

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

01 червня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 – Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Адаптивний інтернет-магазин військового спорядження»
здобувача групи ІН – 03 Ніколаєнка Мирослава Олеговича

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

Мирослав НІКОЛАЄНКО

(підпис)

Керівник,

асистент кафедри комп'ютерних наук

кандидат фізико-математичних наук

Олександр ВЛАСЕНКО

(підпис)

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

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 – Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-03 Ніколаєнка Мирослава Олеговича

1. Тема роботи: «Адаптивний інтернет-магазин військового спорядження»
затверджую наказом по СумДУ від «22» квітня 2024р. № 0414-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 01 червня 2024 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, ознайомлення з існуючими рішеннями, постановка й формування завдань дослідження. 2) Огляд засобів та інструментів. 3) Розробка дизайну адаптивного інтернет-магазину військового спорядження. 4) Програмна реалізація 5) Аналіз отриманих результатів розробки.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «08» квітня 2024 р.

Завдання прийняв до виконання _____
(підпис)

Керівник _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, ознайомлення з існуючими рішеннями, постановка й формування завдань дослідження.</i>	06.05.24-07.05.24	
2	<i>Огляд засобів та інструментів</i>	08.05.24-09.05.24	
3	<i>Розробка дизайну адаптивного інтернет-магазину військового спорядження</i>	09.05.24-12.05.24	
4	<i>Програмна реалізація</i>	13.05.24-16.05.24	
5	<i>Аналіз отриманих результатів розробки.</i>	17.05.24-18.05.24	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	18.05.24- 20.05.24	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

АНОТАЦІЯ

Записка: 56 стор., 44 рисунки, 1 таблиця, 2 додатки, 17 використаних джерел.

Обґрунтування актуальності теми роботи – тема кваліфікаційної роботи є дуже актуальною, оскільки сучасний світ насичений викликами та загрозами, які ставлять перед нами питання безпеки та захисту. У такому контексті, інтернет-магазини військового спорядження набувають особливого значення. Вони не лише надають можливість придбати необхідні засоби захисту, а й створюють унікальну платформу, де клієнти можуть знайти все необхідне для своєї безпеки та безпеки своїх рідних.

Об’єкт дослідження – адаптивний інтернет-магазини військового спорядження.

Мета роботи – розробка адаптивного інтернет-магазину військового спорядження, який дозволить спрощено замовляти товар у сфері інтернет магазинів.

Методи дослідження – ознайомлення з програмними засобами та інструментами для розробки адаптивного дизайну інтернет-магазину військового спорядження. Розробка дизайну та функціоналу, що буде оптимально працювати на різних пристроях з різними розмірами екранів.

Результати – розроблено front-end частину інтернет-магазин військового спорядження, який задовольняє потреби користувачів, надаючи унікальний користувацький досвід, це важливо. Забезпечуючи адаптивність сторінок під різні пристрої, магазин дозволяє користувачам зручно переглядати та придбати товари навіть на мобільних пристроях.

ІНТЕРНЕТ-МАГАЗИН, АДАПТИВНИЙ ДИЗАЙН, HTML, CSS, GULP,
JAVASCRIPT, SASS, JSON

ЗМІСТ

ВСТУП	5
1. АНАЛІТИЧНИЙ ОГЛЯД.....	6
1.1. Основні етапи створення інтернет-магазину.....	6
1.2. Ознайомлення з існуючими рішеннями.....	7
1.3. Постановка задачі.....	11
2. ВИБІР МЕТОДУ РІШЕННЯ.....	13
2.1. Вибір засобів та інструментів	13
2.2. Вибір системи керування версіями проєкту.....	19
2.3. Розробка структури сайту.....	20
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	22
3.1. Створення дизайну сайту.....	22
3.2. Підготовка до проєкту	27
3.3. Верстка інтернет-магазину.....	31
3.4. Перевірка адаптивності.....	39
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
Додаток А.....	48
Додаток Б	51

ВСТУП

Актуальність. У сучасному цифровому світі інтернет-магазини стали невід'ємною частиною нашого способу життя. За допомогою всесвітньої мережі, ми можемо придбати практично будь-який товар, не виходячи з дому. Однак, у зв'язку з постійним розвитком технологій та змінами у споживацькому поведінці, інтернет-магазини також повинні адаптуватися до потреб користувачів, що швидко змінюються. У цьому контексті, поняття "адаптивний інтернет-магазин" набуває все більшого значення. Адаптивність означає здатність пристосовуватися до різних умов і змінюватися залежно від контексту. У контексті інтернет-магазинів, це означає створення таких платформ, які забезпечують комфортне та ефективне спілкування з клієнтами незалежно від їхнього пристрою, операційної системи чи розміру екрану.

