

### 2.2.3 PHPMyAdmin

PHPMyAdmin – це широко використовуваний вебдодаток, що надає графічний користувальницький інтерфейс (GUI) для керування базами даних SQL або MySQL. Він написаний на PHP і дає змогу користувачам виконувати різні завдання, пов'язані з адмініструванням баз даних, таких як створення баз даних, таблиць і запитів, а також імпорт і експорт даних.

Засновником проекту PHPMyAdmin є Тобіас Ратшиллер, програміст із Німеччини, який використовував як фундамент схоже програмне забезпечення MySQL-Webadmin, створене 1997 року. Статус засновника ідеї він передав у 2001 році своєму колезі – програмістові Марку Деліслу, який створив зі своєю командою саме те, чим користуються майже всі, хто займається розробкою вебзастосунків з використанням баз даних.

З PHPMyAdmin користувачі можуть легко взаємодіяти зі своїми базами даних SQL або MySQL без необхідності вручну писати складні SQL-команди. Інтерфейс забезпечує зручне для користувача середовище, в якому користувачі можуть переміщатися по своїх базах даних, таблицях і даних, що робить його важливим інструментом для веб-розробників і адміністраторів баз даних.

Однією з основних функцій PHPMyAdmin є можливість створювати бази даних і керувати ними. Користувачі можуть створювати нові бази даних лише кількома клацаннями миші, вказавши ім'я і набір символів. Після створення бази даних користувачі можуть створювати таблиці в базі даних, визначати їхню структуру і встановлювати відносини між ними. PHPMyAdmin також дає змогу користувачам змінювати наявні таблиці, додаючи або

видаляючи стовпці, змінюючи типи даних або встановлюючи первинні ключі та індекси.

Ще однією важливою функцією PHPMyAdmin є можливість виконувати SQL-запити. Користувачі можуть писати і виконувати SQL-вирази безпосередньо в застосунку, що дає їм змогу витягувати, змінювати і видаляти дані зі своїх баз даних. Ця функція особливо корисна для розробників, які хочуть тестувати свої запити або виконувати складні операції зі своїми даними.

PHPMyAdmin також пропонує низку інструментів для імпорту та експорту даних. Користувачі можуть імпортувати дані з файлів різних форматів, таких як файли CSV або SQL, у свої бази даних. Ця функція зручна під час перенесення даних з однієї бази даних в іншу або під час заповнення нової бази даних наявними даними. З іншого боку, користувачі можуть експортувати дані зі своїх баз даних у різні формати, як-от CSV, SQL або XML, для цілей резервного копіювання або подальшого аналізу з використанням інших інструментів.

Крім того, PHPMyAdmin надає різні інструменти для управління привілеями користувачів і налаштуваннями безпеки. Користувачі можуть створювати облікові записи та керувати ними, призначати різні рівні привілеїв для кожного облікового запису та контролювати доступ до певних баз даних або таблиць. Ця функція гарантує, що тільки уповноважені особи можуть отримувати доступ і змінювати дані, що зберігаються в базах даних.

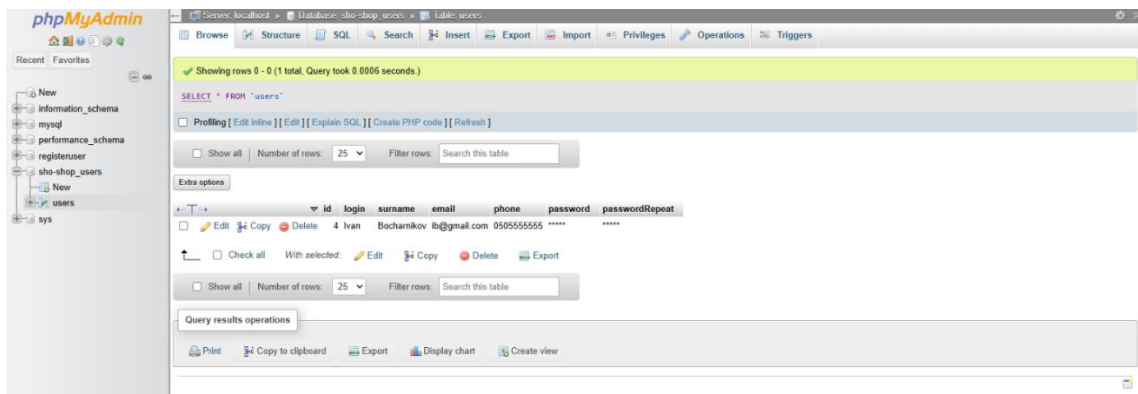


Рисунок 2.1 – Інтерфейс PHPMyAdmin

PHPMuAdmin – це потужний вебзастосунок, що спрощує керування базами даних SQL та MySQL. Він пропонує зручний інтерфейс для створення та управління базами даних, виконання запитів SQL, імпорту та експорту даних і управління привілеями користувачів. Ця програма, залишаючись незмінно зручною і практично не змінюючи свого інтерфейсу, постійно модифікується і поліпшується слідом за появою нових версій PHP та SQL. Повний набір функцій робить його незамінним інструментом для веброзробників і адміністраторів баз даних.

## 2.3 Математичні та логічні основи роботи вебзастосунку

Розробка вебзастосунку передбачає не тільки програмної логіки, а ще низьки математичних понять і формальних виразів, які лежать в основі обробки даних, оптимізації запитів та побудови алгоритмів.

### 2.3.1 Обробка даних на сервері

У Back-End частини під час обробки певних вхідних даних реалізовано перевірку правильності введеної інформації. Наприклад, є форма авторизації до особистого кабінету у вебзастосунку. Ми вводимо пошту та пароль для входу. Під час обробки даних перевіряється наявність такої пошти та паролю у базі даних. Якщо у базі даних існує така пошта та пароль то ми успішно входимо до особистого кабінету. Якщо ж такої пошти або паролю немає, то сервер каже, що такої пошти або паролю немає в базі даних, та входу до особистого кабінету не відбувається. Це відповідає базовій логічній структурі умовного оператора:

$$f(x) = \begin{cases} 1, & \text{якщо } x \in D \\ 0, & \text{інакше} \end{cases}, \quad (2.1)$$

де  $x$  – вхідне значення;

$D$  – множина допустимих значень (наприклад, непорожній рядок, правильна пошта, або число у заданому діапазоні).

### 2.3.2 Оптимізація SQL-запитів

Під час формування SQL-запитів важливо враховувати кількість зв'язків між таблицями у випадку використання операторів JOIN. Теоретично загальна кількість можливих комбінацій записів двох таблиц визначається як:

$$N = n \times m, \quad (2.2)$$

де  $n$  – кількість рядків у першій таблиці;

$m$  – кількість рядків у другій таблиці.

### 2.3.3 Алгоритм обробки форми користувача

При надсиланні форми користувача, наприклад реєстрації або входу, дані проходять через певні етапи обробки. Загальний час виконання запиту можна оцінити як суму окремих операцій:

$$T = t_1 + t_2 + t_3 + \dots + t_i, \quad (2.3)$$

де  $t_i$  – час виконання окремого етапу (перевірка, запис у базу, відповідь сервера тощо).

### 2.3.4 Шифрування паролів

Коли користувач реєструється на вебзастосунку, його персональні дані попадають в дазу даних. Серед цих даних також є пароль. Якщо базу даних взламують, то крадії з легкістю можуть побачити пароль будь-якого користувача вебзастосунку. Щоб такого уникнути, розробники мають робити хешування паролю при реєстрації користувача перед тим, як пароль збережеться у базі даних. Хешування – це перетворення інформації за допомогою математичних формул, в результаті чого виникає хеш – відображення даних у вигляді короткої строки, в ідеалі – унікальний для кожного набору інформації. Функція хешування визначається як:

$$H(m) = h, \quad (2.4)$$

де  $H$  – хеш-функція;

$m$  – вхідне повідомлення (пароль);

$h$  – результат (хеш).

