

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп'ютерної інженерії та кібербезпеки

**Звіт**  
**з виробничої практика**

студента 3 курсу групи КБ-20-1  
спеціальності 125 «Кібербезпека»

Фоміна Владислава Васильовича

(прізвище, ім'я та по-батькові)

Керівник ст. викладач кафедри КІ та КБ  
Окунькова О.О.

Дата захисту: " " \_\_\_\_\_ 2023 р.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії

_____	<u>О. О. Окунькова</u>
(підпис)	(прізвище та ініціали)
_____	<u>Семенець С. П.</u>
(підпис)	(прізвище та ініціали)
_____	<u>Воротнікова В. В.</u>
(підпис)	(прізвище та ініціали)

Житомир – 2023

## ЗМІСТ

<b>РЕФЕРАТ.....</b>	<b>3</b>
<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ 1. ВІДОМОСТІ ПРО МІСЦЕ ПРОХОДЖЕННЯ ПРАКТИКИ .....</b>	<b>5</b>
<b>1.1 Sana Commerce. ....</b>	<b>5</b>
<b>1.2 Основні види діяльності Sana Commerce.....</b>	<b>5</b>
<b>1.3 Ступінь розвитку інформаційної системи підприємства.....</b>	<b>8</b>
<b>1.4 Програмне забезпечення, яке використовується на підприємстві.....</b>	<b>8</b>
<b>РОЗДІЛ 2. ХАРАКТЕРИСТИКА ТА ПЕРЕЛІК ВИКОНАНИХ РОБІТ .....</b>	<b>9</b>
<b>2.1 Адаптивна верстка (HTML, CSS, SCSS).....</b>	<b>9</b>
<b>2.2 Мова програмування JavaScript. ....</b>	<b>21</b>
<b>2.3 Технологія ASP.NET MVC Core .....</b>	<b>28</b>
<b>ВИСНОВКИ.....</b>	<b>41</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>42</b>
<b>ДОДАТКИ.....</b>	<b>43</b>

					ДУ «Житомирська політехніка».23.125.21.000 - ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Фомін В.В.			Виробнича практика	Літ.	Арк.
Перевір.		Окунькова О.О.					4
Керівник						ФІКТ Гр. КБ-20-1	
Н. контр.							
Зав. каф.							
						Аркушів	61

## РЕФЕРАТ

Звіт з виробничої практики складається зі вступу, двох розділів, висновку та списку використаної літератури.

Текстова частина викладена на 42 сторінках друкованого тексту. Список використаних джерел містить 13 найменувань і займає 1 сторінку. В роботі наведено 49 рисунків. Наведено додаток на 15 сторінок.

У першого розділі було виконано загальний опис підприємства.

У другому розділі було описано виконану роботу під час проходження виробничої практики на підприємстві.

Висновок містить у собі результати виконаної роботи.

КЛЮЧОВІ СЛОВА: ВИРОБНИЧА ПРАКТИКА, SANA COMMERCE, HTML, CSS, SCSS, JavaScript, ASP.NET Core MVC, бази даних.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

**Актуальність теми** полягає у тому що, виробнича практика студентів це важливий вид навчальної роботи і здійснюється з метою набуття виробничих навиків приймати самостійно рішення в реальних виробничих умовах. Вона покликана підготувати майбутніх спеціалістів до реальної практичної роботи, забезпечити належний рівень їхньої професійної підготовки.

**Метою** даної практики є вивчення напрямків діяльності та організації роботи Sana Commerce, закріплення, поглиблення і розширення знань з теоретичних дисциплін веб-розробки та їх застосування до розв'язання актуальних проблем.

**Об'єкт дослідження практики** – діяльність компанії Sana Commerce.

**Предметом дослідження практики** є здобуття досвіду роботи в компанії Sana Commerce.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 1. ВІДОМОСТІ ПРО МІСЦЕ ПРОХОДЖЕННЯ ПРАКТИКИ

### *1.1 Sana Commerce.*

ТОВ «Сана Коммерс Україна» - міжнародна компанія, що займається як постійною розробкою власного e-Commerce продукту Sana Commerce Cloud, так і реалізацією численних проєктів на основі цього продукту. Компанія створює e-Commerce рішення, які у реальному часі інтегруються з ERP-системами замовників, такими як SAP або Microsoft Dynamics. Надійність і зручність таких інтегрованих рішень допомагають виробникам, дистриб'юторам та оптовикам досягати успіху, розвиваючи тривалі стосунки з їх клієнтами.



Рис. 1.1. Логотип компанії “Sana Commerce”.

Головний офіс Sana Commerce знаходиться у Роттердамі, у величній будівлі фабрики Van Nelle. Крім того, офіси компанії розташовані у Житомирі, Нью Йорку, Дубаї, Коломбо, Медельїні, Манчестері, Мюнхені.

Загальна кількість спеціалістів в Україні: 100+

Загальна кількість спеціалістів в світі: 500+

### *1.2 Основні види діяльності Sana Commerce.*

**1) Розробка інтернет-магазинів.** Щоб виділитися з-поміж своїх конкурентів, компанія забезпечує ідеальну роботу технічної сторони. Інтернет-

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

магазин повинен швидко завантажуватись і легко знаходитись пошуковими системами, наприклад, Google.

Під час створення інтернет-магазину компанія використовує власну платформу SANA.

Надається підтримка магазину у режимі 24/7. Команда висококваліфікованих експертів компанії з хостингу гарантує, що інтернет-магазин працюватиме цілодобово, навіть у випадку перевантаження трафіку.

Після запуску інтернет-магазину починається фаза оптимізації. Це не короткостроковий проект, а постійний процес вимірювання, звітності, аналізу та оптимізації. У секторі електронної комерції постійно відбуваються технологічні зміни, і застосовується до інтернет-магазину. Саме завдяки цьому замовник завжди знаходиться попереду конкурентів.

**2) Дизайн інтернет-магазинів.** Метою інтернет-магазину є обслуговування онлайн-покупців на вищому рівні, адже від цього прямо залежить успіх клієнта.

*Функціональний дизайн.* Функціональний дизайн – це схематичний проект інтернет-магазину. У цьому проекті розробники концентруються на його функціональних можливостях.

*Технічний дизайн.* Інтернет-магазин – це складна система, що включає зв'язки з платіжними системами, сторонніми агентами з логістики та системами складів. Крім того, також потрібно буде мати можливість ефективно переглядати інформацію щодо клієнтів та продуктів. Під час виконання технічного дизайну фахівці концентруються на всіх технічних аспектах інтернет-магазину, щоб забезпечити його безперебійну роботу 24 години на добу.

*Інтерактивний дизайн.* Успіх онлайн-бізнесу залежить від клієнтів. Якщо їх потреби не задовольняються, вони покидають інтернет-магазин, так нічого і не придбавши. Підхід компанії до розробки інтернет-магазину базується на роках сумлінного досвіду. Таким чином, гарантується, що очікування і потреби клієнтів завжди є головним пріоритетом.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

*Графічний дизайн.* Дизайнери у сфері електронної комерції забезпечать втілення в інтернет-магазині бренду замовника.

**3) Онлайн-маркетинг.** Компанія може допомогти клієнту отримати більше доходів від його інтернет-магазину. Команда з онлайн-маркетингу виконає оптимізацію інтернет-магазину, щоб забезпечити залучення якомога більшої кількості відвідувачів і перетворення їх на постійних клієнтів.

*Збільшення кількості відвідувачів.* Збільшення кількості відвідувачів інтернет-магазину – це спеціалізація Sana Commerce. Досвідчена команда у сфері електронної комерції є майстрами своєї справи. Вони знаються на інтернет-магазинах і можуть надати допомогу замовнику у наступних питаннях:

- медійна реклама;
- ретаргетинг (ремаркетинг);
- реклама у Marktplaats;
- реклама у соцмережах, наприклад, у Facebook чи Twitter;
- реклама у пошукових системах, наприклад, Google Adwords чи Bing Advertising;
- пошукова оптимізація сайту (SEO);
- вебсайти, що виконують порівняння цін, наприклад, beslist.nl чи kieskeurig.nl;
- партнерський маркетинг;
- перетворення відвідувачів на постійних клієнтів.

Багато відвідувачів самі по собі не приносять прибутку, тому їх потрібно перетворити на постійних клієнтів. Шляхом передбачення потреб відвідувачів можна заохочувати їх робити покупки. Для того, щоб спрогнозувати потреби, ми в першу чергу вивчаємо відвідувачів. Аналізуються дані з Google Analytics, проводиться юзабіліті і А/В тести.

*Підвищення лояльності клієнтів.* Дуже важливо залучати нових клієнтів, але ще важливіше зберігати існуючу базу постійних покупців, адже саме вони забезпечують стабільний дохід. Sana Commerce може допомогти змовникам у

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

розробці та реалізації ефективних кампаній, які підвищують лояльність існуючих клієнтів.

### ***1.3 Ступінь розвитку інформаційної системи підприємства.***

Компанія має розвинену структуру інформаційної системи. Працівники компанії є фахівцями з наступних технологій.

- C#
- .NET
- .NET Core
- ASP.NET Core
- GraphQL
- .NET
- MSSQL
- ES6
- HTML
- CSS
- SCSS
- Client libraries: React, Redux, RxJS etc.

### ***1.4 Програмне забезпечення, яке використовується на підприємстві.***

Windows 10

Microsoft Office 365

Microsoft Visual Studio 2019

Microsoft SQL Server

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



## РОЗДІЛ 2. ХАРАКТЕРИСТИКА ТА ПЕРЕЛІК ВИКОНАНИХ РОБІТ

### 2.1 Адаптивна верстка (HTML, CSS, SCSS).

Протягом першого тижня мені необхідно було реалізувати адаптивну верстку за макетом у Figma.

**Figma** — векторний онлайн-сервіс розробки інтерфейсів та прототипування з можливістю організації спільної роботи, що розробляється однойменною компанією. Працює у двох форматах: у браузері та як клієнтський додаток на десктопі користувача. Зберігає онлайн-версії файлів, з якими працював користувач.

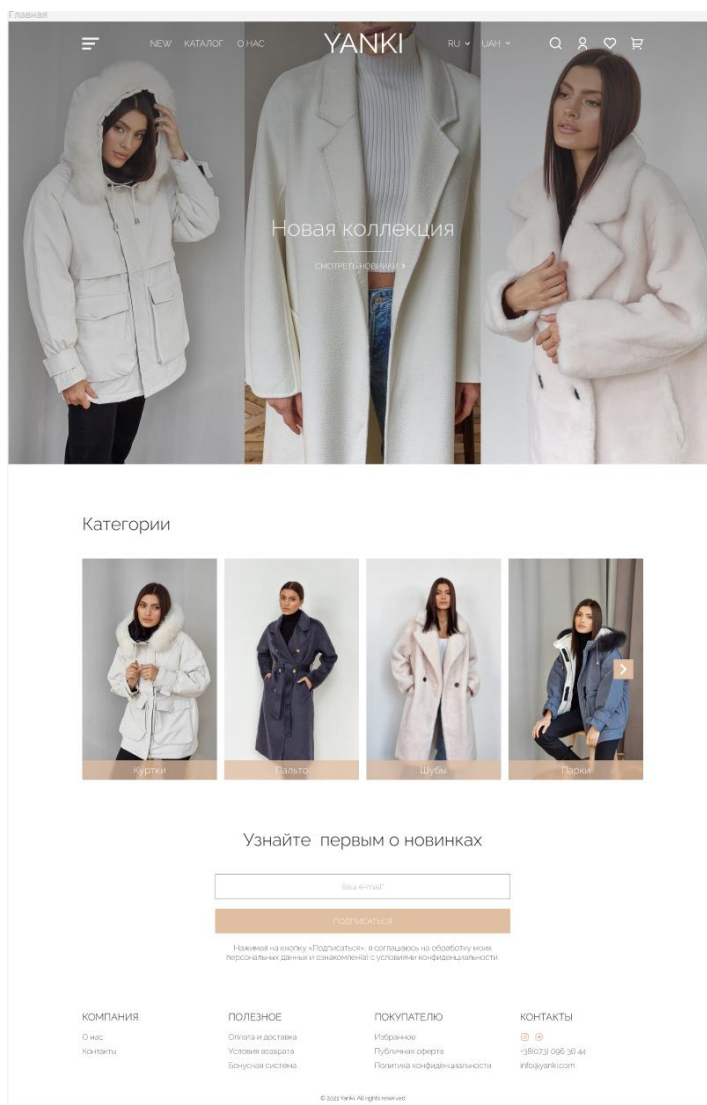


Рис. 2.1. Приклад макету головної сторінки з Figma.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

SCSS - це препроцесор для написання CSS коду. SCSS дає додаткові можливості. З ним можна використовувати змінні, вкладені селектори та медіазапити, збирати кілька файлів в один, міксини, успадкування та інші корисні речі, що спростять і структурують роботу з CSS кодом.

SCSS це окремий формат файлу, в якому пишуться стилі для сайту. Він дуже схожий на CSS, але на додачу має безліч поліпшень. Буквально SCSS розшифровується як Super CSS - тобто CSS с супер-силою.

SCSS пишеться в окремих файлах із розширенням .scss, наприклад main.scss.

SCSS не під'єднується в браузер безпосередньо. Браузери не вміють із ним працювати. SCSS компілюється (збирається) у звичний для браузера CSS код, і вже CSS файл підключається на сторінку.

Для того щоб скомпілювати SCSS у CSS, потрібно використовувати компілятор. Це може бути спеціальна програма: Koala, Prepros, або інша. Або плагін для редактора. Або інструмент для складання проєкту: Gulp, Webpack чи інший.

БЕМ (Блок, Елемент, Модифікатор) - компонентний підхід до веб-розробки. У його основі лежить принцип поділу інтерфейсу на незалежні блоки. Він дає змогу легко і швидко розробляти інтерфейси будь-якої складності та повторно використовувати наявний код, уникаючи "Copy-Paste".

Блок – це функціонально незалежний компонент сторінки, який може бути повторно використаний. У HTML блоки представлені атрибутом class.

Особливості:

- Назва блоку характеризує сенс ("що це?" - "меню": menu, "кнопка": button), а не стан ("який, який вигляд має?" - "червоний": red, "великий": big).
- Блок не повинен впливати на своє оточення, тобто блоку не слід задавати зовнішню геометрію (у вигляді відступів, кордонів, що впливають на розміри) і позиціонування.

У CSS за БЕМ також не рекомендується використовувати селектори за тегами або id.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<!-- Correct. The `error` block is semantically meaningful -->
<div class="error"></div>

<!-- Incorrect. It describes the appearance -->
<div class="red-text"></div>

```

Рис. 2.2. Приклад правильного та неправильного вигляду блоку

Елемент – це складова частина блоку, яка не може використовуватися у відриві від нього.

Особливості:

- Назва елемента характеризує зміст ("що це?" - "пункт": item, "текст": text), а не стан ("який, який вигляд має?" - "червоний": red, "великий": big).
- Структура повного імені елемента відповідає схемі: ім'я-блоку\_\_ім'я-елемента. Ім'я елемента відокремлюється від імені блоку двома підкресленнями(\_\_).