Об'єкт дослідження. Адаптивний інтернет-магазин військового спорядження.

Предмет дослідження. Методи створення інтернет-магазинів.

Гіпотеза. Одним із ключових аспектів адаптивних інтернет-магазинів є респонсивний дизайн. Це означає, що вебсторінка адаптується до різних розмірів екранів. Користувачі, відкриваючи інтернет-магазин на своєму мобільному пристрої, повинні мати таку ж зручність та функціональність, як і при відкритті на комп'ютері.

Новизна. Новизна роботи полягає в тому, що написання інтернет-магазину на чистому JavaScript для забезпечення високої продуктивності та інтерактивності користувацького інтерфейсу. Крім того, впроваджено унікальні рішення для адаптивного дизайну, які дозволяють інтернет-магазину бездоганно функціонувати на різних типах пристроїв, від мобільних телефонів до настільних комп'ютерів.

Структура. Робота складається зі вступу, аналітичного огляду, постановки задачі, вибір методу рішення поставленої задачі, опису програмного забезпечення інформаційної системи, висновку, списку використаних джерел та додатків.

1. АНАЛІТИЧНИЙ ОГЛЯД

1.1. Основні етапи створення інтернет-магазину

Сьогодні просування власного бізнесу в Інтернеті є найактуальнішою тенденцією. Всесвітня мережа відкриває безліч можливостей для розвитку, тут можна досить ефективно реалізовувати товари і послуги. Саме тому все більше людей цікавиться тим, як відкрити інтернет-магазин. Даний рід діяльності може бути дуже прибутковим, якщо розумно підійти до питання. Те, наскільки успішним буде сайт, залежить від ряду факторів, які варто розібрати докладніше. Власне кажучи, є кілька етапів:

Етап перший вибір роду діяльності – потрібно визначитися, що створюваний вами інтернет-магазин буде продавати. Це може бути одяг або побутова техніка, косметика, посуд тощо. Можна також створити універсальний магазин, у якому є всі популярні товари різноманітних категорій. Але не обов'язково ставити такі складні завдання на початковому етапі. Багато відомих інтернет-магазини починали з вузькопрофільної діяльності, а потім поступово розширювали свій асортимент. Тому краще наслідувати їх приклад, це буде значно простіше і ефективніше.

Дизайн – це другий етап створення вебсайту. На цьому етапі продумується і розробляється дизайн сайту, який привертатиме увагу користувачів та допомагатиме їм легко знайти необхідну інформацію. Дизайн повинен бути простим, легко читатися та зручним у використанні.

Третій етап – це верстка є важливою складовою розробки вебсайтів і додатків. Вона включає процес створення структури, оформлення та розміщення елементів на вебсторінці з використанням мов розмітки, таких як HTML, CSS та іноді JavaScript. При верстці необхідно враховувати різні аспекти, такі як респонсивний дизайн (щоб сторінка коректно відображалася на різних пристроях), кросбраузерність (сумісність з різними вебпереглядачами), доступність (забезпечення доступу до контенту для людей з обмеженими

можливостями), швидкодію (оптимізація завантаження сторінки) та інші.

Четвертий етап – це тестування і оптимізація, можна сказати, що після завершення процесу верстки і розробки вебсайту важливо провести тестування для перевірки його функціональності та виявлення можливих помилок. Тестування допомагає забезпечити, що всі компоненти вебсайту працюють належним чином і забезпечують задану функціональність. Під час тестування слід перевірити, чи працюють всі інтерактивні елементи, форми, посилання, анімації тощо. Також важливо впевнитися, що вебсторінка коректно відображається на різних пристроях і в різних веб-переглядачах. Після успішного завершення тестування можна перейти до етапу оптимізації. Оптимізація включає ряд заходів для покращення продуктивності та ефективності вебсайту. Це може включати оптимізацію швидкості завантаження сторінок, зменшення розміру файлів, використання кешування.