Лістинг 2.1 Функція хешування у мові програмування PHP:

```
$hash = password_hash($password, PASSWORD_DEFAULT);
```

### 3 КОМП'ЮТЕРНА МОДЕЛЬ ВЕБЗАСТОСУНКУ ІНТЕРНЕТ-МАГАЗИНУ ВЗУТТЯ

#### 3.1 Обґрунтування вибору середовища програмної реалізації

Для створення вебзастосунку інтернет-магазину взуття в рамках кваліфікаційної роботи було обрано програмне середовище VS Code.

Visual Studio Code (VS Code) є безкоштовним та відкритим редактором коду, розробленим компанією Microsoft. Він призначений для розробки програмного забезпечення на різних мовах програмування та підтримується на платформах Windows, macOS та Linux.

VS Code активно використовується програмістами. Серед мов програмування, які підтримує середовище є JavaScript, HTML, CSS, PHP, GO, Ruby, Python, C#, TypeScript. Також VS Code підтримує підключення розширень та фреймворків найбільш відомих мов програмування – React JS, Vue.js, LESS, SASS тощо.

Теоретично Visual Studio Code може використовуватися для будь-яких цілей під час розроблення програмних продуктів. На практиці ситуація інакша. Зазвичай редактор застосовується тоді, коли для створення програми не потрібні потужності повноцінної IDE.

VS Code від компанії Microsoft дає змогу писати, форматовувати, а також редагувати код найрізноманітнішими мовами розробки. Він дає можливість швидкого створення проекту зі структурою файлів. Вміє підсвічувати синтаксис коду. Відповідний редактор підтримує можливість швидкого виправлення помилок, а також функції налагодження. Вміє запускати код деякими мовами.

Відмінною рисою VS Code є те, що він легко розширюється. Щоб додати до стандартних опцій нові, достатньо завантажити і встановити

плагін/доповнення з офіційного вбудованого каталогу. Усі інструменти, які підтримує VS Code, як і сам редактор, поширюються абсолютно безкоштовно.

Серед переваг VS Code є вбудована підтримка контролю версій Git, що дозволяє зручно працювати з репозиторіями, виконувати коміти, розділяти гілки та виконувати інші операції Git безпосередньо з редактора.

Недоліки Visual Studio Code:

- важкість для великих проєктів: У порівнянні з іншими інтегрованими середами розробки, VS Code може стикатися з певною тяжкістю при роботі з великими та складними проєктами, оскільки його можливості можуть бути обмежені порівняно з розширеними IDE;

- запуск VS Code може вимагати певних ресурсів системи, особливо при використанні багатьох встановлених розширень або при роботі з великим обсягом коду;

- у деяких випадках, підтримка певних мов програмування або інших технологій може бути менш розвиненою порівняно зі спеціалізованими IDE, що призводить до меншого набору функціональних можливостей;

- недостатня інтеграція з деякими платформами: Хоча VS Code є кроссплатформним, у деяких випадках можуть виникати проблеми з інтеграцією на певних платформах чи середовищах розробки.

Враховуючи ці переваги та недоліки, Visual Studio Code залишається одним із популярних виборів серед редакторів коду для багатьох розробників, особливо для роботи з невеликими та середніми проєктами.

При розробці вебзастосунку було використано: бібліотека jQuery та система SQL.

jQuery – це популярна бібліотека JavaScript, яка значно полегшує роботу з HTML документами, обробкою подій, серверною взаємодією та створенням анімацій на вебсторінках. Вона виступає зручною обгорткою для стандартного JavaScript, що робить код зрозумілішим і спрощує доступ до DOM-елементів.

Завдяки простоті, великому набору можливостей і якісній документації, jQuery здобула широке визнання серед розробників. Вона дозволяє заощадити

час і оптимізувати розробку вебдодатків, особливо коли йдеться про роботу з DOM, реалізацію AJAX-запитів чи анімацій.

SQL – одна з найпоширеніших мов запитів, яка є відкритою та безкоштовною. Вона надає розробникам ефективні інструменти для зберігання, керування та доступу до інформації. SQL використовується як у малих вебпроектах, так і у масштабних корпоративних системах, завдяки своїй надійності, високій швидкодії та ефективності при роботі з даними.

### 3.2 Програмна реалізація клієнтської частини вебзастосунок

Перше, що бачить користувач при переході на вебзастосунок інтернет-магазину взуття – головна сторінка (рис. 3.1). Зверху сторінки можна побачити так названу «шапку», де розташовані назва магазину та кнопки входу до особистого кабінету або налаштувань сторінки. Якщо користувач наведе курсор миши на кнопку входу або налаштувань, її відтінок кольору буде змінений для кращого сприйняття та розуміння, що буде натиснена потрібна користувачу кнопка.

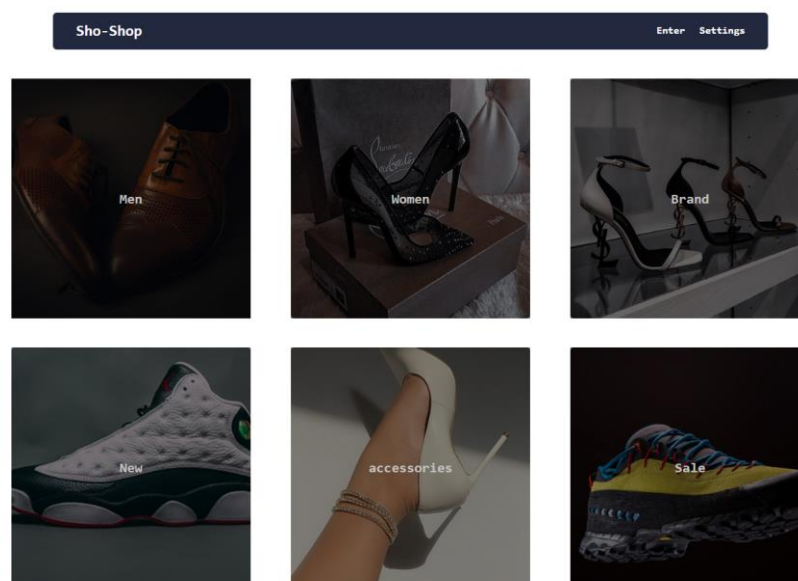


Рисунок 3.1 – Головна сторінка вебзастосунок

Нижче «шапки» розташований блок категорій взуття, де користувач може обрати, яка саме категорія його цікавить. При наведенні курсору на потрібну категорію вона стане яскравішою та пропаде назва категорії, для більш розумілого сприйняття (рис. 3.2).