```

<!-- `search-form` block -->
<form class="search-form">
  <!-- `input` element in the `search-form` block -->
  <input class="search-form__input">

  <!-- `button` element in the `search-form` block -->
  <button class="search-form__button">Search</button>
</form>

```

Рис. 2.3. Приклад правильного вигляду елемента

Модифікатор – це сутність, що визначає зовнішній вигляд, стан або поведінку блоку чи елемента.

Особливості:

- Назва модифікатора характеризує зовнішній вигляд ("який розмір?", "яка тема?" тощо. - "розмір": size\_s, "тема": theme\_islands), стан ("чим відрізняється від інших?" - "відключений": disabled, "сфокусований":

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

focused) і поведінка ("як поводитьсья?", "як взаємодіє з користувачем?" - "напрямок": directions\_left-top).

- Ім'я модифікатора відокремлюється від імені блоку або елемента одним підкресленням (\_).

```
<!--
  Correct. The `search-form` block has the `theme` modifier with
  the value `islands`
-->
<form class="search-form search-form_theme_islands">
  <input class="search-form__input">

  <button class="search-form__button">Search</button>
</form>

<!-- Incorrect. The modified class `search-form` is missing -->
<form class="search-form_theme_islands">
  <input class="search-form__input">

  <button class="search-form__button">Search</button>
</form>
```

Рис. 2.4. Приклад правильного та неправильного вигляду модифікатора

Завдання звучало так:

1. Виконати верстку макета (4 сторінки: головна, каталог, картка товару, кошик) з використанням SCSS та методології BEM.
2. Створити репозиторій на GitHub з назвою "Practice" та розмістити у ньому в папці Week1 файли з виконаним завданням.
3. Відкрити доступ для акаунта morozov@ztu.edu.ua.

Я буду використовувати WebStorm від JetBrains як своє середовище розробки. Почну розробку з написання розмітки HTML, використовуючи методологію БЕМ. Нижче наведений приклад використання методології на основній частині сторінки каталогу:

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<div class="main">
  <div class="main__filter-title"...>
  <div class="main__filters">
    <div class="main__filter-item">
      <div class="main__filter-subtitle">
        Розмер <span class="icon-arrow-down-small"></span>
      </div>
      <div class="main__subfilters"...>
    </div>
    <div class="main__filter-item"...>
    <div class="main__filter-item"...>
    <div class="main__filter-item"...>
  </div>
  <main class="main__gallery">
    <div class="main__item"...>
    <div class="main__item"...>
    <div class="main__item"...>
    <div class="main__item"...>
    <div class="main__item"...>

```

Рис. 2.5. Написання розмітки

Як можна побачити з прикладу, основний контент сторінки буде «обгорнутий» батьківським блоком з класом “main”. Усі дочірні елементи мають назву за зразком “main\_\_назва-елементу”, селектори id не використовуються.

Написавши базову розмітку, переглянемо створений сайт.

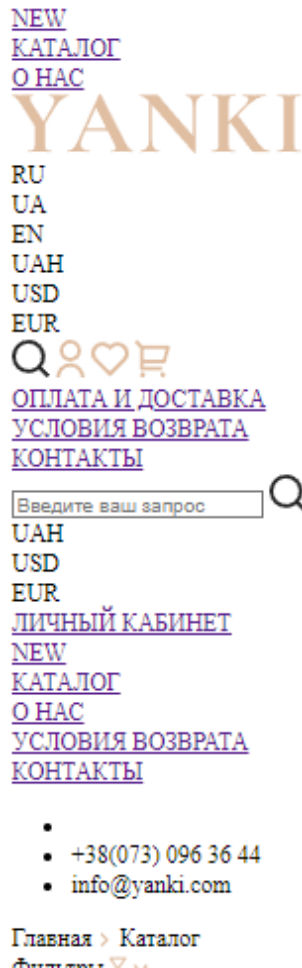


Рис. 2.6. Вигляд сторінки без .css-стилів

Переконавшись, що розмітка відповідає нашим потребам, перейдемо до створення .css-стилів. Відповідно до завдання, я використовуватиму препроцесор SCSS, що є більш зручним у використанні ніж звичайні .css-стилі. Весь код стилів пишеться в окремому файлі з розширенням .scss, а далі за допомогою налаштованого File Watcher, перетворюється у зрозумілий браузеру файл стилів .css.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

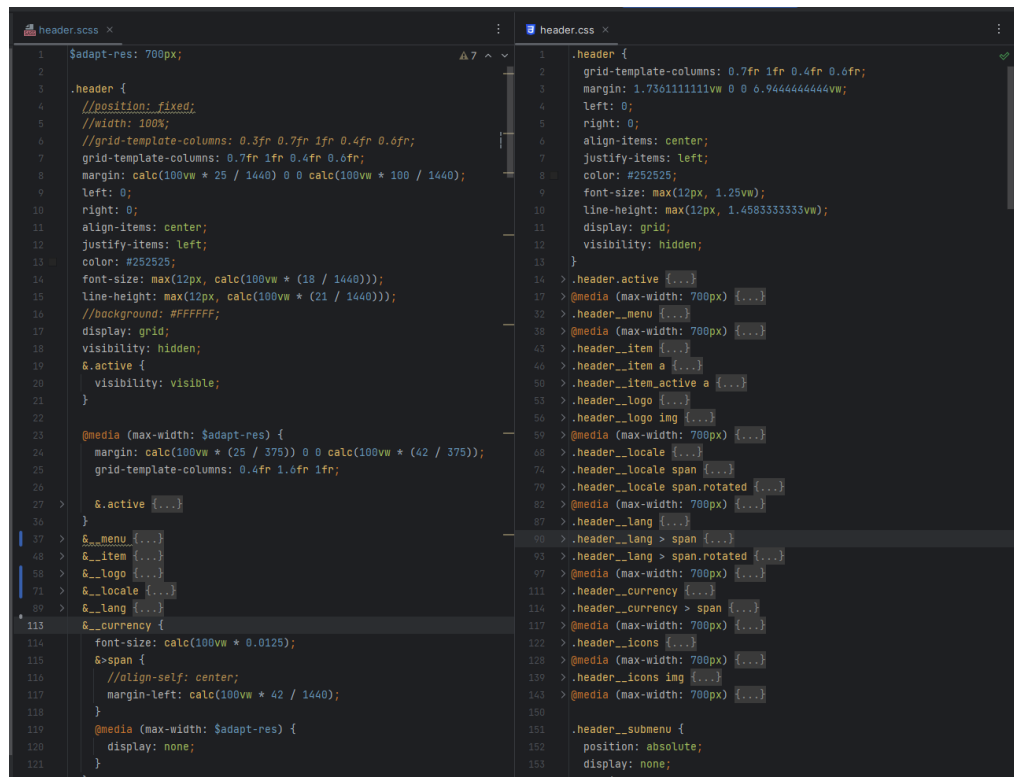


Рис. 2.7. Порівняння коду препроцесора та оброблений код у .css.

При написанні стилів, було реалізовано адаптивність сайту завдяки відсутнім статичним розмірам елементів, та реалізовано мобільне представлення.

Також для іконок сайту я згенерував шрифт з іконок за допомогою сервісу IcoMoon.

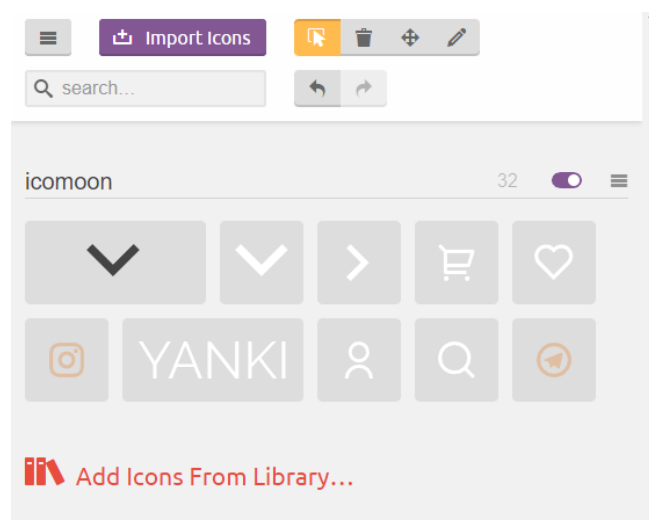


Рис. 2.8. Створення власного шрифту іконок.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Завершивши написання HTML-розмітки та CSS-стилів, переглянемо фінальні результати:

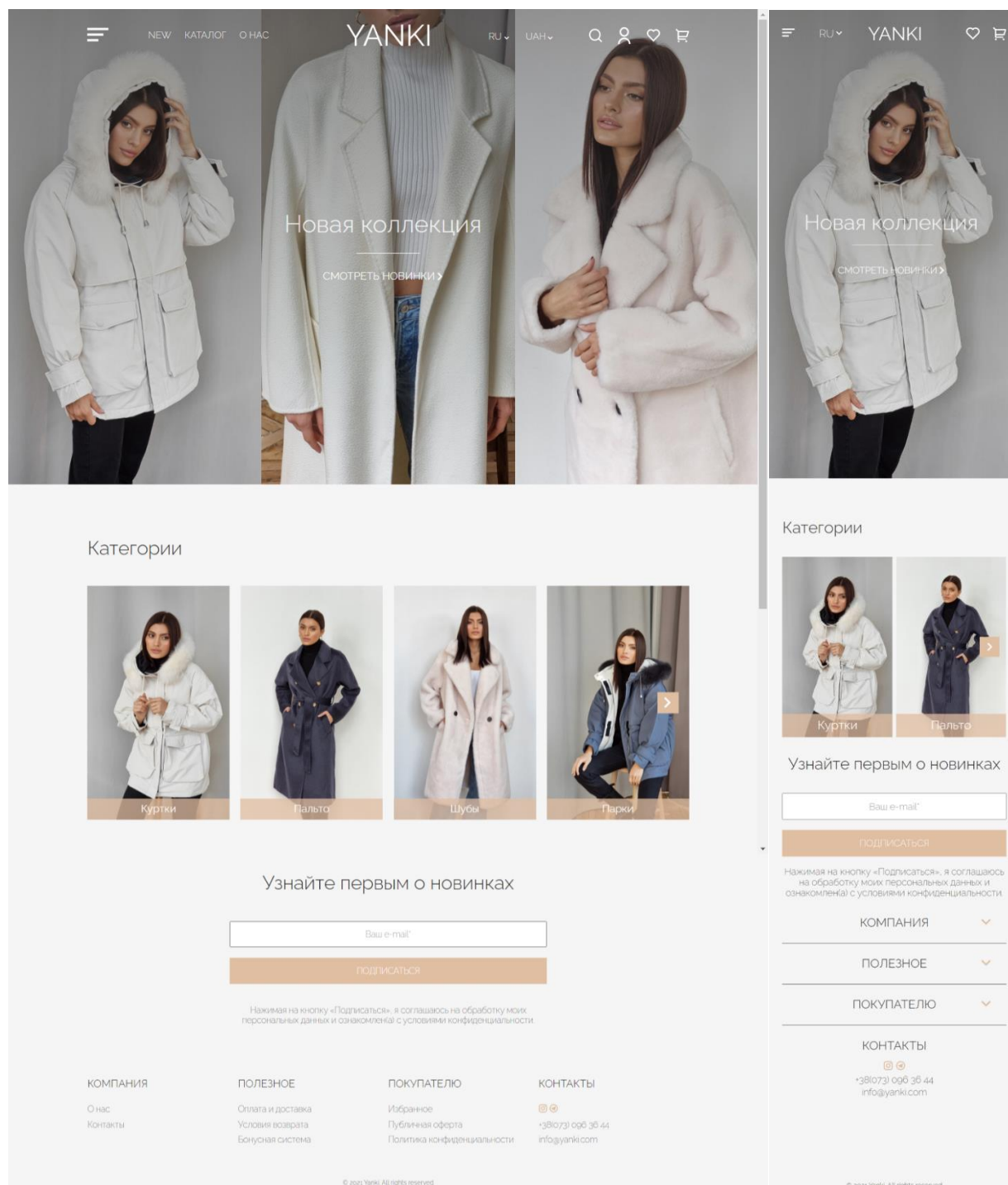


Рис. 2.9. Результат верстки головної сторінки та її мобільне представлення.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		



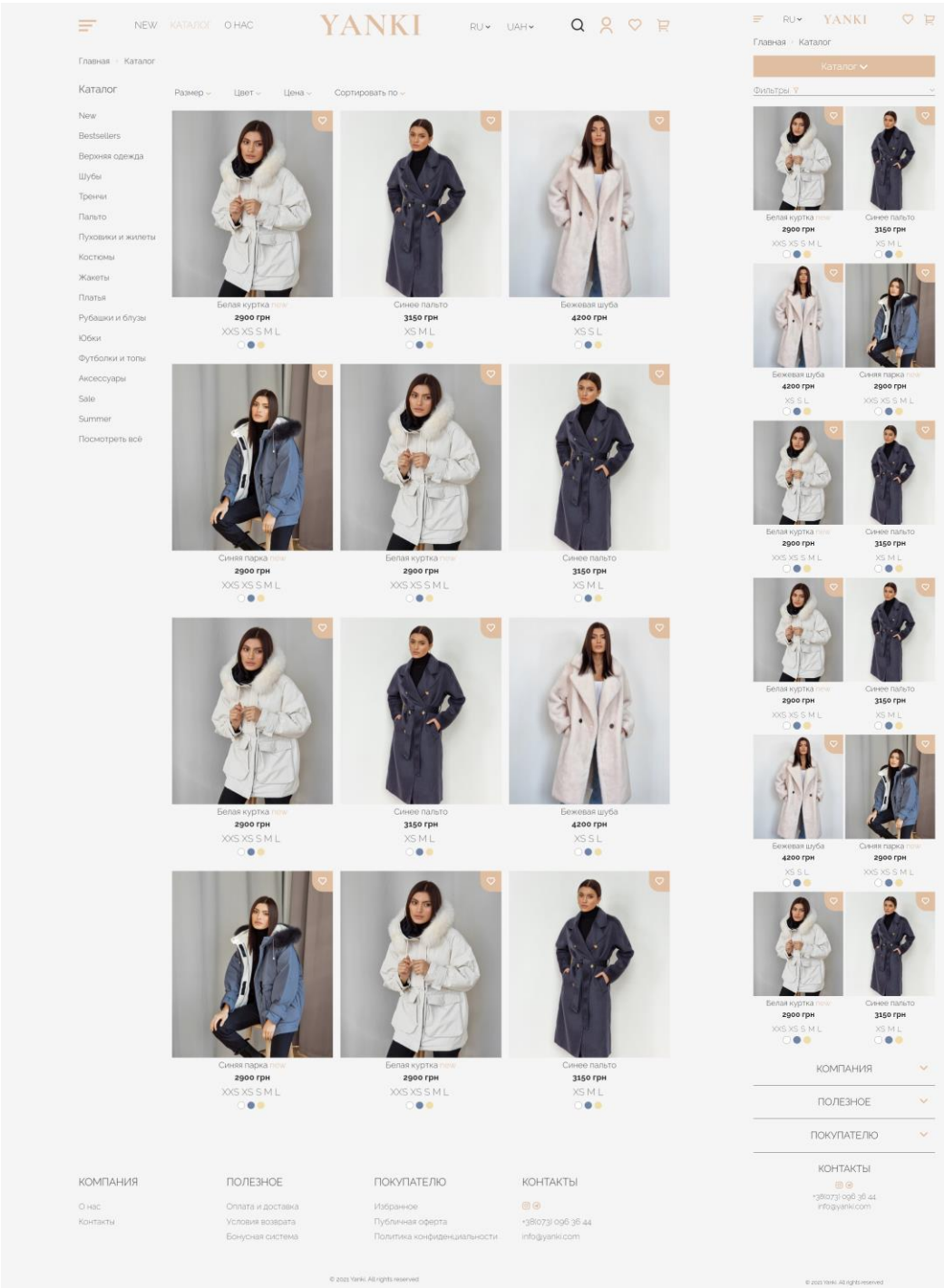


Рис. 2.10. Результат верстки сторінки каталогу та його мобільне представлення.