П'ятий етап – це запуск. Після завершення всіх етапів створення сайту, включаючи верстку, тестування та оптимізацію, настав час запуснути ваш сайт. На цьому етапі можна завантажити сайт на хостинг та підключити доменне ім'я до сайту. Після запуску сайту необхідно регулярно оновлювати його контент та функціональність.

1.2. Ознайомлення з існуючими рішеннями

На сьогоднішній день в Інтернеті присутня значна кількість інтернет-магазинів, що спеціалізуються на продажу військового одягу та екіпіровки. Перед розробкою інтернет-магазину військового одягу, важливо провести аналіз ринку та дослідити існуючі аналоги. Було обрано та оглянуто три існуючих сайти:

- abrams.com.ua
- militarist.ua
- agressor.com.ua

1.2.1 Інтернет-магазин Abrams.com.ua

Abrams.com.ua – це один із найпопулярніших з військових інтернет-магазинів в Україні. У магазині дуже велика вибірка товарів, кожен покупець зможе знайти собі товар який він шукає. У цьому магазині сучасний дизайн сайту. Головну сторінку сайту можна побачити на рисунку 1.1. Серед переваг можу виділити слайдер, на ньому можна побачити інформацію про нові товари, акції та знижки. Проте, серед недоліків можна відзначити, що розділи "Популярні" та "Нові надходження" на головній сторінці можуть бути трохи неясними. Не завжди зрозуміло, які саме товари входять до розділу "Нові надходження", а які – до "Популярних". Це може призвести до певної плутанини для користувачів, які хочуть швидко знайти певну категорію товарів.

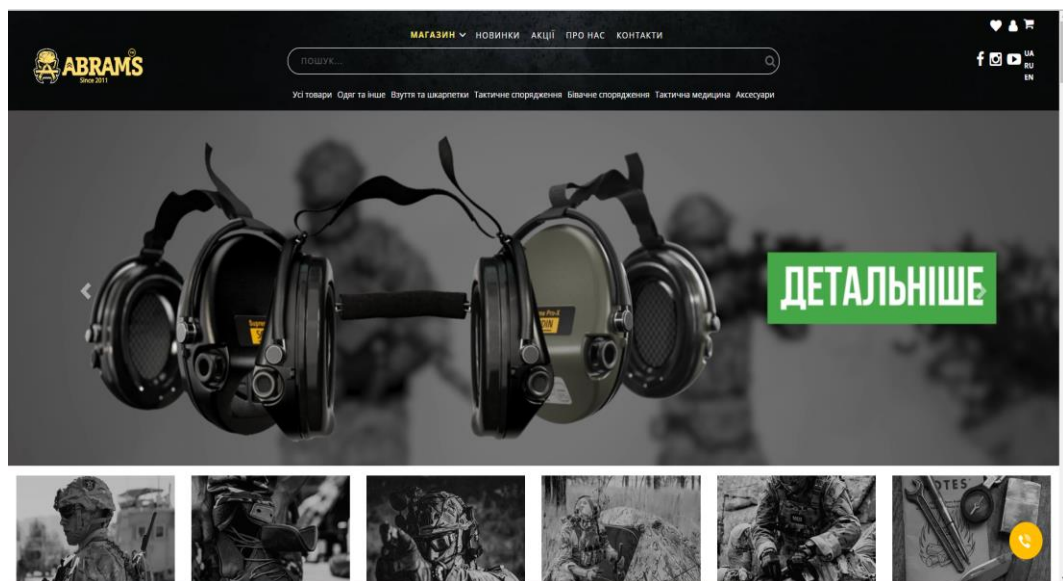


Рисунок 1.1 – Головна сторінка Abrams.com.ua

Як можна побачити, на сайті є дуже важливі елементи як:

- каталог товарів;
- слайдер;
- корзина;
- пошук товарів;
- контакти де вони знаходяться і номери телефонів.

Це все демонструє гарну структуру, яка забезпечує комфортне користування сайтом.

1.2.2 Інтернет-магазин Militarist.ua

Militarist.ua – це найпопулярніших військових інтернет-магазинів в Україні, який можна знайти за адресою <https://militarist.ua/ua/>. У цьому магазині можна знайти великий вибір товарів.