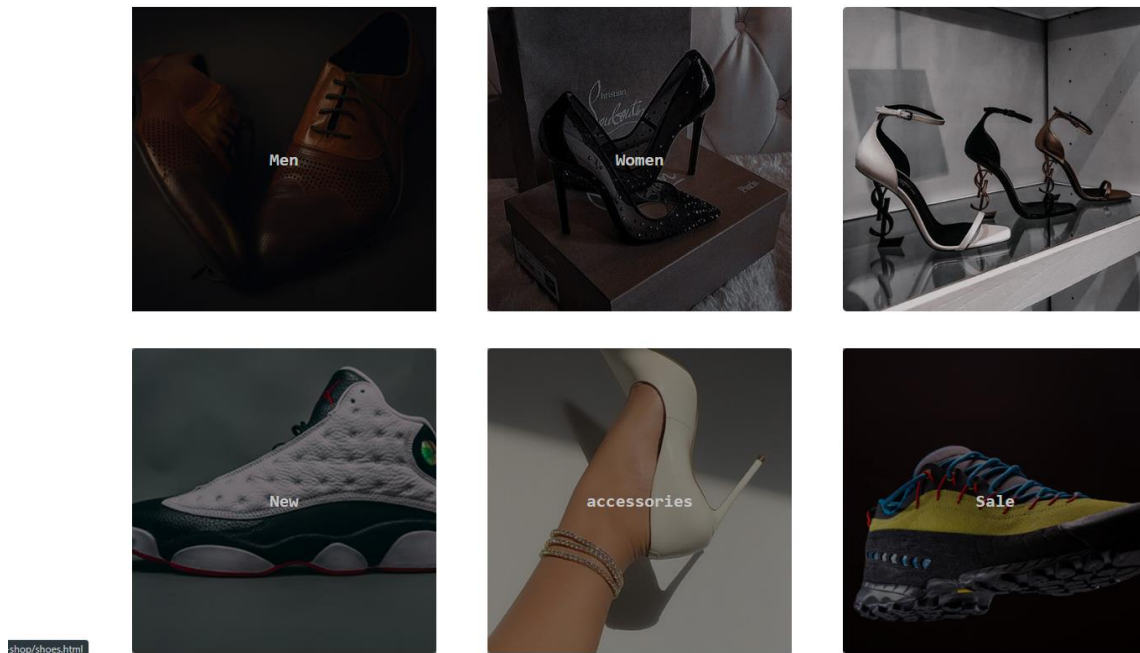


Рисунок 3.2 – Ефект при наведенні на категорію взуття

Після категорій взуття іде блок, в якому розповідається, що це за магазин та інша корисна інформація для клієнтів які потрапили на вебзастосунок вперше (рис. 3.3).

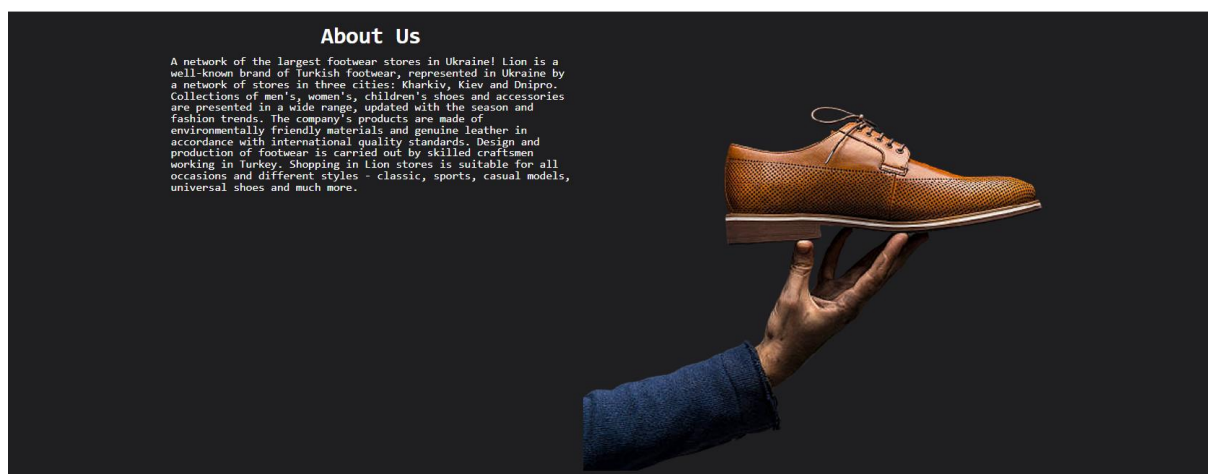


Рисунок 3.3 – Блок з інформацією про магазин

Наступний блок – новини та акції. Тут користувач може побачити актуальну інформацію про акції та зниження цін на взуття. Є фото взуття та великими цифрами написаний відсоток знижки (рис. 3.4). Якщо користувач наведе курсор на зацікавлене взуття, яке й ще по знижці, то буде ефект пропадання напису акції та більш яскраве фото для ознайомлення (рис. 3.5).



Рисунок 3.4 – Блок з новинами та акціями



Рисунок 3.5 – Ефект при наведенні на блок

Останній блок – посилання та контакти магазину (рис. 3.6). Тут користувач побачить корисні посилання, такі як інформація про магазин, контакти, пошта технічної підтримки, відповіді на питання. Цей блок розподілений на 3 розділи:

- компанія. Тут є інформація про магазин, контакти;

- покупцю. Тут можна знайти інформацію про доставку та повернення, часті запитання;
- контакти. Робота електронної пошти, куди покупець може написати свої питання.

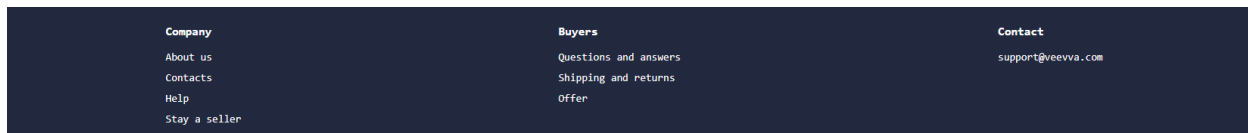


Рисунок 3.6 – Останній блок сторінки

Наступна сторінка вебзастосунку – карточки товару взуття (рис. 3.7). На кожній сторінці буде одна і та ж «шапка» та останній блок з інформацією, тому що це основні блоки, які є актуальними на кожній сторінці. На сторінці з карточками товару можна побачити актуальне взуття яке є в наявності. Для кожного товару є своя карточка, де є фото товару, його назва та ціна. При наведенні курсору на потрібну карточку – колір назви та ціни змінюється на червоний.