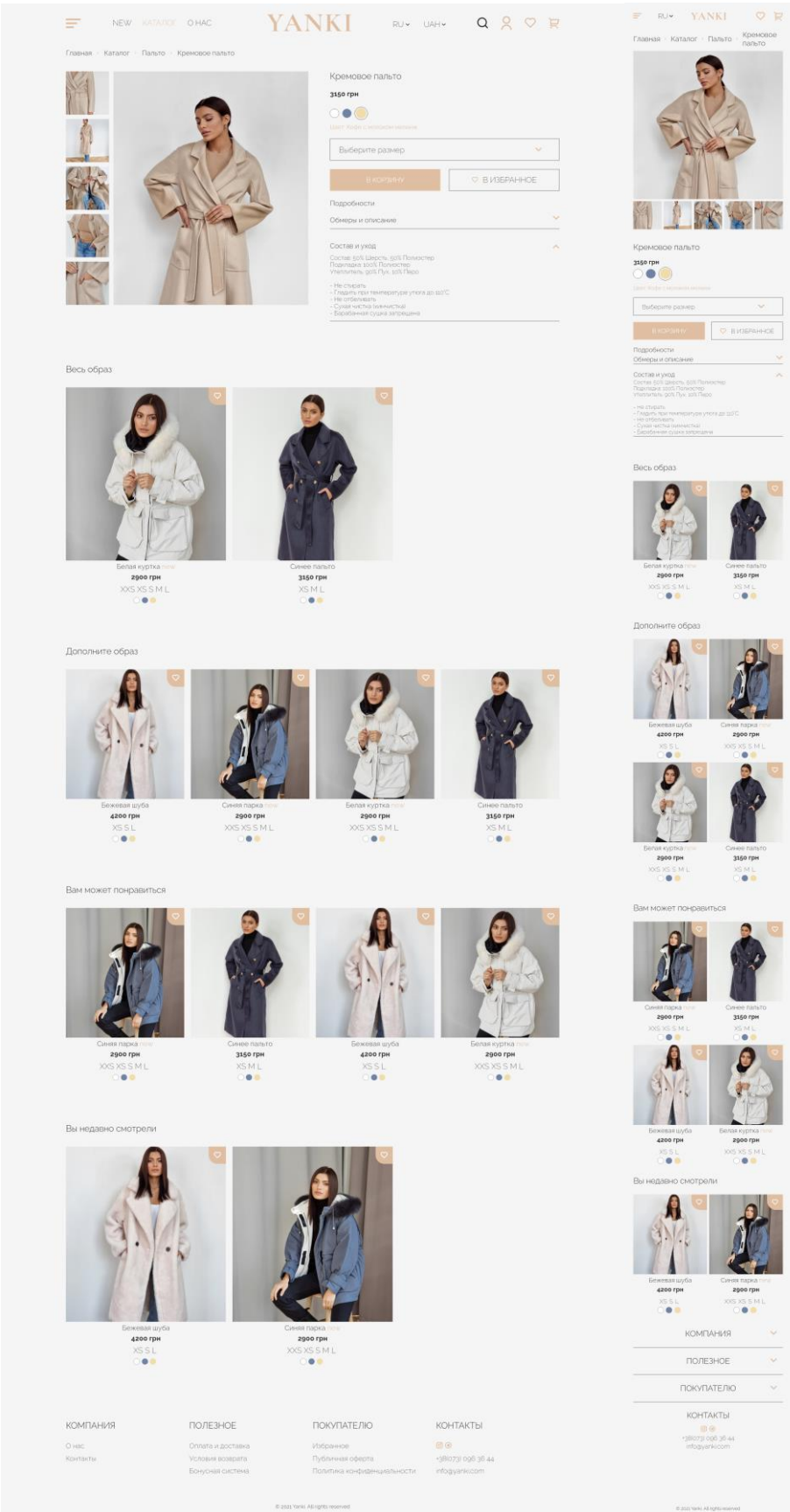


Рис. 2.11. Результат верстки сторінки продукту та її мобільне представлення.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

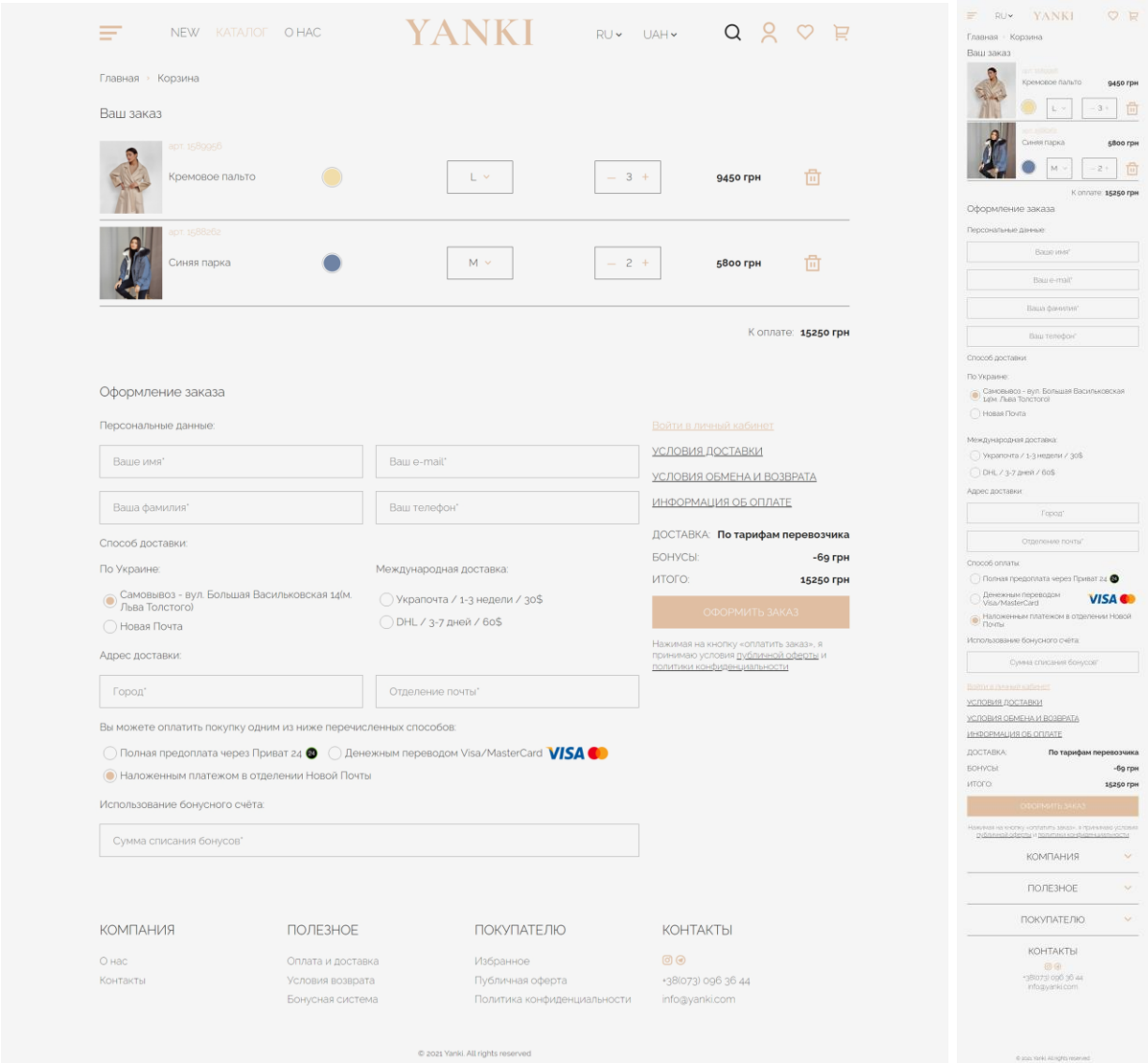


Рис. 2.12. Результат верстки сторінки кошику та її мобільне представлення.

Після реалізації та перевірки адаптивності верстки, перейдемо до завантаження наших файлів проекту на GitHub.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

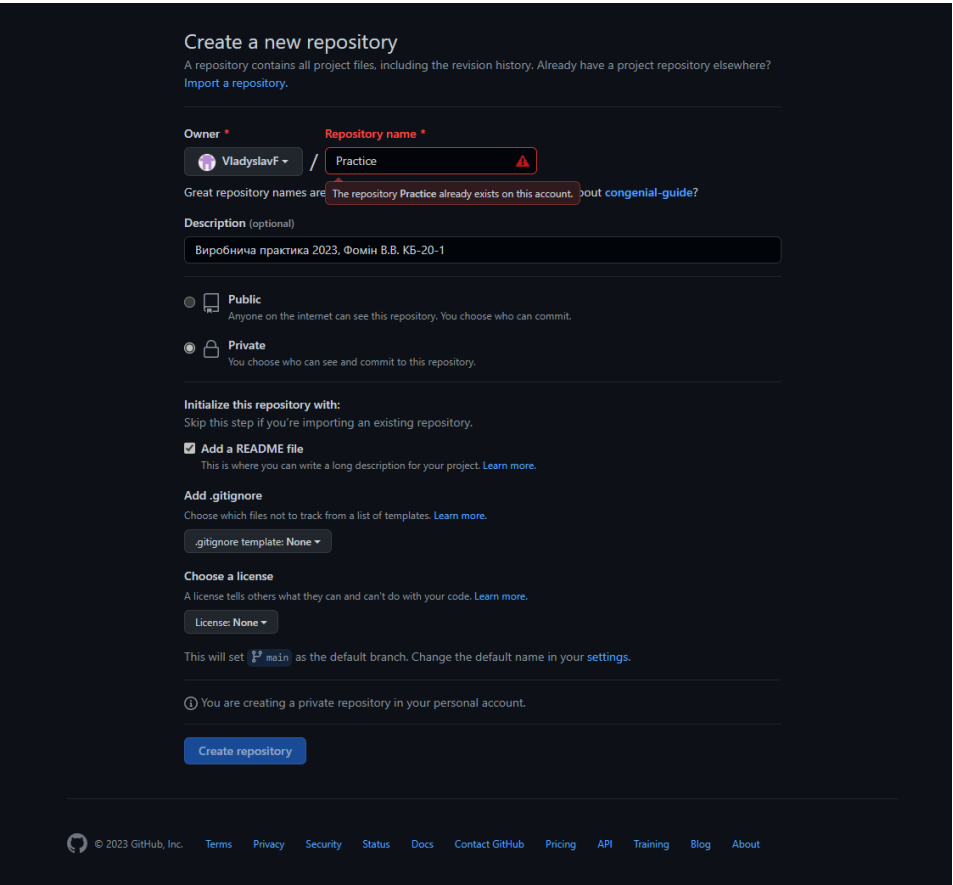


Рис. 2.13. Приклад створення власного репозиторію на GitHub

Для завантаження файлів можна скористатись самим сайтом, або програмним застосунком Git.