Один із головних плюсів Militarist.ua – сучасний дизайн його вебсайту. На головній сторінці бачимо слайдер, який розміщений на рисунку 1.2. Він демонструє інформацію про нові товари, акції та знижки. Це дозволяє швидко ознайомитися з останніми новинками та вигідними пропозиціями.

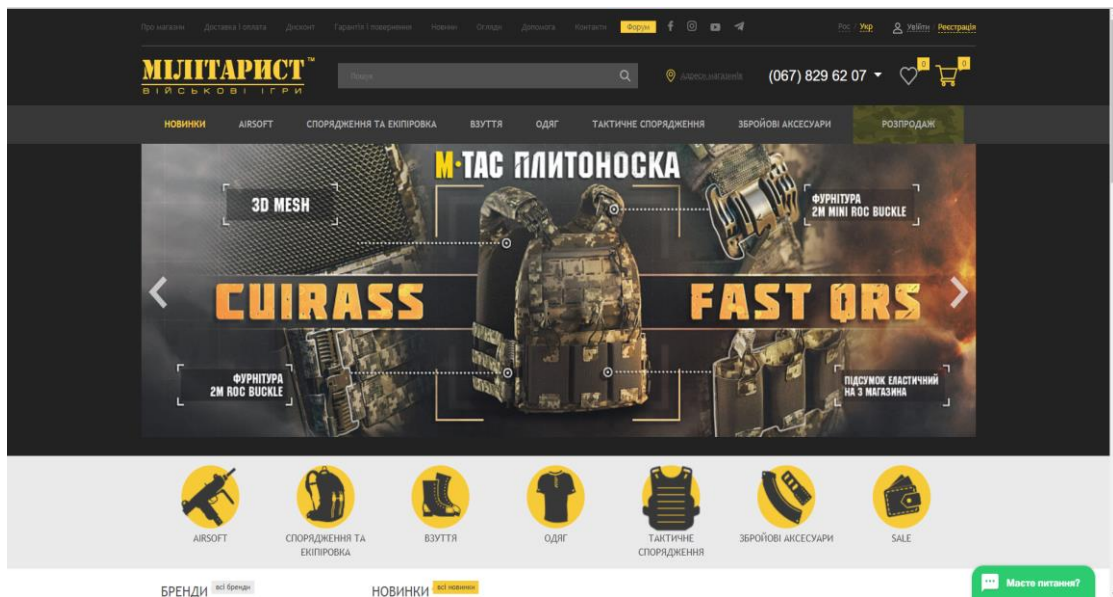


Рисунок 1.2 – Головна сторінка Militarist.ua

На сайті також є дуже важливі елементи такі як:

- слайдер;
- пошук товарів;
- меню;
- каталог товарів;
- бренди;

- новинки;
- переваги.

Militarist.ua продовжує бути популярним магазином завдяки своєму широкому асортименту військового спорядження та продукції, що задовольняє потреби клієнтів.

1.2.3 Інтернет-магазин Agressor.com.ua

Agressor.com.ua – це ще один популярний військовий інтернет-магазин в Україні, який можна знайти за адресою <https://agressor.com.ua/ua>. Цей магазин пропонує широкий асортимент товарів для задоволення потреб військових ентузіастів.

Одна з головних переваг Agressor.com.ua – це сучасний дизайн його вебсайту. На головній сторінці є слайдер (рис. 1.3), який дозволяє швидко ознайомитися з останніми надходженнями та вигідними пропозиціями. Також на сайті присутній зручний пошуковий рядок, який допомагає знайти потрібний товар швидко і ефективно. Серед інших переваг Agressor.com.ua можна виділити великий асортимент товарів. Незалежно від того, що необхідно придбати – одяг, взуття, рюкзаки, або засоби особистої безпеки, все знайдеться в одному місці.