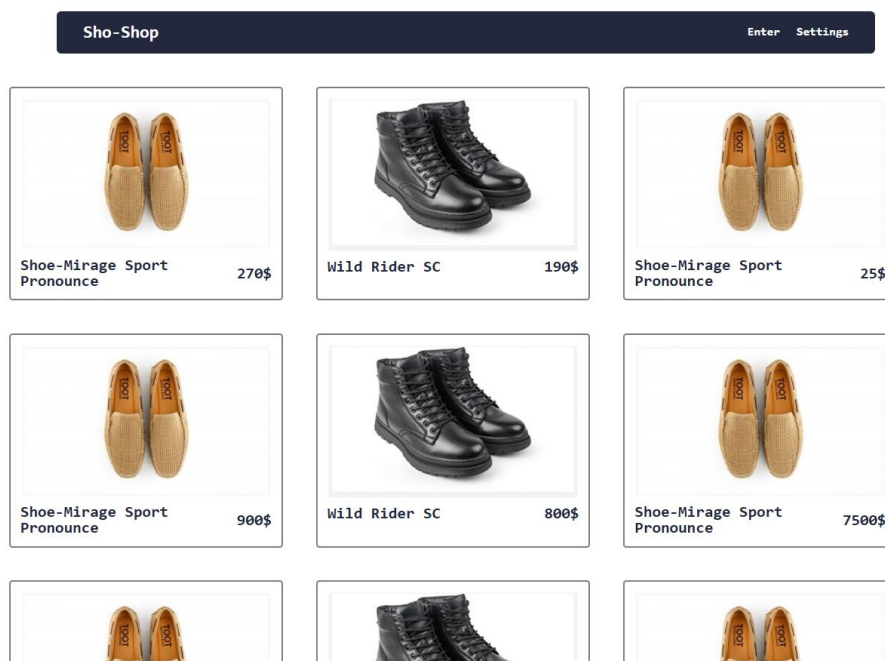


Рисунок 3.7 – Карточки товару взуття

Генерація карточок товару формується з об'єкту у файлі shoesList.js, де внесена необхідна інформація про товари (рис. 3.8). У файлі shoesList.js було створено об'єкт, у якій занесено інформацію про кожний товар, а саме його номер, назва, ціна, зображення, детальний опис.

```

1  const ShoesObject = [
2
3    {
4      id: 0,
5      name: "Shoe-Mirage Sport Pronounce",
6      price: "270$",
7      img: "img/shoes/shoe-2.png",
8      alt: "2",
9      subtitle: "Command the streets in a bold, retro vibe. Blaze Pro fuses vintage suede with a cushioned midsole for smooth moves and effortless fl
10
11    },
12
13    {
14      id: 1,
15      name: "Wild Rider SC",
16      price: "190$",
17      img: "img/shoes/shoe-1.png",
18      alt: "1",
19      subtitle: "Step into the spotlight. Nova Shift combines lightweight build with standout design – made for fast breaks and fast looks."
20
21    },
22
23    {
24      id: 2,
25      name: "Shoe-Mirage Sport Pronounce",
26      price: "25$",
27      img: "img/shoes/shoe-2.png",
28      alt: "2",
29      subtitle: "Push limits with every stride. Storm Rise packs energy return and a sleek frame that turns hustle into highlight."
30
31    },
32
33    {
34      id: 3,
35      name: "Shoe-Mirage Sport Pronounce",
36      price: "900$",
37      img: "img/shoes/shoe-2.png",
38      alt: "2",
39      subtitle: "Bring heat to the blacktop. Court Flow delivers soft landings and a streamlined profile – all style, all drive."
40
41    },
42
43    {
44      id: 4,
45      name: "Wild Rider SC",
46      price: "800$",
47      img: "img/shoes/shoe-1.png",
48      alt: "1",
49      subtitle: "Precision meets swagger. Aero Step's layered sole and crisp lines are built for those who move with intent."
50
51    },
52
53    {
54      id: 5,
55      name: "Shoe-Mirage Sport Pronounce",
56      price: "7500$",
57      img: "img/shoes/shoe-2.png",
58
59    }
60  ]

```

Рисунок 3.8 – Об'єкт карточок товару

Після того, як користувач відкриває сторінку з карточками товару, працює перевірка що відкрилась потрібна сторінка та іде процес генерації карточок товару на сторінці (рис. 3.9). У цьому коді можна побачити цикл for, який перевіряє кожну властивість (інформацію про товар) об'єкту та вставляє потрібні дані кожної властивості у заготовлений HTML-фрагмент, який потім можна побачити на сторінці з карточками товару.

При натисканні на потрібний користувачу товар буде відкрито нову сторінку з інформацією про його (рис. 3.10).

```

if (window.location.pathname.endsWith('shoes.html')) {
  const shoesContainer = document.querySelector('.shoes__container');

  for (let i = 0; i < ShoesObject.length; i++) {
    let shoeCard = `<a href="merch.html" class="shoes__block shoes-block" id="${ShoesObject[i].id}">
      <div class="shoes-block__img">
        
      </div>
      <div class="shoes-block__subblock">
        <p class="shoes-block__name">${ShoesObject[i].name}</p>
        <p class="shoes-block__price">${ShoesObject[i].price}</p>
      </div>
    </a>`;
    shoesContainer.insertAdjacentHTML('beforeend', shoeCard)
  }
}

```

Рисунок 3.9 – Генерація карточок товару на сторінку

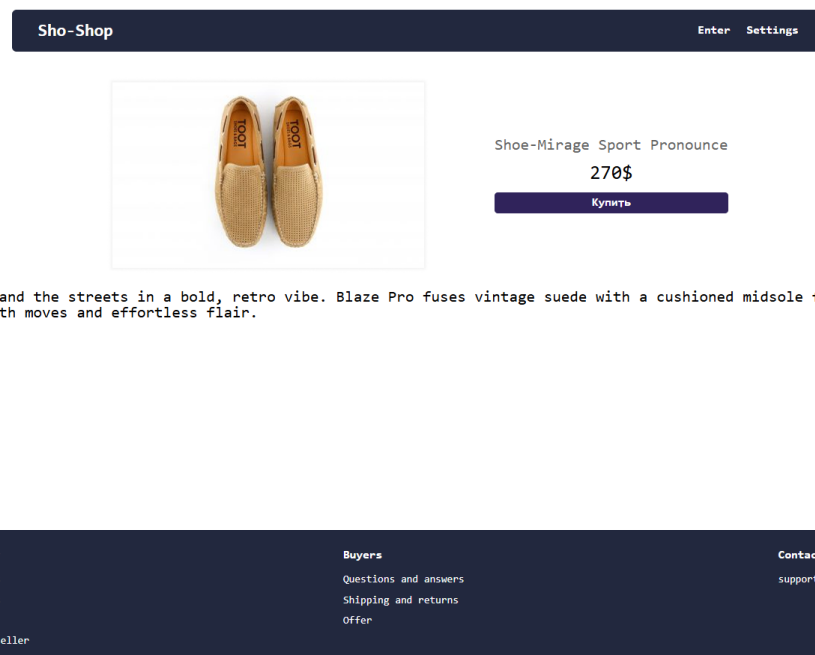


Рисунок 3.10 – Повна інформація про товар

Генерація нової сторінки відбувається таким чином, що на сторінці з карточками при наведенні на товар, його номер зберігається у локальне сховище (localStorage) (рис. 3.11). За допомогою номеру товару який був збережений у локальне сховище формується сторінка з повною інформацією про товар. У коді, який генерує нову сторінку можна побачити цикл for, який записує номер товару у локальне сховище при наведенні на карточку товару. За допомогою цього номеру можна знайти іншу інформацію про товар у об'єкті. Після цього за допомогою HTML-фрагменту заготовленого блоку про товар можна внести потрібну інформацію з об'єкту та буде виведено готову сторінку товару.

```

const shoesBlockId = document.querySelectorAll('.shoes__block');

let merchId;

for (let i = 0; i < shoesBlockId.length; i++) {
  const block = shoesBlockId[i];
  block.addEventListener('mouseover', function (e) {
    merchId = e.target.getAttribute('id');
    localStorage.setItem('idOfMerch', merchId);
  })
}

if (window.location.pathname.endsWith('merch.html')) {
  const shoeId = localStorage.getItem('idOfMerch');

  const merchContainer = document.querySelector('.merch__container');

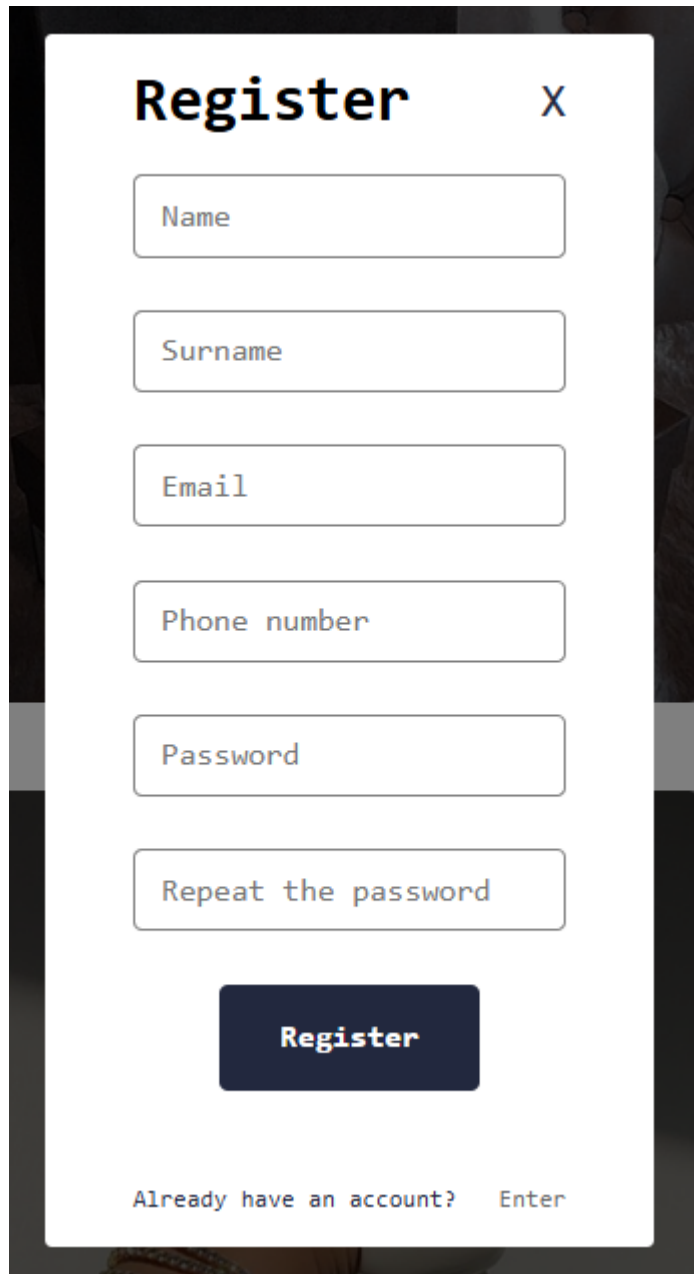
  let merchCard = `<div class="merch__blocks">
    <div class="merch__top">
      <div class="merch__img">
        
      </div>
      <div class="merch__info">
        <div class="merch__name">${ShoesObject[shoeId].name}</div>
        <div class="merch__price">${ShoesObject[shoeId].price}</div>
        <a href="#" class="merch__btn">Купити</a>
      </div>
    </div>
    <div class="merch__subtitle">${ShoesObject[shoeId].subtitle}</div>
  </div>`;
  merchContainer.insertAdjacentHTML('beforeend', merchCard)
}