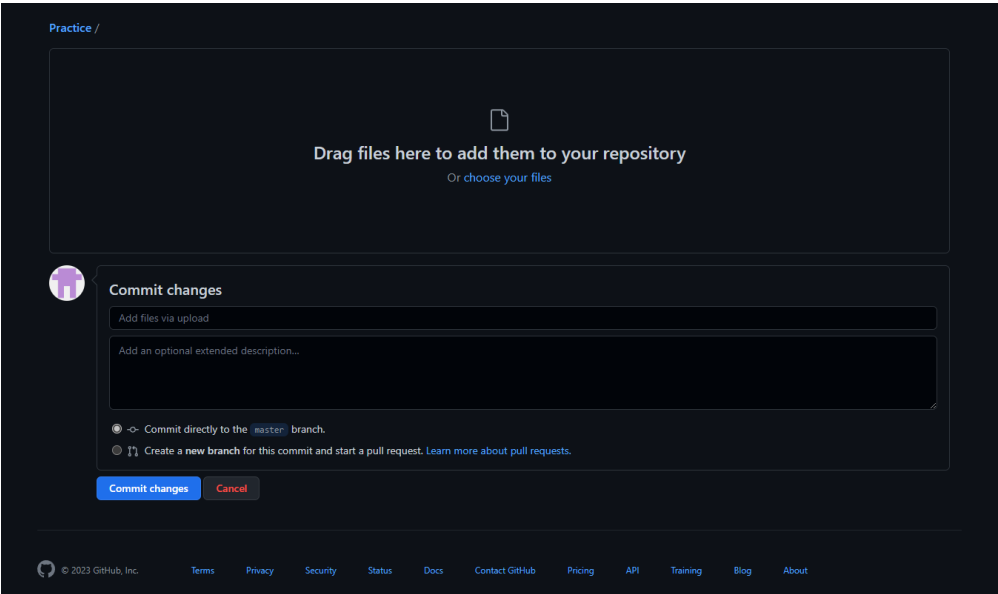


Рис. 2.14. Завантаження файлів у репозиторій

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

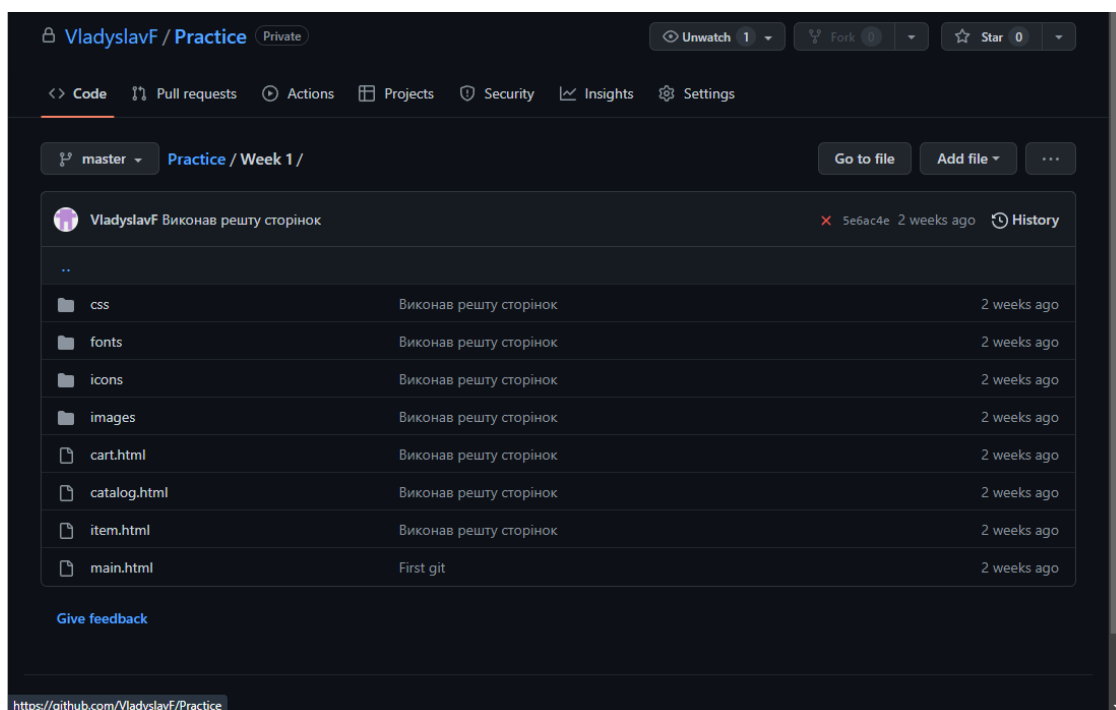


Рис. 2.15. Створений та заповнений репозиторій

## 2.2 Мова програмування JavaScript.

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Моїм завданням стало:

- Реалізація випадаючих списків меню;
- Показ/приховування мобільного меню;
- Реалізація випадаючих списків адаптивної верстки у нижньому меню.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

- Реалізація можливості вибору фотографії для перегляду на сторінці «Картка товару».

Першим я вирішив реалізувати базовий функціонал випадаючих списків.

Для цього за допомогою метода `.querySelector()` необхідно вибрати батьківський блок, де знаходитиметься наш випадаючий список і на нього встановити прослуховувач подій натискання на елемент (`.addEventListener('click')`).

Далі додамо до елемента ще один клас, який ми перемикатимемо при натисканні на елемент – це і буде відображати наш список.

```

VladyslavF
document.querySelector(selectors: '.aside>.main__title').addEventListener( type: 'click', listener: (event: Event) => {
  let elem = document.querySelector(selectors: '.aside');
  let list = elem.parentNode.querySelector(selectors: '.main__links');
  let check = elem.parentNode.querySelector(selectors: '.main__title>span');
  console.log(elem);
  list.classList.toggle( token: 'opened');
  check.classList.toggle( token: 'opened');
});

VladyslavF
document.querySelector(selectors: '.main__filter-title').addEventListener( type: 'click', listener: (event: Event) => {
  //let elem = event.target;
  let elem = document.querySelector(selectors: 'main');
  let title = document.querySelector(selectors: '.main__filter-title');
  let list = elem.parentNode.querySelector(selectors: '.main__filters');
  let check = elem.parentNode.querySelector(selectors: '.main__filter-title>img:last-child');
  console.log(elem);
  title.classList.toggle( token: 'opened');
  list.classList.toggle( token: 'opened');
  check.classList.toggle( token: 'rotated');
});

VladyslavF
document.querySelectorAll(selectors: '.main__filter-subtitle').forEach( callbackfn: function (item: Element) {
  item.addEventListener( type: 'click', listener: (event: Event) => {
    let elem = event.target;
    if (elem === document.querySelector(selectors: '.main__filter-subtitle>span')) {
      return 0;
    }
    let parent = elem.parentNode;

    parent.querySelector(selectors: '.main__filter-subtitle>span').classList.toggle( token: 'rotated');
    parent.classList.toggle( token: 'opened');
    parent.querySelector(selectors: '.main__subfilters').classList.toggle( token: 'opened');
  });
});

```

Рис. 2.16. Реалізація випадаючих списків на JavaScript

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

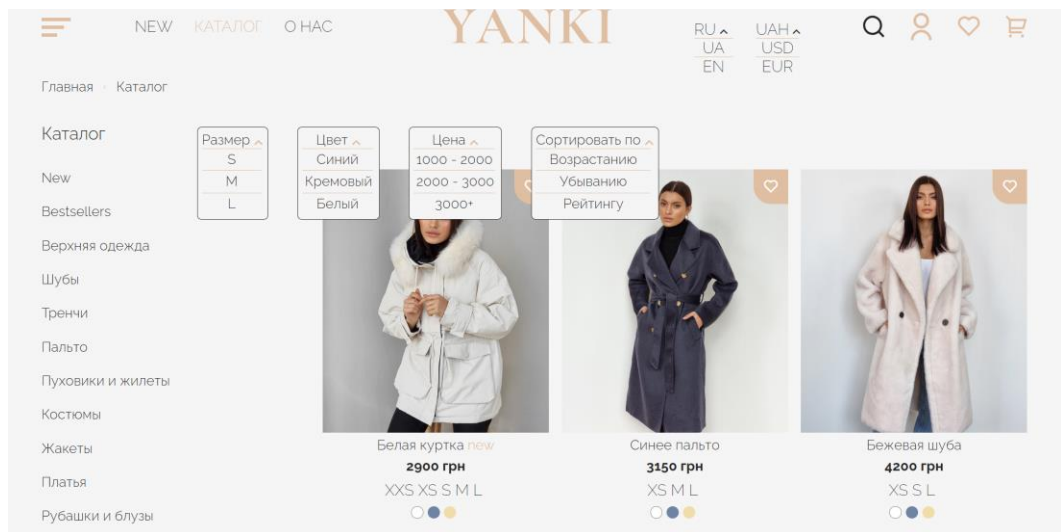


Рис. 2.17. Перевірка роботи випадаючих списків на сторінці каталогу

Також додатково реалізуємо випадаючі списки з можливістю вибору розміру одягу, можливість змінювати кількість придбаного товару та видалення товару з кошику.

Вибір елементу зі списку реалізуємо копіюванням тексту з вибраного елементу на місце першого.

```
document.querySelectorAll(selectors: '.order__sizes>.order_dropdown>.sizes_item').forEach(callbackfn: function (item :Element) {
  item.addEventListener( type: 'click', listener: (event :Event) => {
    let elem = event.target;
    let parent = elem.parentNode.parentNode;
    console.log(elem);
    let temp = parent.querySelector(selectors: '.order__size').innerText;
    console.log(temp);
    parent.querySelector(selectors: '.order__size').innerText = elem.innerText;
    elem.innerText = temp;
  });
});
```

Рис. 2.18. Реалізація вибору елементу зі списку





Рис. 2.19. Перевірка роботи випадаючих списків, зміни кількості товару та видалення елементів з корзини.

Перейдемо до реалізації меню «бургер» та мобільного меню. Кнопка «бургер» анімована за допомогою CSS, поява меню за допомогою переключання класів за допомогою JS. Також не варто забувати за перевірку розширення екрану, щоб вдало застосовувати мобільне меню.

```
document.querySelector(selectors: '.burger__lines').addEventListener( type: 'click', listener: (event :Event) =>{
    let elem = event.target;
    elem.parentNode.querySelector(selectors: '.burger__line1').classList.toggle( token: 'rotated');
    elem.parentNode.querySelector(selectors: '.burger__line2').classList.toggle( token: 'rotated');
    elem.parentNode.querySelector(selectors: '.burger__line3').classList.toggle( token: 'rotated');

    let main_menuElem = document.querySelector(selectors: '.header');
    let secondary_menuElem = document.querySelector(selectors: '.header__submenu');
    let mobile_main_menuElem = document.querySelector(selectors: '.mobile-menu');
    let vw = Math.max( values: document.documentElement.clientWidth || 0, window.innerWidth || 0);
    if (vw <= 700) {
        //main_menuElem.classList.toggle('active');
        secondary_menuElem.classList.toggle( token: 'active');
        mobile_main_menuElem.classList.toggle( token: 'active');
    }
    else if (vw > 700) {
        main_menuElem.classList.toggle( token: 'active');
        secondary_menuElem.classList.toggle( token: 'active');
    }
});
```

Рис. 2.20. Реалізація меню «бургер».

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				24
Змн.	Арк.	№ докум.	Підпис	Дата		





Рис. 2.21. Перевірка роботи меню «бургер» в звичайному режимі

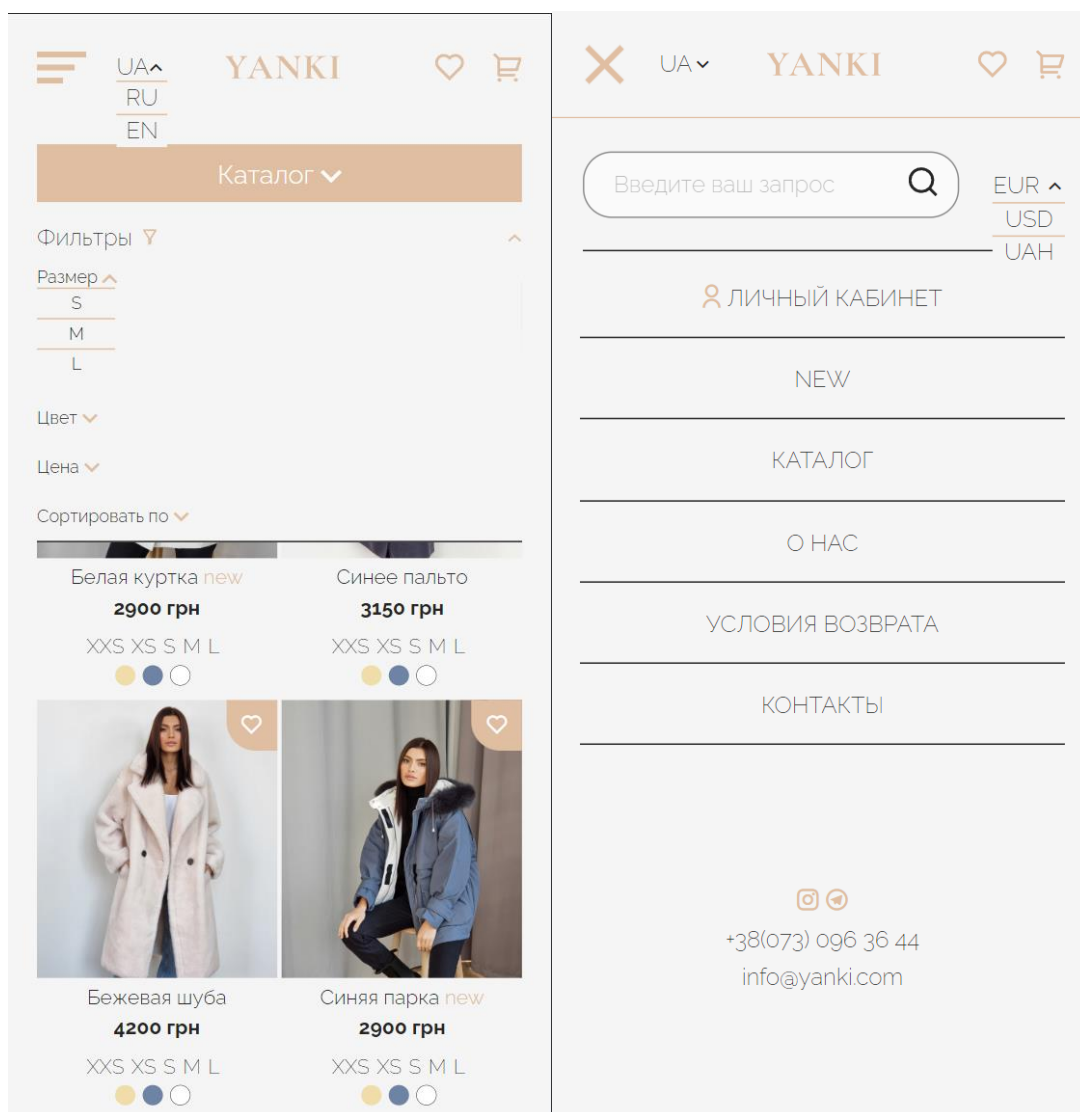


Рис. 2.22. Перевірка роботи мобільного меню

Тепер, коли ми переконалися що мобільне меню працює коректно, перейдемо до реалізації випадаючих списків адаптивної верстки у нижньому меню. Принцип виконання тут такий же як і у попередніх випадаючих списків, тому проблем не виникло.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

```
document.querySelector( selectors: '.footer__columns').addEventListener( type: 'click', listener: (event :Event ) => {
    let elem = event.target;
    let list = elem.parentNode.querySelector( selectors: '.footer__list');
    let check = elem.parentNode.querySelector( selectors: 'span')
    console.log(elem);
    list.classList.toggle( token: 'opened');
    check.classList.toggle( token: 'rotate');
});
```

Рис. 2.23. Реалізація адаптивного випадаючого списку нижнього меню.



Рис. 2.24. Вигляд нижнього меню за звичайного перегляду.