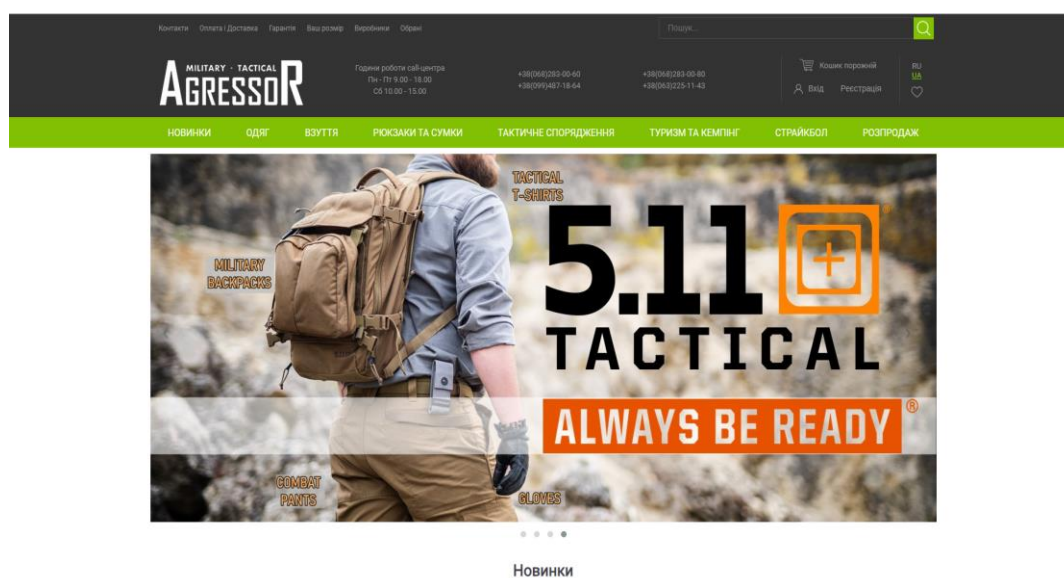


Рисунок 1.3 – Головна сторінка Agressor.com.ua

У будь-якому разі, Agressor.com.ua є привабливим вибором для тих, хто шукає військові товари в Україні. З його широким асортиментом, сучасним дизайном та можливістю знайти акційні пропозиції, цей магазин може задовольнити потреби клієнтів, зацікавлених у військовій продукції. Представимо аналіз сайтів у вигляді таблиці 1.1

Таблиця 1.1 – Порівняльний аналіз інтернет-магазинів військового одягу

Критерії	Abrams.com.ua	Militarist.ua	Agressor.com.ua
Дизайн сайту	+	+	+
Каталог товарів	+	+	+
Фільтр товарів	+	-	+
Адаптивність	+	+	+
Зручність у використанні	-	+	+
Швидкість завантаження	+	+	+
Контакти	+	+	+

Було розглянуто три різних військових інтернет-магазини, і виявлено, що вони мають багато спільних рис, але також відрізняються один від одного. Кожен з них має свої переваги і недоліки, які можна врахувати для створення більш конкурентоспроможного інтернет-магазину.

1.3. Постановка задачі

Оскільки метою роботи є розробка структури адаптивного інтернет-магазину з продажу військового одягу, то проаналізувавши аналогічні рішення, та визначивши необхідні вимоги для сайту, можна зробити висновки, що інтернет-сторінка повинна мати привабливий та сучасний дизайн, гарний шрифт, інтуїтивно зрозумілий інтерфейс.

Перейдемо до написання необхідних задач, які потребують реалізації. Результат онлайн магазину повинен містити наступні сторінки та функціонал:

- Головна сторінка повинна містити інформація щодо контактів для зв'язку та фізичного розташування магазинів у місті, головне меню із швидкою навігацією за категоріями та каталог товару, функціонал (пошук по сайту, сортування товарів);

- сайт повинен бути адаптивним, однаково гарно відображатися на різних моніторах та пристроях;

- сторінка з описом конкретного продукту повинна містити опис продукту, його ціну, зображення;

- сторінка реєстрації у магазині;

- сторінка особистого кабінету;

- сторінка оформлення замовлення має містити форму для вказання даних отримувача, вибору способу доставки;

- Кошик повинен відображати усі товари, додані до нього.

Також слід не забути про те, що сайт повинен швидко та ефективно оброблювати усі дії користувача.

2. ВИБІР МЕТОДУ РІШЕННЯ

2.1. Вибір засобів та інструментів

Перед початком розробки будь-якого проєкту проходить безліч процесів планування і прийняття рішень. Для створення сучасного інтернет-магазину використовуються різноманітні інструменти та технології. Для полегшення та швидкості роботи використовуються препроцесори Sass та формат обміну даними JSON. Також у процесі розробки використовується автоматизація завдань з допомогою інструментів, таких як Gulp і Node.js. Для створення дизайну використовується графічний редактор Figma. Розглянемо обрані інструменти для виконання поставлених задач.