```

Рисунок 3.11 – Фрагмент коду для генерації сторінки товару

### 3.3 Реалізація авторизації та реєстрації користувача

Якщо користувач бажає створити особистий кабінет, то необхідно у «шапці» сторінки натиснути на кнопку «Enter», після чого з'явиться модальне вікно де треба натиснути кнопку «Register» і з'явиться модальне вікно для реєстрації (рис. 3.12). В цьому вікні можна побачити поля для введення особистих даних, серед яких є ім'я, прізвище, електронна пошта, номер телефону, пароль, повтор пароля. Після введення усіх даних потрібно натиснути кнопку «Register» для успішної реєстрації особистого кабінету. Якщо користувач заповнює не всі поля реєстрації та натисне кнопку реєстрації, то він побачить помилку про незаповнене поле, яке буде мати червону границю для залучення уваги (рис. 3.13).



**Register** X

Name

Surname

Email

Phone number

Password

Repeat the password

**Register**

Already have an account? Enter

Рисунок 3.12 – Модальне вікно реєстрації

Якщо користувач заповнив всі поля, але паролі не будуть співпадати, то буде показану помилку, яка свідчить про це (рис. 3.14).

При введенні даних, якщо всі поля заповнені та паролі співпадають, система буде шукати електронну пошту яку написав користувач у базі даних. Якщо така пошта вже буде існувати у базі даних, то користувач побачить помилку (рис. 3.15). Якщо ж всі дані введені коректно та написаної електронної пошти не буде знайдено в базі даних, то реєстрація буде успішно завершена (рис. 3.16). Після успішної реєстрації особистого кабінету,

користувач може закрити модальне вікно реєстрації або натиснути кнопку «Enter», щоб увійти у щойно зареєстрований особистий кабінет.