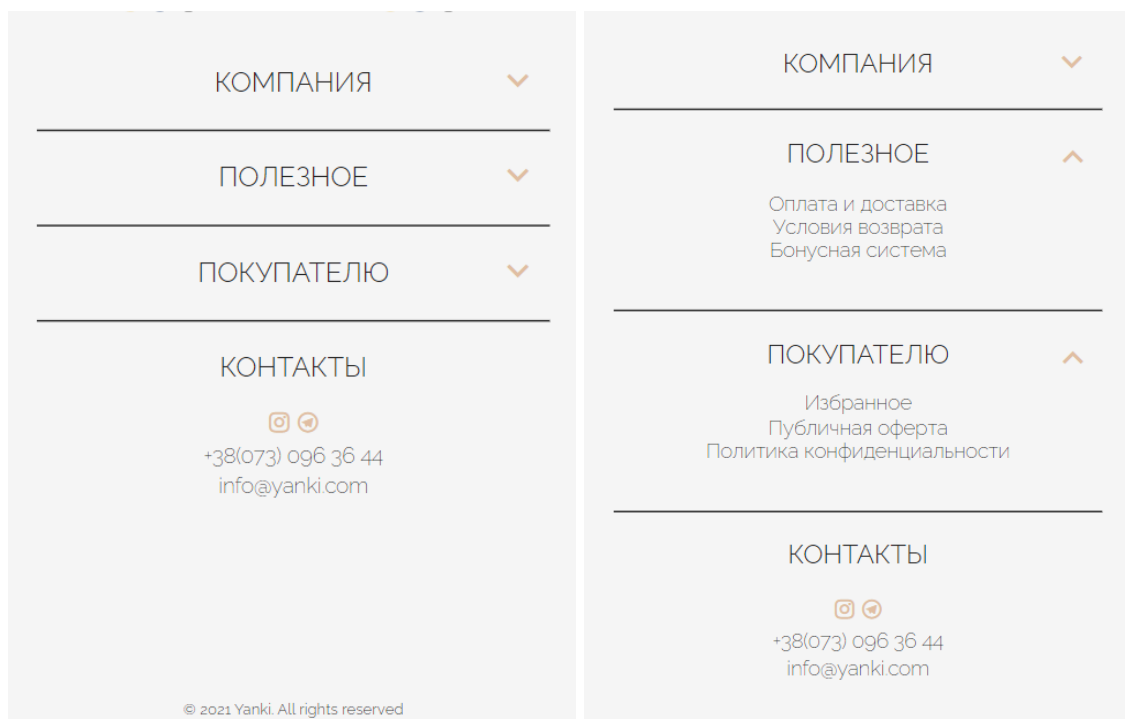


Рис. 2.25. Перевірка роботи нижнього меню за мобільного перегляду.

Перейдемо до реалізація вибору картинки для перегляду на картці товару. Відбуватиметься це шляхом заміти шляху зображення для великого зображення на шлях вибраного.

```
document.querySelector(selectors: '.main__small').addEventListener( type: 'click', listener: (event :Event ) => {
    let elem = event.target;
    let big = document.querySelector(selectors: '.main__big>.main__photo>img');
    console.log(elem);
    let temp = big.src;
    big.src = elem.src;
    elem.src = temp;
});
```

Рис. 2.26. Реалізація вибору зображення для перегляду.

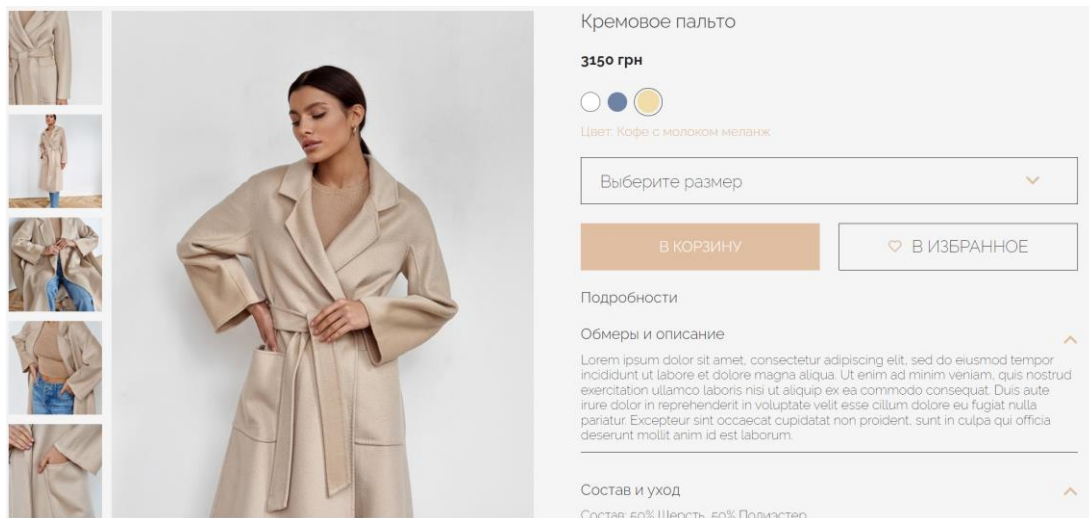


Рис. 2.27. Вигляд картки товару за замовчуванням.

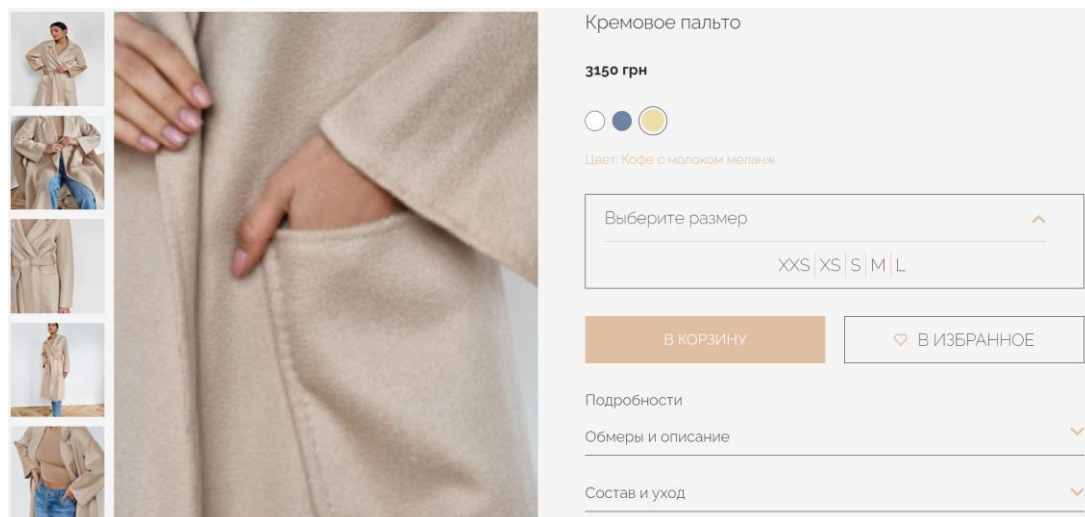


Рис. 2.28. При натискання на менше зображення, активне зображення змінюється на вибране.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

### 2.3 Технологія ASP.NET MVC Core

Фреймворк ASP.NET Core MVC є частиною платформи ASP.NET Core, його відмінна риса - застосування патерну MVC. Перевагою використання фреймворка ASP.NET Core MVC порівняно з "чистим" ASP.NET Core є те, що він спрощує в низці ситуацій і сценаріїв організацію та створення додатків, особливо це стосується великих додатків.

Варто зазначити, що паттерн MVC не є винятковою особливістю ASP.NET Core MVC, цей паттерн з'явився ще наприкінці 1970-х років у компанії Xerox як спосіб організації компонентів у графічному застосунку на мові Smalltalk і наразі застосовується на багатьох платформах і для різних мов програмування. Особливо популярний паттерн MVC у веб-додатках.

Концепція патерну MVC передбачає поділ програми на три компоненти:

**Модель (model):** описує використовувані в застосунку дані, а також логіку, яка пов'язана безпосередньо з даними, наприклад, логіку валідації даних. Як правило, об'єкти моделей зберігаються в базі даних.

У MVC моделі представлені двома основними типами: моделі уявлень, які використовуються уявленнями для відображення і передавання даних, і моделі домену, які описують логіку управління даними.

Модель може містити дані, зберігати логіку управління цими даними. Водночас модель не повинна містити логіку взаємодії з користувачем і не повинна визначати механізм обробки запиту. Крім того, модель не повинна містити логіку відображення даних у поданні.

**Подання (view):** відповідають за візуальну частину або призначений для користувача інтерфейс, нерідко html-сторінка, через який користувач взаємодіє з додатком. Також подання може містити логіку, пов'язану з відображенням даних. Водночас подання не повинно містити логіку опрацювання запиту користувача або управління даними.

**Контролер (controller):** представляє центральний компонент MVC, який забезпечує зв'язок між користувачем і додатком, поданням і сховищем даних. Він містить логіку обробки запиту користувача. Контролер отримує дані, що

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

вводяться користувачем, і обробляє їх. І залежно від результатів обробки відправляє користувачеві певний висновок, наприклад, у вигляді подання, наповненого даними моделей.

Я отримав такі завдання:

- 1) Перенести верстку на сервер
- 2) Моделі(товари, категорії, користувачі і тд...) + БД зі зв'язками
- 3) Реєстрація та аутентифікація користувачів + ролі доступу "користувач" та "адміністратор"
- 4) CRUD операції для адміністратора

Почнемо з перенесення верстки на сервер. Для початку створимо проект ASP.NET Core у Visual Studio.

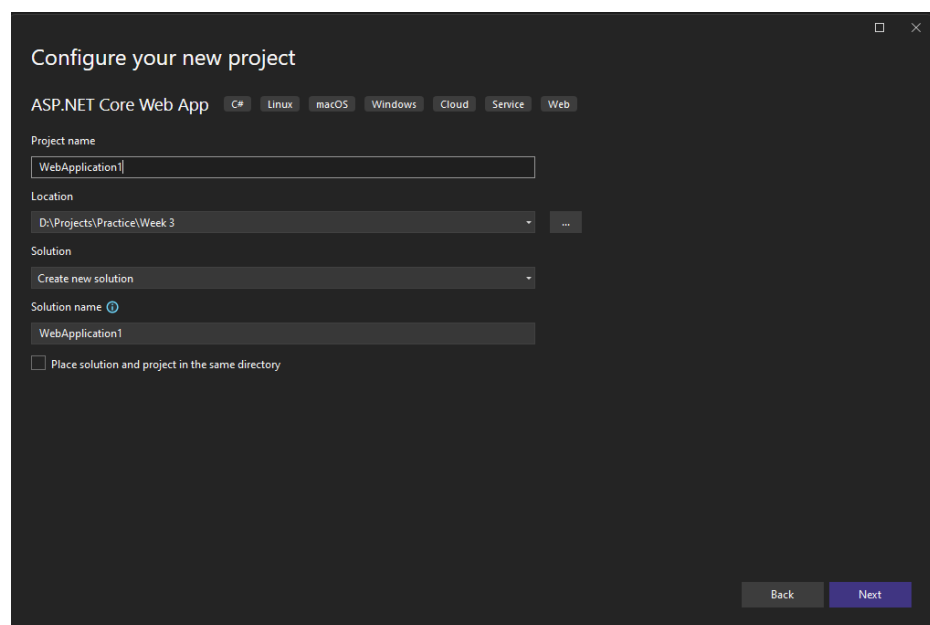


Рис. 2.29. Створення проекту.

Тепер перенесемо наші .js, .css файли та зображення у рут-каталог створеного проекту, за шляхом wwwroot.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

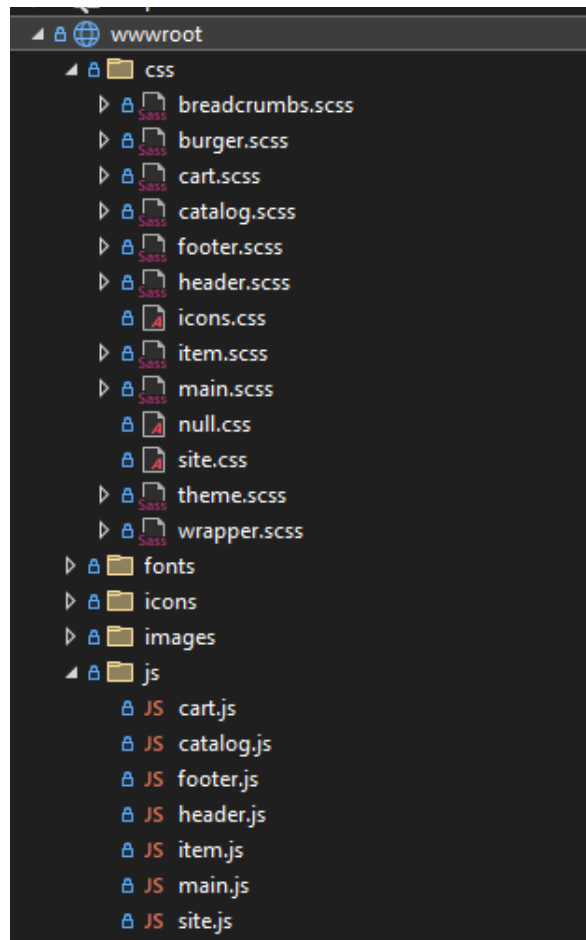


Рис. 2.30. Перенесення раніше створених файлів у проект.

Після того як ми перенесли необхідні нам файли у проект, почнемо перенесення верстки. Для початку розіб'ємо нашу сторінку на 3 частини: «шапка» - header, та «нижнє меню» - footer – елементи які повторюються між сторінками і винесемо їх у окремий \_Layout:

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

```

html    _Layout.cshtml  _ViewStart.cshtml  appsettings.json  Create.cshtml  Item.cs  Index.cshtml  _Layo
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - WebApplication1</title>
  <link rel="stylesheet" href="~/css/null.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/css/icons.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/css/theme.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/css/burger.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/css/header.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/css/breadcrumbs.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/css/wrapper.css" asp-append-version="true" />
  <link rel="stylesheet" href="~/css/footer.css" asp-append-version="true" />
</head>
<body>
  <div class="wrapper">
    <div class="burger">...
    <header class="header active">...
    <div class="header__submenu">...
    <div class="breadcrumbs">...
    @RenderBody()
    <footer class="footer">...
  </div>
  <script src="~/js/header.js" defer></script>
  <script src="~/js/footer.js" defer></script>
  @await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```

Рис. 2.31. Файл \_Layout.cshtml.

Головну частину сторінки яка буде відрізнятися виклатимемо методом @RenderBody(). Цю головну частину ми перенесемо у окремі представлення, відповідно до назви сторінки:

```

Index.cshtml  _Layout.cshtml  _ViewStart.cshtml  appsettings.json  Create.cshtml  Item.cs  Index.cshtml  _Layout_Main-page.cshtml  C
@*
  For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
  ViewData["Title"] = "YANKI - Catalog";
}
@using WebApplication1.Models.Shop
@model ShopFull
@{
  <div class="main">
    <div class="main__filter-title">...
    <div class="main__filters">...
    <main class="main__gallery">
      @foreach (var item in Model.products)
      {
        <div class="main__item">...
      }
    </main>
    <aside class="main__sidebar">
      <div class="main__title">...
      <div class="main__links">
        <a href="#">New</a>
        <a href="#">Bestsellers</a>
        @foreach (var category in Model.categories)
        {
          <a href="#">@category.Name</a>
        }
        <a href="#">Sale</a>
        <a href="#">Summer</a>
        <a href="#">Посмотреть все</a>
      </div>
    </aside>
  </div>
  <script src="~/js/catalog.js" defer></script>
  <link rel="stylesheet" href="~/css/catalog.css"/>
}

```

Рис. 2.32. Головна частина сторінки каталогу в окремому представлені.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

Залишилося написати контроллер для наших сторінок, який повертатиме створенні представлення. Наведено приклад контроллера для головної сторінки каталогу:

```
using Microsoft.AspNetCore.Mvc;
using WebApplication1.Data;
using WebApplication1.Models.Shop;

namespace WebApplication1.Controllers
{
    1 reference
    public class ShopController : Controller
    {
        0 references
        protected ShopDbContext shopDbContext;
        public ShopController(ShopDbContext shopDbContext)
        {
            this.shopDbContext = shopDbContext;
        }

        0 references
        public IActionResult Index()
        {
            0 references
            public IActionResult Item()
            {
                return View();
            }
            0 references
            public IActionResult Cart()
            {
                return View();
            }
        }
    }
}
```

Рис. 2.33. Створений контролер для сторінок каталогу, кошику та картки товару.

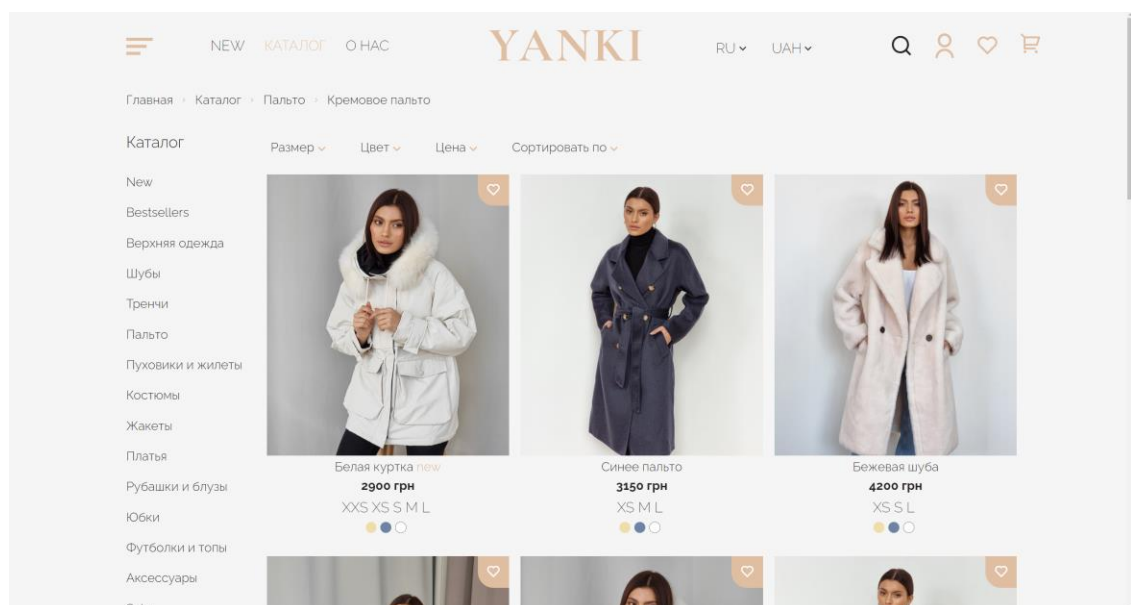


Рис. 2.34. Перевірка коректної роботи представлення

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				32
Змн.	Арк.	№ докум.	Підпис	Дата		



Тепер, коли ми переконалися що наша перенесена верстка працює, перейдемо до реалізації моделей, таких як категорії, товари, кольори, розміри і т.д. та створення відповідної бази даних для того щоб «оживити» наш сайт.

У моделі ми прописуємо поля таблиці бази даних, а для того щоб реалізувати зв'язки спочатку прописуємо поле яке виступатиме зовнішнім ключем таблиці та поле яке виступатиме об'єктом зв'язаної таблиці. Як приклад наведено модель продукту та її зв'язки:

```
namespace WebApplication1.Models.Shop
{
    22 references
    public class Product
    {
        15 references
        public int Id { get; set; }
        13 references
        public string Name { get; set; }
        13 references
        public double Price { get; set; }
        8 references
        public int CategoryId { get; set; }
        9 references
        public Category Category { get; set; }

        13 references
        public string ImageUrl { get; set; }

        0 references
        public ICollection<Product_Sizes> Sizes { get; set; }

        // public int SizesId { get; set; } ...

        8 references
        public int ColorId { get; set; }
        9 references
        public Color Color { get; set; }
        12 references
        public string? Description { get; set; }
        12 references
        public string? Ingredients { get; set; }

        13 references
        public string? Status { get; set; }
    }
}
```

Рис. 2.35. Створена модель продукту.

```
namespace WebApplication1.Models.Shop
{
    public class Product_Sizes
    {
        8 references
        public int Id { get; set; }
        8 references
        public int ProductId { get; set; }
        6 references
        public Product Product { get; set; }
        8 references
        public int SizeId { get; set; }
        6 references
        public Size Size { get; set; }
    }
}

namespace WebApplication1.Models.Shop
{
    20 references
    public class Size
    {
        12 references
        public int Id { get; set; }
        13 references
        public string Name { get; set; }
        12 references
        public int Number_size { get; set; }
        0 references
        public ICollection<Product_Sizes> Product_Sizes { get; set; }
    }
}
```

Рис. 2.36. Зв'язані моделі.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

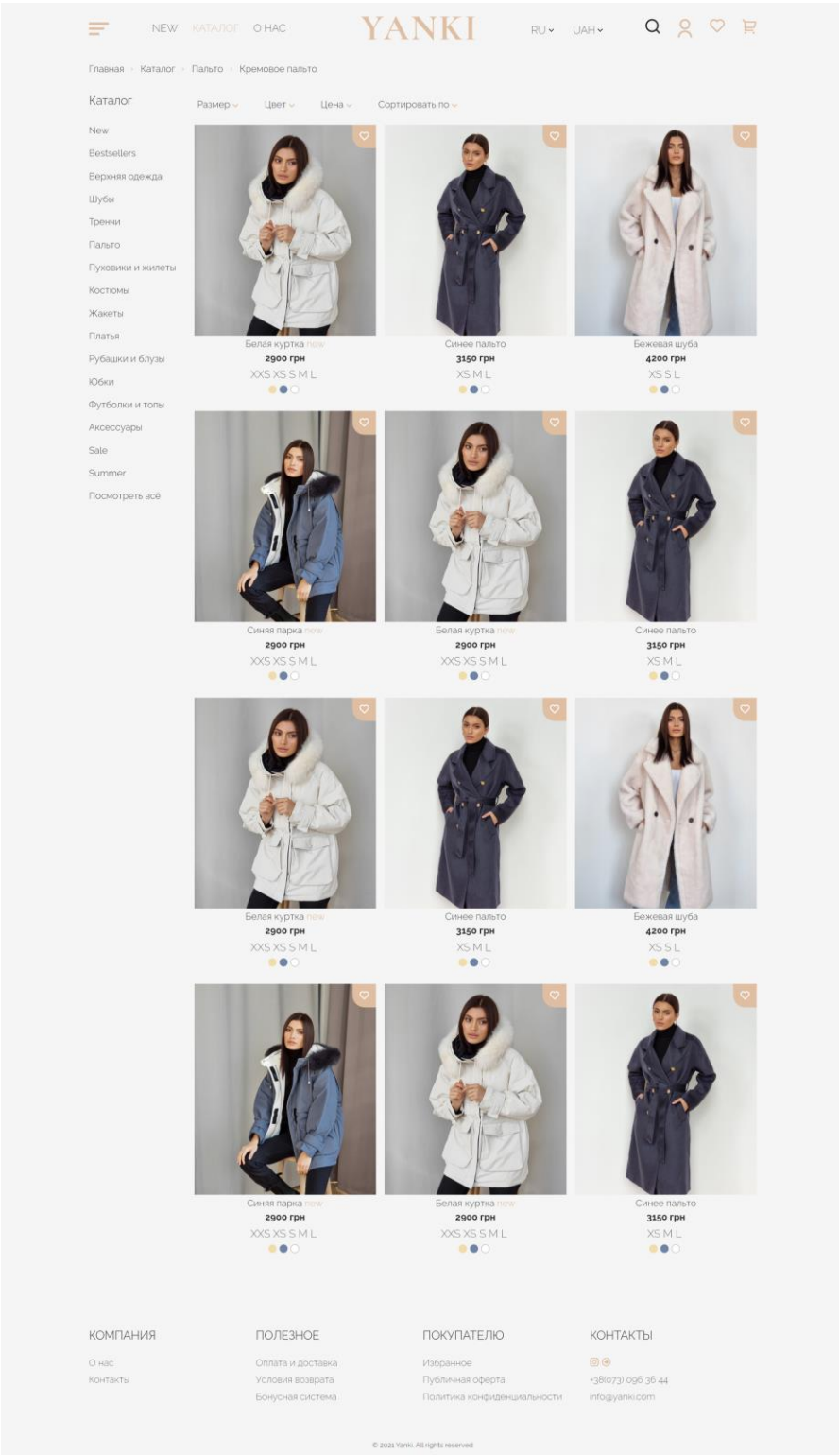


Рис. 2.37. Функціонування створених моделей.

Я створив моделі для списку каталогів, товарів, розмірів та елементів меню. Також на майбутнє створено моделі для авторизації користувачів:

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

## Create

### Product

Name

Кремове пальто

Price

3500

CategoryId

Верхня одежа

ImageUrl

img-c-1.png

ColorId

Beige

Description

Кремове пальто

Ingredients

Хлопок

Status

new

Create

[Back to List](#)

КОМПАНИЯ

О нас

Контакты

ПОЛЕЗНОЕ

Оплата и доставка

Условия возврата

Бонусная система

ПОКУПАТЕЛЮ

Избранное

Публичная оферта

Политика конфиденциальности

КОНТАКТЫ

+38(073) 096 36 44

info@yanki.com

© 2021 Yanki. All rights reserved

Рис. 2.37. Приклад створенного представлення для моделі продукту

```
namespace WebApplication1.Models.Shop
{
    14 references
    public class User
    {
        2 references
        public int Id { get; set; }
        3 references
        public string Name { get; set; }
        7 references
        public string Email { get; set; }
        6 references
        public string Password { get; set; }
        3 references
        public bool KeepLoggedIn { get; set; }
    }
}
```

Рис. 2.38. Модель для користувача.

Фінальна діаграма створених моделей має такий вигляд:

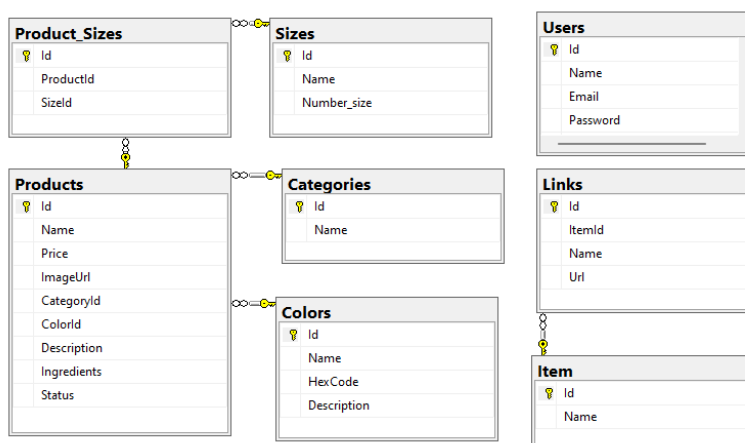


Рис. 2.39. Діаграма таблиць бази даних

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Реалізувавши необхідні моделі, перейдемо до реалізації можливості реєстрації та автентифікації для нашого сайту. Я реалізував ручну авторизацію. Так як модель уже створена, необхідно у основний файл проекту Program.cs додати нову службу, що відповідатиме аутентифікації:

```

1  using Microsoft.AspNetCore.Authentication.Cookies;
13 builder.Services.AddAuthentication(
14     CookieAuthenticationDefaults.AuthenticationScheme)
15     .AddCookie(option =>
16     {
17         option.LoginPath = "/Access/Login";
18         option.ExpireTimeSpan = TimeSpan.FromMinutes(20);
19     });
35 app.UseAuthentication();
36
37 app.UseAuthorization();
38
39 app.MapControllerRoute(
40     name: "default",
41     //pattern: "{controller=Home}/{action=Index}/{id?}");
42     pattern: "{controller=Access}/{action=Login}/{id?}");

```

Рис. 2.39. Додавання аутентифікації до проекту.

Тепер перейдемо до написання контролера для обробки авторизації. Першим підключимо необхідні бібліотеки і напишемо функцію логіну, якщо користувач уже авторизований – його перенаправить на сторінку профілю, де він зможе вийти з акаунта.

```

using Microsoft.AspNetCore.Mvc;
using System.Security.Claims;
using Microsoft.AspNetCore.Authentication;
using Microsoft.AspNetCore.Authentication.Cookies;
using WebApplication1.Models;
using WebApplication1.Models.Shop;
using WebApplication1.Data;

namespace WebApplication1.Controllers
{
    1 reference
    public class AccessController : Controller
    {
        protected ShopDbContext shopDbContext;
        0 references
        public AccessController(ShopDbContext shopDbContext)
        {
            this.shopDbContext = shopDbContext;
        }
        0 references
        public IActionResult Login()
        {
            ClaimsPrincipal claimUser = HttpContext.User;
            if (claimUser.Identity.IsAuthenticated) {
                //return RedirectToAction("Index", "Home");
                return RedirectToAction("Profile", "Access");
            }

            return View();
        }
    }
}

```

Рис. 2.40. Контролер авторизації та базова функція перевірки авторизації.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

Далі для перевірки чи існує користувач у нашій базі даних напишемо функцію обробник отриманих нами даних методом POST, за результатом якого, якщо користувач існує – його перенаправить на головну сторінку, якщо ні – виведе помилку.

```
[HttpPost]
public async Task<IActionResult> Login(User modelLogin)
{
    var db = new ShopFull();
    db.Users = shopDbContext.Users.ToList();
    foreach (var user in db.Users)
    {
        if (modelLogin.Email == user.Email)
        {
            if (modelLogin.Password == user.Password)
            {
                List<Claim> claims = new List<Claim>() {
                    new Claim(ClaimTypes.NameIdentifier, modelLogin.Email),
                    new Claim("OtherProperties", "Other Role")
                };

                ClaimsIdentity claimsIdentity= new ClaimsIdentity(claims,
                    CookieAuthenticationDefaults.AuthenticationScheme);
                AuthenticationProperties properties = new AuthenticationProperties()
                {
                    AllowRefresh = true,
                    IsPersistent = modelLogin.KeepLoggedIn
                };
                await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme,
                    new ClaimsPrincipal(claimsIdentity), properties);

                return RedirectToAction("Index", "Home");
            }
        }
    }
    ViewData["ValidateMessage"] = "User not found";
    return View();
}
```

Рис. 2.41. Реалізація перевірки чи існує користувач у базі даних.

Для реєстрації буде проведена перевірка чи користувач вже авторизований, якщо так – перенаправить на головну сторінку, якщо ні – на сторінку реєстрації.