Для роботи з кодом було вибрано Visual Studio Code – це редактор, який можна використовувати з різними мовами програмування, включаючи C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust. Він базується на структурі Electron, яка використовується для розробки вебзастосунків Node.js, які працюють на механізмі компонування Blink. Visual Studio Code використовує той самий компонент редактора (під кодовою назвою «Monaco»), який використовується в Azure DevOps[1].

HTML – стандартизована мова розмітки документів для перегляду вебсторінок у браузері. Браузери отримують HTML документ від сервера за протоколами HTTP/HTTPS або відкривають з локального диска, далі інтерпретують код в інтерфейс, який відобразатиметься на екрані монітора. У HTML текст позначається за допомогою тегів. Кожен HTML документ буде складатися з деякої групи елементів, де кожен елемент буде визначатися (починатися та закінчуватися) певним тегом (але слід також звернути увагу, що для деяких елементів кінцевий тег не є обов'язковим). Тег – це назва елемента, записана у кутових дужках[3].

```

<section class="start">
  <div class="container">
    <div class="start_wrapper">
      <div class="start_body">
        <div class="start_column">
          <h2 class="start_title">Ready to start scaling <br> your business now?</h2>
          <div class="start_text">
            <p>Lorem ipsum dolor sit am consectetur adipiscing varius enim in eros.</p>
          </div>
        </div>
        <div class="start_column">
          <div class="start-actions">
            <a href="#" class="start-actions_button button">Our services</a>
            <a href="#" class="start-actions_button button button_dark">Contact us</a>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

```

Рисунок 2.1 – Приклад HTML коду

Cascading Style Sheets – це спеціальна мова стилів, за допомогою якої оформлюються усі веб сторінки, написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі розмічені мовою HTML, XHTML та XML. CSS стилі можна зручно зберігати у вигляді окремого текстового файлу, та додавати його за необхідності у початку коду[4].

```

.header {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  box-sizing: border-box;
  height: 100vh;
  padding: 64px 120px;
  background-image: url(img/header-bg.png);
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

```

Рисунок 2.2 – Приклад CSS коду

JavaScript – мова програмування, що дозволяє реалізувати ряд складних рішень в вебдокументах. Вона допомагає зробити сторінки сайту більш

інтерактивною, обробляючи дії користувачів сайту. Це об'єктно-орієнтована клієнтська мова, яка підтримується додатками, що працюють з дизайном сайту. JavaScript став ще більш популярним серед розробників, коли з'явилася AJAX-технологія, що призвело до нового етапу в розробці сайтів. За допомогою цієї мови можна додавати функції та сценарії, що розпізнають та реагують на дії користувача[5].

```
document.addEventListener("click", documentActions);

function documentActions(e) {
  const targetElement = e.target;
  //tabs
  if (targetElement.closest('.popular-header__item')) {
    const tabNavItem = targetElement.closest('.popular-header__item');
    if (!tabNavItem.classList.contains('active')){
      const activeTabNavItem = document.querySelector('.popular-header__item.active');
      activeTabNavItem.classList.remove('active');
      tabNavItem.classList.add('active');

      const tabItems = document.querySelectorAll('.popular-content__tab');
      const activeTabItem = document.querySelector('.popular-content__tab.active');

      activeTabItem.classList.remove('active');
      tabItems[getIndex(tabNavItem)].classList.add('active');
    }
  }
}
```

Рисунок 2.3 – Приклад JavaScript коду

SASS (Syntactically Awesome Style Sheets) – це препроцесор CSS, що означає, що він розширює можливості CSS, роблячи його більш потужним та простим у використанні. SASS написаний мовою програмування Ruby, але можна використовувати з будь-яким проектом веброботи. Sass має два синтаксиси, проте .scss використовується найчастіше. Це надбудова CSS, що означає, що всі дійсні CSS також є дійсними SCSS. Синтаксис є більш незвичайним: він використовує відступи, а не фігурні дужки для вкладення операторів, і символи нового рядка замість крапки з комою для їх розділення[6].

```
.footer {  
  background: #343A40;  
  padding: 30px;  
  &__wrapper {  
    display: flex;  
    justify-content: space-between;  
    gap: 15px;  
  }  
  &__text {  
    color: #E5E5E5;  
    font-weight: 700;  
    line-height: 1.28;  
    text-transform: uppercase;  
  }  
  &__up {  
    background: none;  
    color: #E5E5E5;  
    &:hover {  
      text-decoration: underline;  
    }  
  }  
}
```