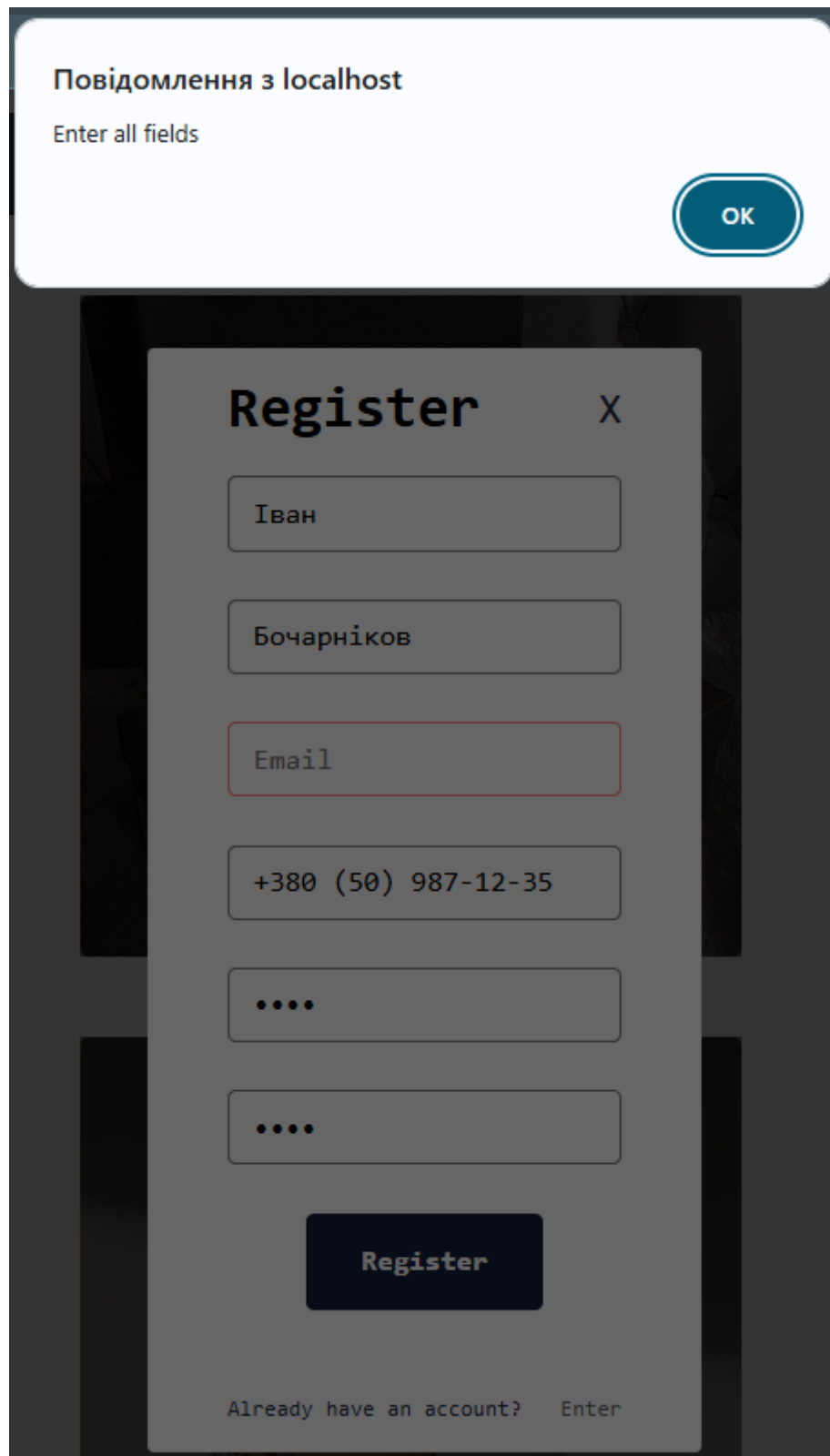


Рисунок 3.13 – Повідомлення про незаповнені поля

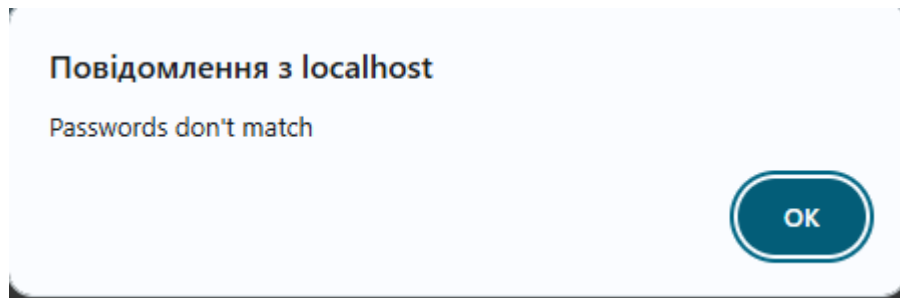


Рисунок 3.14 – Повідомлення про розбіжність пароля

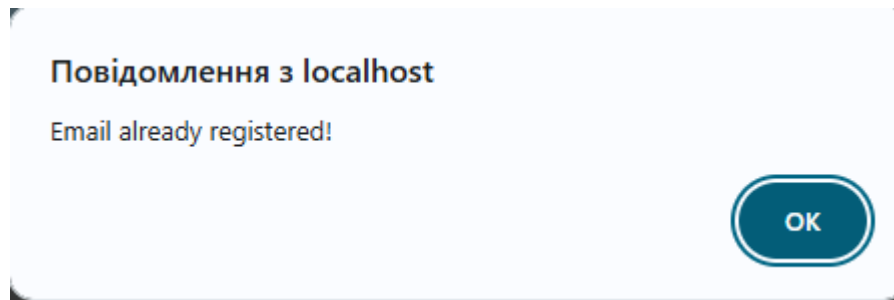


Рисунок 3.15 – Повідомлення про існуючу електрону пошту

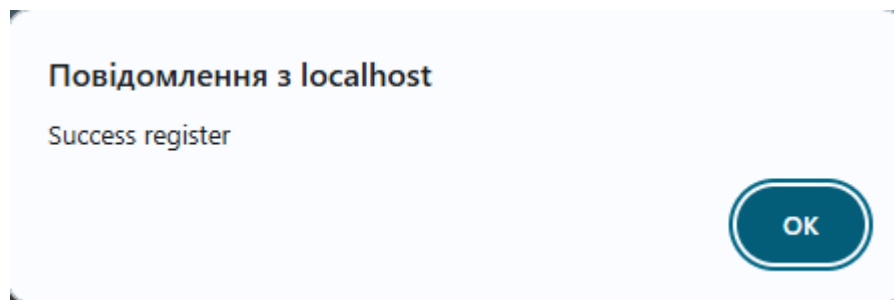


Рисунок 3.16 – Повідомлення про успішну реєстрацію

У коді файлу «register.php» можна побачити наглядно як працює процес реєстрації нового користувача (рис. 3.17). Спочатку іде з'єднання з базою даних у файлі «db.php», де ми записуємо дані про базу даних, а саме назву сервера, логін та пароль від бази даних. Потім іде перевірка з'єднання, після якої у випадку помилки, вона буде виводитись на екран. Якщо все вказано вірно та з'єднання успішне, буде продовжено код у вигляді процесу реєстрації (рис. 3.18).

Після успішного підключення до бази даних буде створено нові змінні з даними, які написав користувач у модальному вікні реєстрації. Після чого

ідуть перевірки на заповнення полів, схожість паролів, перевірка електронної пошти.

```

1 <?php
2 error_reporting(error_level: E_ALL);
3 ini_set(option: 'display_errors', value: 1);
4 require_once('db.php');
5
6 $login = $_POST['name'];
7 $surname = $_POST['surname'];
8 $email = $_POST['email'];
9 $phone = $_POST['phone'];
10 $password = $_POST['password'];
11 $passwordRepeat = $_POST['passwordRepeat'];
12
13 if (empty(($login) || ($surname) || ($email) || ($phone) || ($password) || ($passwordRepeat))) {
14     echo "Enter all fields";
15 } else {
16
17     if ($password != $passwordRepeat) {
18         echo "Passwords don't match";
19     } else {
20
21         $checkQuery = "SELECT id FROM users WHERE email = ?";
22         $stmt = $conn->prepare(query: $checkQuery);
23         $stmt->bind_param(types: "s", var: &$email);
24         $stmt->execute();
25         $result = $stmt->get_result();
26
27         if ($result->num_rows > 0) {
28             echo "Email already registered!";
29         } else {
30             $sql = "INSERT INTO `users` (login,surname,email,phone,password,passwordRepeat) VALUES ('$login', '$surname',
31             if ($conn->query(query: $sql) === TRUE) {
32                 echo "Success register";
33             } else {
34                 echo "Fail: " . $conn->error;
35             }
36         }
37     }
38 }
39 ?>

```

Рисунок 3.17 – Фрагмент коду процесу реєстрації нового користувача

```

1 <?php
2
3 $servername = "localhost";
4 $username = "root";
5 $password = "root";
6 $dbname = "sho-shop_users";
7
8 $conn = mysqli_connect(hostname: $servername, username: $username, password: $password, database: $dbname);
9
10 if(!$conn) {
11     die("Connection failed". mysqli_connect_error());
12 } else {
13     ?
14 } ?>

```

Рисунок 3.18 – Фрагмент коду з'єднання з базою даних

Після успішної реєстрації, особисті дані користувача попадають в базу даних. Ці особисті дані можна переглядати або редагувати через вебдодаток РНРМуAdmin (рис. 3.19). Тут можна редагувати дані користувача якщо це того

потребує, наприклад, користувач змінив електронну пошту. Також можна видалити особистий кабінет користувача за потреби.

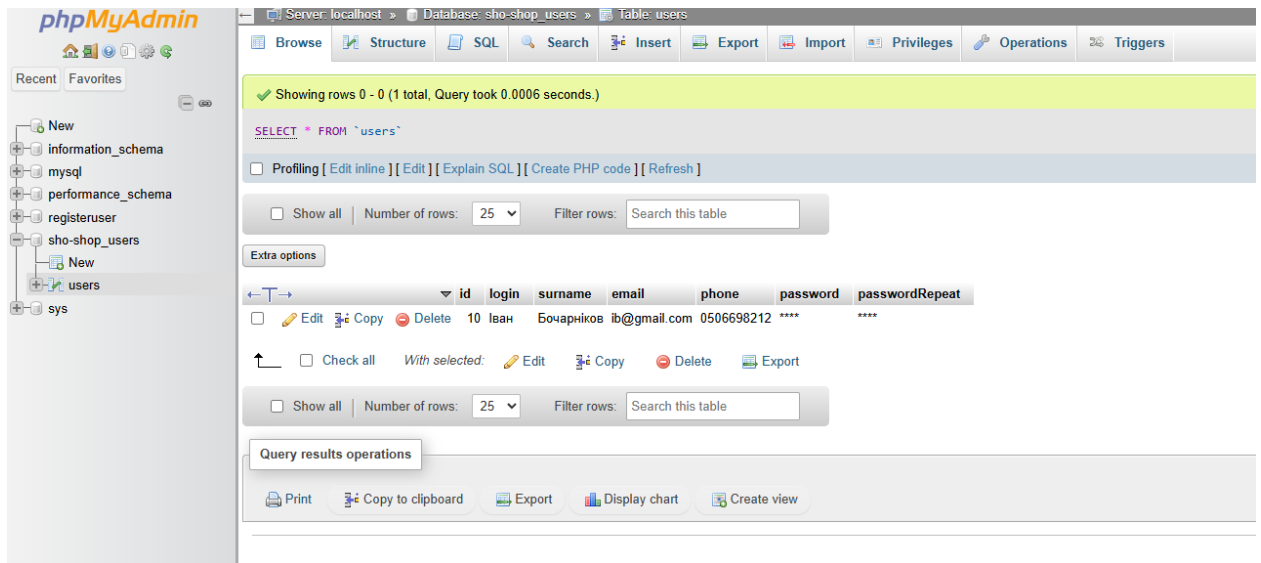


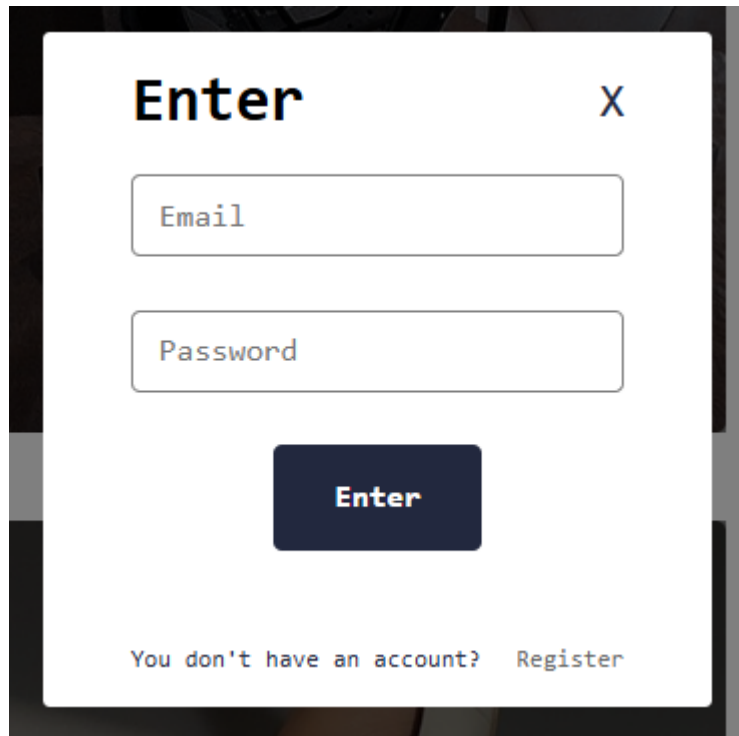
Рисунок 3.19 – Перегляд особистих даних користувачів через РНРMyAdmin