```
0 references
public IActionResult Register()
{
    ClaimsPrincipal claimUser = HttpContext.User;
    if (claimUser.Identity.IsAuthenticated)
    {
        return RedirectToAction("Index", "Home");
    }

    return View();
}

0 references
public IActionResult Profile()
{
    return View();
}
```

Рис. 2.42. Функції реєстрації на відображення профілю.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

Реєстрація здійснюватиметься за допомогою автоматично створеного представлення від контролера користувачів.

```
0 references
public IActionResult Create()
{
    return View();
}

// POST: Users/Create
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create([Bind("Id,Name,Email,Password,KeepLoggedIn")] User user)
{
    if (ModelState.IsValid)
    {
        _context.Add(user);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(user);
}
```

Рис. 2.43. Функція реєстрації користувача.

Реалізувавши необхідні контролери та функції, перейдемо до створення представлень для логіну, реєстрації та профілю. Для прикладу наведено представлення сторінки логіну:

```
@{
    ViewData["Title"] = "Login";
    Layout = null;
}
@using WebApplication1.Models.Shop;
@model User;
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <link rel="stylesheet" href="/css/login.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">
</head>
<body>
    <div>
        <form asp-controller="Access" asp-action="Login" method="post">
            <div class="title">Авторизация</div>
            <div class="fields">
                <input type="email" placeholder="Ваш e-mail*" asp-for="Email" />
                <input type="password" placeholder="Ваш пароль*" asp-for="Password" />
            </div>
            <div>
                <input type="checkbox" asp-for="KeepLoggedIn" />
                Keep me logged in
            </div>
            <a asp-controller="Access" asp-action="Register">Регистрация</a>
            <br>
            <button type="submit">ВОЙТИ</button>
            @if (ViewData["ValidateMessage"] != null)
            {
                <br />
                <label>@ViewData["ValidateMessage"]</label>
            }
        </form>
    </div>
    <div>
        <a asp-controller="Home" asp-action="Index">Назад на главную</a>
    </div>
</body>
</html>
```

Рис. 2.44. Представлення сторінки логіну.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 2.45. Вигляд сторінки логіну.

Рис. 2.46. Вигляд сторінки реєстрації.

Для сторінки профілю реалізовано простий вигляд для можливості зміни авторизованого користувача.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

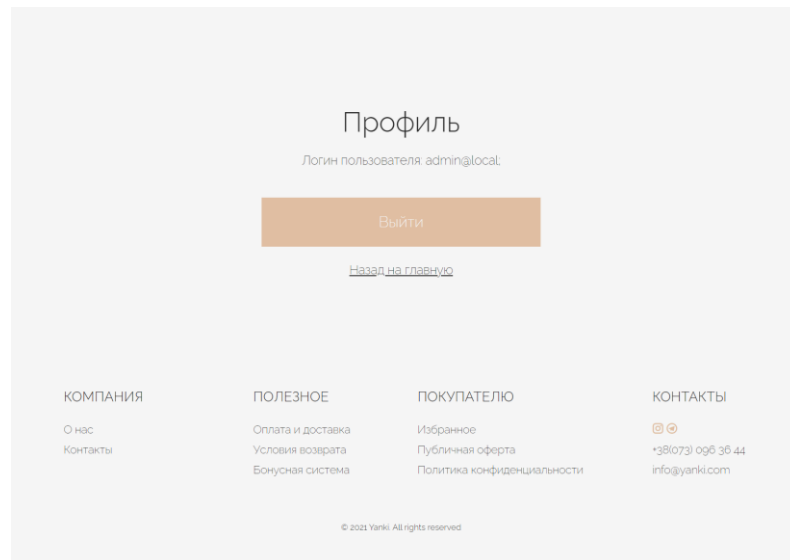


Рис. 2.47. Вигляд сторінки профілю.

Тепер після того як реєстрацію та авторизацію користувачів успішно реалізовано, необхідно дозволити доступ до CRUD-операцій лише адміністратору. Для цього на усіх представленнях доступ до створення, редагування, оновлення та видалення для усіх інших користувачів додамо перевірку авторизованого акаунта, щоб звичайного користувача при спробі перейти на сторінку операцій моделі автоматично пересилало на головну сторінку.

```
public IActionResult Create()
{
    var userID = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
    if (userID != "admin@local" )
    {
        return RedirectToAction("Index", "Home");
    }
    ViewData["CategoryId"] = new SelectList(_context.Categories, "Id", "Id");
    ViewData["ColorId"] = new SelectList(_context.Colors, "Id", "Id");
    return View();
}
```

Рис. 2.48. Реалізація перевірки авторизованого користувача.

Тепер звичайний користувач не зможе виконувати CRUD-операції.



## ВИСНОВКИ

У першому розділі було виконано опис підприємства, визначено його основні діяльності, переваги над конкурентами, основні здобутки за роки. Також було описано технології, які використовуються підприємством.

В результаті проходження виробничої практики було отримано достатню кількість практичного досвіду у верстці та налагодження бек-енду інтернет-магазину.

В основному розділі було виконано завдання з технологічної практики, розглянуто:

- Верстку сайту за макетом у Figma з використанням технологій HTML, CSS, SCSS.
- Реалізація адаптивності створених сторінок сайту.
- Реалізація випадаючих списків та мобільних меню за допомогою JavaScript.
- Реалізація бекенду інтернет-магазину за допомогою ASP.NET Core MVC та використання баз даних.

Я вважаю, що з поставленими задачами на практиці я впорався успішно і можливо, продовжу покращувати функціонал створеного сайту, для кращого засвоєння матеріалу. Завдяки даній виробничій практиці я отримав нові знання та досвід з веб- програмування, що дозволяє мені і далі розвиватись у даній сфері та покращувати свої навички веб-розробки.

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технічна документація SQL Server. Microsoft. URL:  
<https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>.
2. Фрілансер по життю. Адаптивна верстка сайтів за 37 минут + макет. Тебе не нужен Bootstrap! [ENG SUBS], 2019. YouTube. URL:  
<https://www.youtube.com/watch?v=XbnAKjJlGc4>.
3. Фрілансер по життю. Адаптивные изображения. Практические приемы верстки картинок. CSS object-fit HTML picture., 2020. YouTube. URL:  
<https://www.youtube.com/watch?v=nTtuiBHKp88>.
4. Фрілансер по життю. Адаптивный шрифт. Пример реализации отзывчивого (резинового) шрифта на CSS + SCSS миксин, 2020. YouTube. URL: <https://www.youtube.com/watch?v=HJZP5QsrpXs>.
5. Фрілансер по життю. Геометрические фигуры на HTML и CSS // Треугольник стрелка круг трапеция и другие, 2019. YouTube. URL:  
<https://www.youtube.com/watch?v=Irj55RYH254>.
6. Фрілансер по життю. Иконочный шрифт. подключение font awesome. создание иконочного шрифта из SVG файлов., 2020. YouTube. URL:  
<https://www.youtube.com/watch?v=QYwI3LRCnRE>.
7. Фрілансер по життю. Методология БЭМ за 17 минут. Пример верстки по БЭМ методологии // Фрилансер по жизни, 2019. YouTube. URL:  
<https://www.youtube.com/watch?v=HihYQVuH64U>.
8. Фрілансер по життю. Отзывчивый выход объекта за пределы сетки CSS HTML. интересный случай с контейнером., 2020. YouTube. URL:  
<https://www.youtube.com/watch?v=NVdlfhkDAGM>.
9. Фрілансер по життю. Препроцессор SASS/SCSS за 38 минут. что такое SASS, SCSS. установка уроки обучение., 2019. YouTube. URL:  
[https://www.youtube.com/watch?v=Mrq2ora\\_p0o](https://www.youtube.com/watch?v=Mrq2ora_p0o).
10. Фрілансер по життю. CSS grid layout. кроссбраузерность. grid и flex. основные понятия., 2020. YouTube. URL:  
<https://www.youtube.com/watch?v=GV92IdMGfFA>.
11. Фрілансер по життю. FLEXBOX. учимся верстать на флексах. флексбокс уроки., 2019. YouTube. URL:  
[https://www.youtube.com/watch?v=EO8hH\\_2OwCU](https://www.youtube.com/watch?v=EO8hH_2OwCU).
12. Фрілансер по життю. GULP 2022 установка настройка плагины. Подробный курс по настройке галпа для быстрой сборки проекта, 2021. YouTube. URL: <https://www.youtube.com/watch?v=jU88mLuLWlk>.
13. I See Sharp. MVC login and registration with identity - identity customization - .NET core 6 - part 1, 2022. YouTube. URL:  
<https://www.youtube.com/watch?v=I-ZzFLruiuo>.

		Фомін В.В.			ДУ «Житомирська політехніка». 23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

# ДОДАТКИ

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

Тут наведено основні лістинги з найбільшим функціоналом, решту можна знайти у моєму репозиторії GitHub за адресою:

<https://github.com/VladyslavF/Practice>

#### Лістинг файлу Program.cs:

```
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.EntityFrameworkCore;
using WebApplication1.Data;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();
builder.Services.AddDbContext<ShopDbContext>(
    options => options.UseSqlServer(
        builder.Configuration.GetConnectionString("ShopDbConnectionString")));

builder.Services.AddAuthentication(
    CookieAuthenticationDefaults.AuthenticationScheme)
    .AddCookie(option =>
    {
        option.LoginPath = "/Access/Login";
        option.ExpireTimeSpan = TimeSpan.FromMinutes(20);
    });
var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    // The default HSTS value is 30 days. You may want to change this for production scenarios, see
    https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    //pattern: "{controller=Home}/{action=Index}/{id?}");
    pattern: "{controller=Access}/{action=Login}/{id?}");
app.Run();
```

#### Лістинг файлу \_Layout.cshtml:

```
@using WebApplication1.Models.Shop
@model WebApplication1.Models.Shop.ShopFull
<!DOCTYPE html>
<html lang="en">
```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - WebApplication1</title>
  <link rel="stylesheet" href="~/css/null.css" asp-append-version="true"/>
  <link rel="stylesheet" href="~/css/icons.css" asp-append-version="true"/>
  <link rel="stylesheet" href="~/css/theme.css" asp-append-version="true"/>
  <link rel="stylesheet" href="~/css/burger.css" asp-append-version="true"/>
  <link rel="stylesheet" href="~/css/header.css" asp-append-version="true"/>
  <link rel="stylesheet" href="~/css/breadcrumbs.css" asp-append-version="true"/>
  <link rel="stylesheet" href="~/css/wrapper.css" asp-append-version="true"/>
  <link rel="stylesheet" href="~/css/footer.css" asp-append-version="true"/>
</head>
<body>
  <div class="wrapper">
    <div class="burger">
      <div class="burger__lines">
        <span class="burger__line1"></span>
        <span class="burger__line2"></span>
        <span class="burger__line3"></span>
      </div>
    </div>
    <header class="header active">
      <nav class="header__menu">
        <div class="header__item">
          <a href="">NEW</a>
        </div>
        <div class="header__item header__item_active">
          <a href="">КАТАЛОГ</a>
        </div>
        <div class="header__item">
          <a href="">О НАС</a>
        </div>
      </nav>
      <div class="header__logo">
        
      </div>
      <div class="header__locale">
        <div class="header__lang">
          <div class="lang__title">
            RU
          </div>
          <div class="lang__dropdown">
            <div class="lang__item">UA</div>
            <div class="lang__item">EN</div>
          </div>
          <span class="icon-arrow-black"></span>
        </div>
        <div class="header__currency">
          <div class="currency__title"> UAH </div>
          <div class="currency__dropdown">
            <div class="currency__item">USD</div>
            <div class="currency__item">EUR</div>
          </div>
          <span class="icon-arrow-black"></span>
        </div>
      </div>
      <div class="header__icons">

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <a href=""></a>
        <a href="~/Access/Login"></a>
        <a href=""></a>
        <a href="Cart"></a>
    </div>
</header>
<div class="header__submenu">
    <nav class="header__menu">
        <div class="header__item">
            <a href="">ОПЛАТА И ДОСТАВКА</a>
        </div>
        <div class="header__item">
            <a href="">УСЛОВИЯ ВОЗВРАТА</a>
        </div>
        <div class="header__item">
            <a href="">КОНТАКТЫ</a>
        </div>
    </nav>
    <div class="mobile-menu">
        <div class="mobile-menu__block">
            <div class="mobile-menu__search">
                <input type="text" id="search" placeholder="Введите ваш запрос">
                <label for="search">
                    
                </label>
            </div>
            <div class="mobile-menu__currency">
                <div class="currency__title">UAH </div>
                <div class="currency__dropdown">
                    <div class="currency__item">USD</div>
                    <div class="currency__item">EUR</div>
                </div>
                <span class="icon-arrow-black"></span>
            </div>
            <div class="mobile-menu__list">
                <div class="mobile-menu__link">
                    <span class="icon-profile-new"></span>
                    <a href="">ЛИЧНЫЙ КАБИНЕТ </a>
                </div>
                <div class="mobile-menu__link">
                    <a href="">NEW</a>
                </div>
                <div class="mobile-menu__link">
                    <a href="">КАТАЛОГ</a>
                </div>
                <div class="mobile-menu__link">
                    <a href="">О НАС</a>
                </div>
                <div class="mobile-menu__link">
                    <a href="">УСЛОВИЯ ВОЗВРАТА</a>
                </div>
                <div class="mobile-menu__link">
                    <a href="">КОНТАКТЫ</a>
                </div>
            </div>
            <div class="mobile-menu__footer">
                <div class="mobile-menu__column">

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <ul>
            <li>
                <a href=""><span class="icon-ig"></span></a>
                <a href=""><span class="icon-tg"></span></a>
            </li>
            <li>+38(073) 096 36 44</li>
            <li>info@yanki.com</li>
        </ul>
    </div>
</div>
</div>
</div>
</div>
<div class="breadcrumbs">
    <a class="breadcrumbs__link" href=" ../Home/Index">
        Главная
    </a>
    
    <a class="breadcrumbs__link" href=" ../Shop/Index">
        Каталог
    </a>
    
    <a class="breadcrumbs__link">
        Пальто
    </a>
    
    <a class="breadcrumbs__link" href="Item">
        Кремовое пальто
    </a>
</div>
@RenderBody()
<footer class="footer">

    <div class="footer__columns">

        @foreach (var item in Model.MenuItems)
        {
            <div class="footer__column">

                <div class="footer__title">
                    @item.Name.ToUpper()
                    @if (item.Name.ToUpper() != "КОНТАКТЫ")
                    {
                        <span class="icon-arrow-down-small"></span>
                    }
                </div>
                @if (item.Name.ToUpper() == "КОНТАКТЫ")
                {
                    <ul class="footer__list opened">
                        @foreach (var link in Model.MenuLinks)
                        {
                            if (item.Id == link.ItemId)
                            {
                                if (link.Name == "ICONS")
                                {
                                    <li>
                                        <a href="" class="icon-ig"></a>
                                        <a href="" class="icon-tg"></a>
                                    </li>
                                }
                            }
                        }
                    </ul>
                }
            }
        }
    </div>

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    else
    {
        <li><a href="@link.Url">@link.Name</a></li>
    }
}
</ul>
}
else {
    <ul class="footer__list">
        @foreach (var link in Model.MenuLinks)
        {
            if (item.Id == link.ItemId)
            {
                if (link.Name == "ICONS")
                {
                    <li>
                        <a href="" class="icon-ig"></a>
                        <a href="" class="icon-tg"></a>
                    </li>
                }
                else
                {
                    <li><a href="@link.Url">@link.Name</a></li>
                }
            }
        }
    </ul>
}
</div>
}
<!--
<div class="footer__column">
    <div class="footer__title">
        КОМПАНИЯ <span class="icon-arrow-down-small"></span>
    </div>
    <ul class="footer__list">
        <li><a href="">О нас</a></li>
        <li><a href="">Контакты</a></li>
    </ul>
</div>
<div class="footer__column">
    <div class="footer__title" id="useful">
        ПОЛЕЗНОЕ <span class="icon-arrow-down-small"></span>
    </div>
    <ul class="footer__list">
        <li><a href="">Оплата и доставка</a></li>
        <li><a href="">Условия возврата</a></li>
        <li><a href="">Бонусная система</a></li>
    </ul>
</div>
<div class="footer__column">
    <div class="footer__title" id="to-buyer">
        ПОКУПАТЕЛЮ <span class="icon-arrow-down-small"></span>
    </div>

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				48
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        <ul class="footer__list">
            <li><a href="">Избранное</a></li>
            <li><a href="">Публичная оферта</a></li>
            <li><a href="">Политика конфиденциальности</a></li>
        </ul>
    </div>
    <div class="footer__column">
        <div class="footer__title">
            КОНТАКТЫ
        </div>
        <ul class="footer__list opened">
            <li>
                <a href="" class="icon-ig"></a>
                <a href="" class="icon-tg"></a>
            </li>
            <li>+38(073) 096 36 44</li>
            <li>info@yanki.com</li>
        </ul>
    </div>
    -->
</div>
<div class="footer__copyright">
    © 2021 Yanki. All rights reserved
</div>
</footer>
</div>
<script src="~/js/header.js" defer></script>
<script src="~/js/footer.js" defer></script>
@await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```

#### Лістинг файлу Shop/Index.cshtml

```

@*
    For more information on enabling MVC for empty projects, visit
    https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
    ViewData["Title"] = "YANKI - Catalog";
}
@using WebApplication1.Models.Shop
@model ShopFull
@{
    <div class="main">
        <div class="main__filter-title">
            Фильтры
            
            
        </div>
        <div class="main__filters">
            <div class="main__filter-item">
                <div class="main__filter-subtitle">
                    Размер <span class="icon-arrow-down-small"></span>
                </div>
                <div class="main__subfilters">
                    <div class="subfilters__item">
                        S
                    </div>
                </div>
            </div>
        </div>
    </div>

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <div class="subfilters__item">
            М
        </div>
        <div class="subfilters__item">
            L
        </div>
    </div>
</div>
<div class="main__filter-item">
    <div class="main__filter-subtitle"> Цвет <span class="icon-arrow-down-small"></span> </div>
    <div class="main__subfilters">
        <div class="subfilters__item">
            Синий
        </div>
        <div class="subfilters__item">
            Кремовый
        </div>
        <div class="subfilters__item">
            Белый
        </div>
    </div>
</div>
<div class="main__filter-item">
    <div class="main__filter-subtitle"> Цена <span class="icon-arrow-down-small"></span> </div>
    <div class="main__subfilters">
        <div class="subfilters__item">
            1000 - 2000
        </div>
        <div class="subfilters__item">
            2000 - 3000
        </div>
        <div class="subfilters__item">
            3000+
        </div>
    </div>
</div>
<div class="main__filter-item">
    <div class="main__filter-subtitle"> Сортировать по <span class="icon-arrow-down-
small"></span> </div>
    <div class="main__subfilters">
        <div class="subfilters__item">
            Возрастанию
        </div>
        <div class="subfilters__item">
            Убыванию
        </div>
        <div class="subfilters__item">
            Рейтингу
        </div>
    </div>
</div>
</div>
<main class="main__gallery">

@foreach (var item in Model.products)
{
    <div class="main__item">
        <div class="main__fav-button">

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <span class="icon-favourites"></span>
    </div>
    
    <div class="main__name @( "main__name_" + item.Status)">
        @item.Name
    </div>
    <div class="main__price">
        @item.Price грн
    </div>
    <div class="main__sizes">

    @foreach (var size in Model.Product_sizes)
    {
        if (size.ProductId == item.Id)
        {
            @foreach (var main_size in Model.sizes)
            {
                if (main_size.Id == size.SizeId)
                {
                    <div class="main__size">
                        @main_size.Name
                    </div>
                }
            }
        }
    }
    </div>
    <div class="main__colors">
    @foreach (var color in Model.colors)
    {
        <div class="main__color">
            
        </div>
    }

    </div>
</div>
}
</main>
<aside class="main__sidebar">
    <div class="main__title">
        Каталог <span class="icon-arrow-down"></span>
    </div>
    <div class="main__links">
        <a href="">New</a>
        <a href="">Bestsellers</a>
        @foreach (var category in Model.categories)
        {
            <a href="">@category.Name</a>
        }
        <a href="">Sale</a>
        <a href="">Summer</a>
        <a href="">Посмотреть всё</a>
    </div>
</aside>
</div>
<script src="~/js/catalog.js" defer></script>

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

```
<link rel="stylesheet" href="~/css/catalog.css"/>
}
```

### Лістинг файлу HomeController.cs:

```
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;
using WebApplication1.Data;
using WebApplication1.Models;
using WebApplication1.Models.Shop;
using Microsoft.AspNetCore.Authentication;
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.AspNetCore.Authorization;

namespace WebApplication1.Controllers
{
    [Authorize]
    public class HomeController : Controller
    {

        protected ShopDbContext shopDbContext;
        public HomeController(ShopDbContext shopDbContext)
        {
            this.shopDbContext = shopDbContext;
        }
        public IActionResult Index()
        {
            var model = new ShopFull();
            model.MenuItems = shopDbContext.MenuItems.ToList();
            model.MenuLinks = shopDbContext.Links.ToList();
            return View(model);
        }

        public IActionResult Privacy()
        {
            return View();
        }

        public async Task<IActionResult> Logout()
        {
            await HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
            return RedirectToAction("Login", "Access");
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier
        });
        }
    }
}
```

### Лістинг файлу AccessController.cs:

```
using Microsoft.AspNetCore.Mvc;
using System.Security.Claims;
using Microsoft.AspNetCore.Authentication;
```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using Microsoft.AspNetCore.Authentication.Cookies;
using WebApplication1.Models;
using WebApplication1.Models.Shop;
using WebApplication1.Data;

namespace WebApplication1.Controllers
{
    public class AccessController : Controller
    {
        protected ShopDbContext shopDbContext;
        public AccessController(ShopDbContext shopDbContext)
        {
            this.shopDbContext = shopDbContext;
        }
        public IActionResult Login()
        {
            ClaimsPrincipal claimUser = HttpContext.User;
            if (claimUser.Identity.IsAuthenticated) {
                //return RedirectToAction("Index", "Home");
                return RedirectToAction("Profile", "Access");
            }

            return View();
        }
        [HttpPost]
        public async Task<IActionResult> Login(User modelLogin)
        {
            var db = new ShopFull();
            db.Users = shopDbContext.Users.ToList();
            foreach (var user in db.Users)
            {
                if (modelLogin.Email == user.Email)
                {
                    if (modelLogin.Password == user.Password)
                    {
                        List<Claim> claims = new List<Claim>() {
                            new Claim(ClaimTypes.NameIdentifier, modelLogin.Email),
                            new Claim("OtherProperties", "Other Role")
                        };

                        ClaimsIdentity claimsIdentity = new ClaimsIdentity(claims,
                            CookieAuthenticationDefaults.AuthenticationScheme);
                        AuthenticationProperties properties = new AuthenticationProperties()
                        {
                            AllowRefresh = true,
                            IsPersistent = modelLogin.KeepLoggedIn
                        };
                        await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme,
                            new ClaimsPrincipal(claimsIdentity), properties);

                        return RedirectToAction("Index", "Home");
                    }
                }
            }
            ViewData["ValidateMessage"] = "User not found";
            return View();
        }
    }
}

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public IActionResult Register()
{
    ClaimsPrincipal claimUser = HttpContext.User;
    if (claimUser.Identity.IsAuthenticated)
    {
        return RedirectToAction("Index", "Home");
    }

    return View();
}
public IActionResult Profile()
{
    return View();
}
}
}

```

### Лістинг файлу ProductController.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using WebApplication1.Data;
using WebApplication1.Models.Shop;

namespace WebApplication1.Controllers
{
    public class ProductsController : Controller
    {
        private readonly ShopDbContext _context;

        public ProductsController(ShopDbContext context)
        {
            _context = context;
        }
        // GET: Products
        public async Task<IActionResult> Index()
        {
            var userID = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
            if (userID != "admin@local")
            {
                return RedirectToAction("Index", "Home");
            }
            var shopDbContext = _context.Products.Include(p => p.Category).Include(p => p.Color);
            return View(await shopDbContext.ToListAsync());
        }

        // GET: Products/Details/5
        public async Task<IActionResult> Details(int? id)
        {
            var userID = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
            if (userID != "admin@local")

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            return RedirectToAction("Index", "Home");
        }
        if (id == null || _context.Products == null)
        {
            return NotFound();
        }

        var product = await _context.Products
            .Include(p => p.Category)
            .Include(p => p.Color)
            .FirstOrDefaultAsync(m => m.Id == id);
        if (product == null)
        {
            return NotFound();
        }

        return View(product);
    }

    // GET: Products/Create
    public IActionResult Create()
    {
        var userID = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
        if (userID != "admin@local")
        {
            return RedirectToAction("Index", "Home");
        }
        ViewData["CategoryId"] = new SelectList(_context.Categories, "Id", "Name");
        ViewData["ColorId"] = new SelectList(_context.Colors, "Id", "Name");
        return View();
    }

    // POST: Products/Create
    // To protect from overposting attacks, enable the specific properties you want to bind to.
    // For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult>
Create([Bind("Id,Name,Price,CategoryId,ImageUrl,ColorId,Description,Ingredients,Status")] Product product)
    {
        if (ModelState.IsValid)
        {
            _context.Add(product);
            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
        ViewData["CategoryId"] = new SelectList(_context.Categories, "Id", "Id", product.CategoryId);
        ViewData["ColorId"] = new SelectList(_context.Colors, "Id", "Id", product.ColorId);
        return View(product);
    }

    // GET: Products/Edit/5
    public async Task<IActionResult> Edit(int? id)
    {
        var userID = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
        if (userID != "admin@local")
        {

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return RedirectToAction("Index", "Home");
    }
    if (id == null || _context.Products == null)
    {
        return NotFound();
    }

    var product = await _context.Products.FindAsync(id);
    if (product == null)
    {
        return NotFound();
    }
    ViewData["CategoryId"] = new SelectList(_context.Categories, "Id", "Name", product.CategoryId);
    ViewData["ColorId"] = new SelectList(_context.Colors, "Id", "Name", product.ColorId);
    return View(product);
}

// POST: Products/Edit/5
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(int id,
[Bind("Id,Name,Price,CategoryId,ImageUrl,ColorId,Description,Ingredients,Status")] Product product)
{
    var userID = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
    if (userID != "admin@local")
    {
        return RedirectToAction("Index", "Home");
    }
    if (id != product.Id)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(product);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!ProductExists(product.Id))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    ViewData["CategoryId"] = new SelectList(_context.Categories, "Id", "Id", product.CategoryId);
    ViewData["ColorId"] = new SelectList(_context.Colors, "Id", "Id", product.ColorId);
    return View(product);
}

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				56
Змн.	Арк.	№ докум.	Підпис	Дата		



```

    }

    // GET: Products/Delete/5
    public async Task<IActionResult> Delete(int? id)
    {
        var userID = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
        if (userID != "admin@local")
        {
            return RedirectToAction("Index", "Home");
        }
        if (id == null || _context.Products == null)
        {
            return NotFound();
        }

        var product = await _context.Products
            .Include(p => p.Category)
            .Include(p => p.Color)
            .FirstOrDefaultAsync(m => m.Id == id);
        if (product == null)
        {
            return NotFound();
        }

        return View(product);
    }

    // POST: Products/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> DeleteConfirmed(int id)
    {
        if (_context.Products == null)
        {
            return Problem("Entity set 'ShopDbContext.Products' is null.");
        }
        var product = await _context.Products.FindAsync(id);
        if (product != null)
        {
            _context.Products.Remove(product);
        }

        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }

    private bool ProductExists(int id)
    {
        return _context.Products.Any(e => e.Id == id);
    }
}

```

#### Лістинг файлу ShopDbContext.cs:

```

using Microsoft.EntityFrameworkCore;
using WebApplication1.Models.Menu;
using WebApplication1.Models.Shop;

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

namespace WebApplication1.Data
{
    public class ShopDbContext : DbContext
    {
        public DbSet<Item> Menu { get; set; }
        public DbSet<Category> Categories { get; set; }
        public DbSet<Product> Products { get; set; }
        public DbSet<Color> Colors { get; set; }
        public DbSet<Size> Sizes { get; set; }
        public DbSet<User> Users { get; set; }
        public DbSet<Product_Sizes> Product_Sizes { get; set; }
        public DbSet<Models.Menu.Item> MenuItems { get; set; }
        public DbSet<Links> Links { get; set; }
        public ShopDbContext(DbContextOptions options) : base(options)
        {
        }
    }
}

```

### Лістинг файлу ShopFull.cs:

```

using WebApplication1.Models.Menu;
namespace WebApplication1.Models.Shop
{
    public class ShopFull
    {
        public IEnumerable<Product> products { get; set; }
        public IEnumerable<Category> categories { get; set; }
        public IEnumerable<Color> colors { get; set; }
        public IEnumerable<Size> sizes { get; set; }
        public IEnumerable<Product_Sizes> Product_sizes { get; set; }

        public IEnumerable<Menu.Item> MenuItems { get; set; }
        public IEnumerable<Menu.Links> MenuLinks { get; set; }

        public IEnumerable<User> Users { get; set; }
    }
}

```

		Фомін В.В.			ДУ «Житомирська політехніка».23.125.21.000 - ПЗ	Арк.
		Окунькова О.О.				58
Змн.	Арк.	№ докум.	Підпис	Дата		