Коли користувач створив новий особистий кабінет, він може одразу увійти до нього, натиснувши кнопку «Enter» як у «шапці» сторінки, так і у модальному вікні реєстрації, після чого буде відкрито нове модальне вікно для входу до особистого кабінету (рис. 3.20).

Для входу до особистого кабінету потрібна тільки електронна пошта та пароль, який був введений при реєстрації кабінету. Якщо заповнити тільки одне поле модального вікна або не заповнити жодного та натиснути кнопку «Enter», то користувач побачить помилку про незаповненні поля (рис. 3.21).

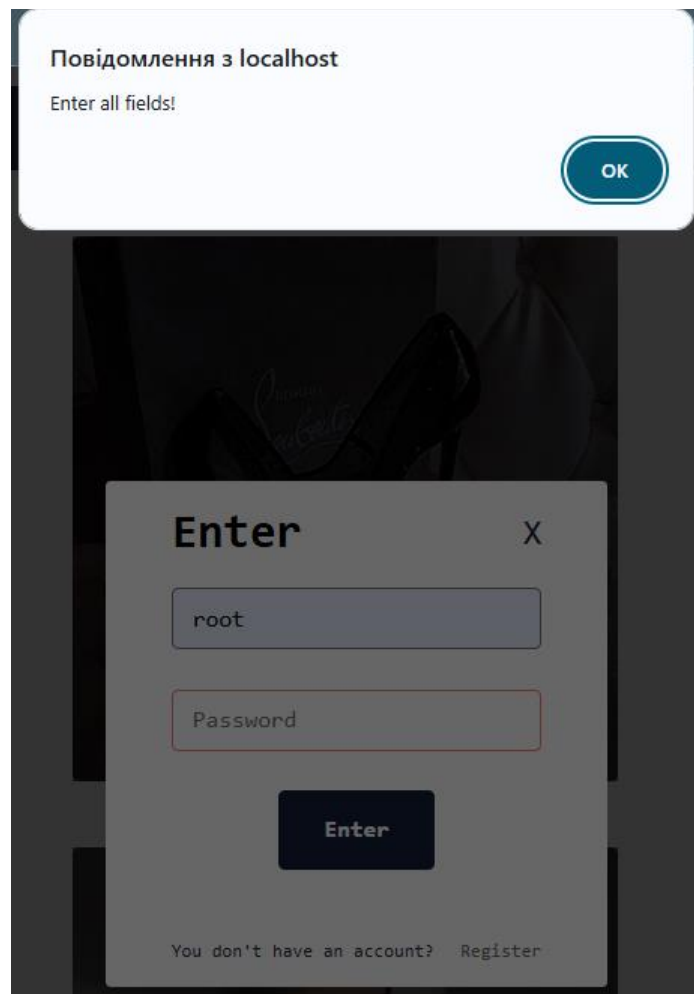
Якщо користувач заповнив усі поля та натиснув кнопку входу, але пошта або пароль не були зареєстровані у базі даних, то з'явиться повідомлення про це (рис. 3.22).

При введенні користувачем електронної пошти, яка існує в базі даних та введенні паролю, що співпадає з тим, що є в базі даних – модальне вікно автоматично закривається, а кнопка входу до особистого кабінету в «шапці» сторінки замінюється на привітання користувачу (рис. 3.23).



The image shows a modal window titled "Enter" with a close button (X) in the top right corner. It contains two input fields: "Email" and "Password". Below the fields is a dark blue button labeled "Enter". At the bottom, there is a link that says "You don't have an account? Register".

Рисунок 3.20 – Модальне вікно входу до особистого кабінету



The image shows a notification box at the top with the text "Повідомлення з localhost" and "Enter all fields!". Below it is a dark blue button labeled "OK". In the background, the "Enter" login modal window is visible, but it is dimmed. The "Email" field contains the text "root" and has a red border. The "Password" field is empty and also has a red border. The "Enter" button and the "Register" link are also visible.

Рисунок 3.21 – Помилка про незаповненні поля

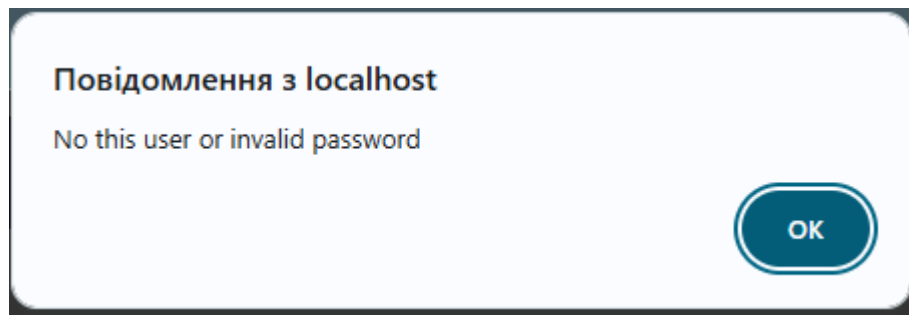


Рисунок 3.22 – Помилка про невірно введенну електронну пошту або пароль

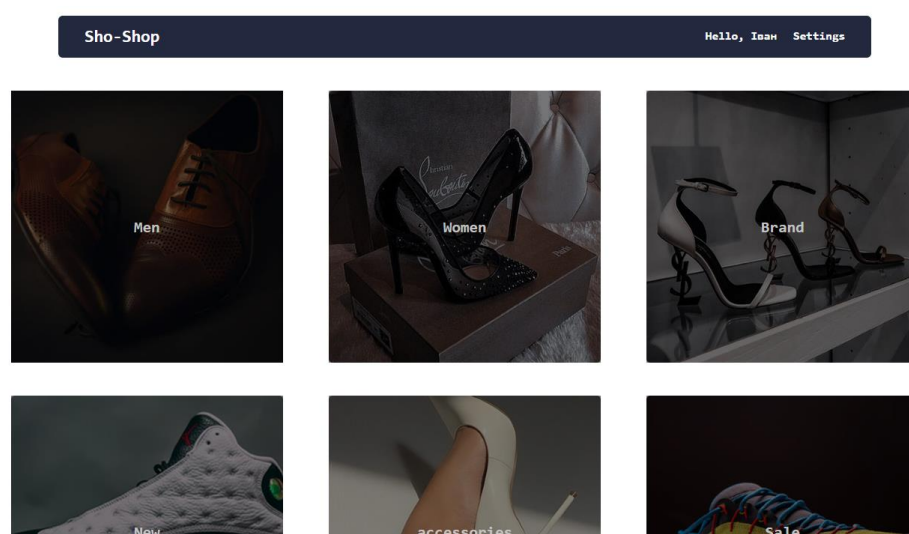


Рисунок 2.23 – Заміна кнопки входу до особистого кабінету на привітання користувачу

Процес входу здійснюється схожим чином, як і реєстрація нового користувача. Спочатку іде підключення до бази даних через файл «db.php». Потім у файлі «login.php» отримується дані, які заповнив користувач у полі електронної пошти та пароля, які записуються у нові змінні (рис. 3.24). Якщо поля були незаповненні – виникає помилка. Якщо дані були введенні – вони перевіряються у базі даних. В позитивному випадку, коли дані вірні, вони відправляються у файл «router.js» для подальшої обробки (рис. 3.25). Якщо електронна пошта або пароль не підходять – користувачу показують помилку.

```

1  <?php
2  require_once('db.php');
3
4  $email = $_POST['email'];
5  $password = $_POST['password'];
6
7  if (empty($_$email) || ($_$password)) {
8      echo "Enter all fields!";
9  } else {
10     $sql = "SELECT * FROM `users` WHERE email = '$email' AND password = '$password'";
11     $result = $conn->query(query: $sql);
12
13     if ($result->num_rows > 0) {
14         while ($row = $result->fetch_assoc()) {
15             echo $row['login'];
16             header(header: 'Content-Type: text/plain');
17         }
18     } else {
19         echo "No this user or invalid password";
20     }
21 }

```

Рисунок 3.24 – Фрагмент коду авторизації у файлі «login.php»

```

219 document.addEventListener("DOMContentLoaded", function () {
220     function sendDataToServer() {
221         const formData = new URLSearchParams();
222         formData.append('email', enterEmail);
223         formData.append('password', enterPassword);
224
225         fetch('login.php', {
226             method: 'POST',
227             headers: {
228                 'Content-Type': 'application/x-www-form-urlencoded',
229             },
230             body: formData
231         })
232             .then(response => response.text())
233             .then(data => {
234                 if (data === "No this user or invalid password" || data === "Enter all fields!") {
235                     alert(data);
236                 } else {
237                     console.log('Login:', data);
238                     document.querySelector('.header__menu-register').textContent = `Hello, ${data}`;
239                     popupClose(document.querySelector('.enterPopup'));
240                 }
241             })
242             .catch(error => {
243                 console.error('Помилка:', error);
244             });
245     }

```

Рисунок 3.25 – Фрагмент коду авторизації у файлі «popup.js»

За допомогою методу POST можна отримати дані із файлу «login.php» у файл «popup.js». При успішній авторизації дані імені користувача попадають у файл «popup.js». В цьому файлі за допомогою HTML-фрагменту можна замінити кнопку входу до особистого кабінету на привітання користувача, що дасть йому зрозуміти, що він успішно виконав вхід до особистого кабінету.

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований вебзастосунок інтернет-магазину взуття за допомогою таких мов програмування, як HTML, CSS, JavaScript, PHP.

Під час виконання роботи було проаналізовано 2 вебзастосунки інтернет-магазину взуття для більшого розуміння їх можливостей. Після аналізу ми стали більше розуміти яким повинен бути вебзастосунок. В процесі роботи було опрацьовано наступні завдання:

- на головній сторінці указано необхідну для клієнта інформацію;
- можливість входу або реєстрації особистого кабінету;
- перегляд кожного товару та детальної інформації.

В результаті роботи ми отримали багатофункціональний вебзастосунок за допомогою якого користувачі швидко та легко можуть обрати та купити взуття на свій смак. Вебзастосунок відповідає сучасним стандартам якості вигляду клієнтського інтерфейсу.

Результати роботи апробовано у вигляді тез доповідей під час XXIX Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті».

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, 5(57).
2. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). *Advances in Spatio-Temporal Segmentation of Visual Data (Vol. 876)*. Springer Nature.
3. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. *Telecommunications and Radio Engineering*, 79(10).
4. Gorokhovatskyi, V., & Tvoroshenko, I. (2023). Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S»* (pp. 25-27).
5. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2023). Statistical data analysis models for determining the relevance of structural image descriptions. *IEEE Access*, 11, 126938-126949.
6. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.
7. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. *Telecommunications and Radio Engineering*, 79(10).
8. Ibrahim, D. Y., Gorokhovatskyi, V., Tvoroshenko, I., & Mujahed, A. D. (2022). Classification of Images Based on a System of Hierarchical Features.
9. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th*

*International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)* (pp. 665-670). IEEE.

10. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

11. Kinoshenko, D., Kobylin, O., Mashtalir, S., & Stolbovyi, M. (2019, March). Metric video retrieval speedup by irrelevant data elimination. In Eleventh International Conference on Machine Vision (ICMV 2018) (Vol. 11041, pp. 176-183). SPIE.

12. Gorokhovatskyi, V. A., Rusakova, N., & Tvoroshenko, I. S. (2020). The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. *Telecommunications and Radio Engineering*, 79(20).

13. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. *International Journal of Advanced Computer Science and Applications*, 11(5).

14. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Vlasenko, N. V. (2020). Using fuzzy clustering in structural methods of image classification. *Telecommunications and Radio Engineering*, 79(9).

15. Yakovleva, O., & Nikolaieva, K. (2020). Research of descriptor based image normalization and comparative analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE descriptors. *Advanced Information Systems*, 4(4), 89-101.

16. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. *International Journal of Advanced Computer Science and Applications*, 11(5).

17. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks. *International Journal of Academic Information Systems Research*, 7(7), (pp. 25-36).

18. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. *International Journal of Advanced Trends in Computer Science and Engineering*. 9(4). – 4701-4706.

19. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60-63). ACM.

20. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

21. Yakovleva, O., Kovtunenکو, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System. In *COLINS* (3) (pp. 69-86).

22. Yakovleva, O., Slyusar, V., Kushnir, O., & Sabovchyk, A. (2021). New trends in scientific and technological revolution (STR) and transformation of science and education systems in the paradigm of sustainable development. In *E3S Web of Conferences* (Vol. 277, p. 06006). EDP Sciences.

23. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, 19(1), 23-28.

24. Gorokhovatskyi, V., Tvoroshenko, I., & Olena, Y. (2024). Transforming image descriptions as a set of descriptors to construct classification features. *Indonesian Journal of Electrical Engineering and Computer Science*, 33 (1), (pp. 113-125)

25. Kuzminska, O., Mazorchuk, M., Morze, N., & Kobylin, O. (2019, June). Digital learning environment of ukrainian universities: The main components to influence the competence of students and teachers. In *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications* (pp. 210-230). Springer, Cham.

26. Кобилін, О. А., & Путятіна, О. Є. (2024). Знешумлення зображень, зіпсованих дробовим шумом, у реальному часі. Системи обробки інформації, (1 (176), 46-51.

27. Gorokhovatskyi , V., Chmutov , Y., Tvoroshenko , I., & Kobylin , O. (2025). Reducing computational costs by compressing the structural description in

image classification methods. *Advanced Information Systems*, 9(1), 5–12.  
<https://doi.org/10.20998/2522-9052.2025.1.01>

28. Кобилін, О., Вечірська, І., Афанасьєв, А. (2024). Аналіз існуючих моделей глибинного навчання в задачах обробки природної мови. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 63–76, <https://doi.org/10.32782/IT/2024-3-7>

29. Кобилін, О., Вечірська, І., Кравченко, О. (2024). Порівняння нейронних мереж типу RNN та LSTM. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 97–107, <https://doi.org/10.32782/IT/2024-3-10>

30. Vechirska, I., Kobylin, O., Prokopiev, S., Vechirska, A., & Kucherenko, M. (2022). Building a logical network for solving the problem of car rental by means algebra of finite predicates. *Computer Systems and Information Technologies*, (2), 78–87. <https://doi.org/10.31891/csit-2022-2-9>

31. Kobylin, O.; Putiatina, O. (2025). Some aspects of real-time image denoising influenced by shot noise and compound Poisson noise. *CEUR Workshop Proceedings*, volume = 3943, 109-117

32. Бочарніков І. В. (2025) *Методи статистичного аналізу даних. Радіоелектроніка і молодь у XXI столітті: тези доповідей 29-го Міжнародного молодіжного форуму (Харків, 16–19 квітня 2025 р.)*. Харків: ХНУРЕ, 2025. Т. 7. С. 23-24.