Рисунок 2.4 – Приклад SASS коду

Node.js є відкритим рішенням для виконання JavaScript-коду на сервері. Основною перевагою Node.js є його асинхронна та подійно-орієнтована модель. Замість того, щоб чекати, поки виконається певна операція (наприклад, читання файлу чи виконання запиту до бази даних), Node.js продовжує виконувати інші задачі, після чого повернеться до обробки результатів. Це дозволяє створювати додатки, які масштабуються дуже ефективно та виконуються швидко, особливо в ситуаціях з великою кількістю одночасних з'єднань, таких як чат-додатки, стрімінгові сервіси та інші застосунки реального часу.

Крім того, Node.js має велику екосистему бібліотек та фреймворків, що значно спрощує створення різноманітних додатків. Наприклад, Express.js і `lprdjzkz` створювати вебдодатки, Socket.IO – реалізувати взаємодію в режимі реального часу, а npm (Node Package Manager) допомагає керувати залежностями проєкту. Загалом, Node.js є потужним інструментом для розробки серверних додатків, який дозволяє розробникам використовувати ту саму мову програмування на обох сторонах – клієнтській та серверній.

Gulp є системою збірки, це означає, що можна використовувати його для

автоматизації типових завдань при розробці сайту[12]. GULP – це інструмент для автоматизації рутинної роботи веброзробки. Він допомагає зменшити кількість ручної роботи, що зазвичай пов'язана з розробкою вебсайтів, таких як мінімізація файлів CSS та JavaScript, оптимізація зображень, автоматична перезавантаження сторінки під час розробки тощо.

Основна перевага GULP полягає у тому, що він працює на базі JavaScript і використовує стріми (stream), що дозволяє швидше виконувати завдання порівняно з аналогічними інструментами. Крім того, GULP має велику кількість плагінів, які розширюють його функціональність і дозволяють виконувати різноманітні завдання з автоматизації веб-розробки. Наприклад, можна зробити конфігураційний файл GULP для автоматичної компіляції препроцесорних мов, таких як Sass або LESS, у CSS, або для автоматичного стискання зображень перед розгортанням вебсайту в продакшн. Загалом, GULP є потужним інструментом, який спрощує процес веброзробки, збільшуючи продуктивність і допомагаючи розробникам швидше створювати якісний вебконтент.

```
1  gulp.task('styles', function() {
2    return gulp.src("src/sass/**/*.+(scss|sass)")
3      .pipe(sass({ outputStyle: 'expanded' })).on('error', sass.logError))
4      .pipe(rename({suffix: '.min', prefix: ''}))
5      .pipe(autoprefixer())
6      .pipe(cleanCSS({compatibility: 'ie8'}))
7      .pipe(gulp.dest("dist/css"))
8      .pipe(browserSync.stream());
9  });
```

Рисунок 2.4 – Приклад Gulp коду

JavaScript Object Notation – це легкий формат обміну даними, який часто використовується для передачі даних між вебсервером і клієнтом, а також для зберігання і структурування даних. JSON базується на JavaScript, але він є незалежним від мови форматом, що робить його популярним для використання з різними мовами програмування[7].

JSON має простий синтаксис, який включає у себе пари ключ-значення,

розділені комами, і об'єкти, що представляють дані в вигляді набору вкладених об'єктів і масивів.

```
1  "products":[{
2      "id": 1,
3      "favorite": true,
4      "url": "product.html",
5      "image": "img/catalog/sweatshirt.webp",
6      "title": "Black Sweatshirt with ...",
7      "label": "Jhanvi's Brand",
8      "price": "$123.00"
9  },
```

Рисунок 2.5 – Приклад JSON коду

Figma – це вебсервіс та програмне забезпечення для дизайну і прототипування інтерфейсів користувача, який дозволяє дизайнерам, розробникам та іншим учасникам команди спільно працювати над проектом у реальному часі. Figma надає зручний інтерфейс для створення макетів, векторних графічних елементів, прототипів і навіть масштабується до створення цілих дизайн-систем[8].

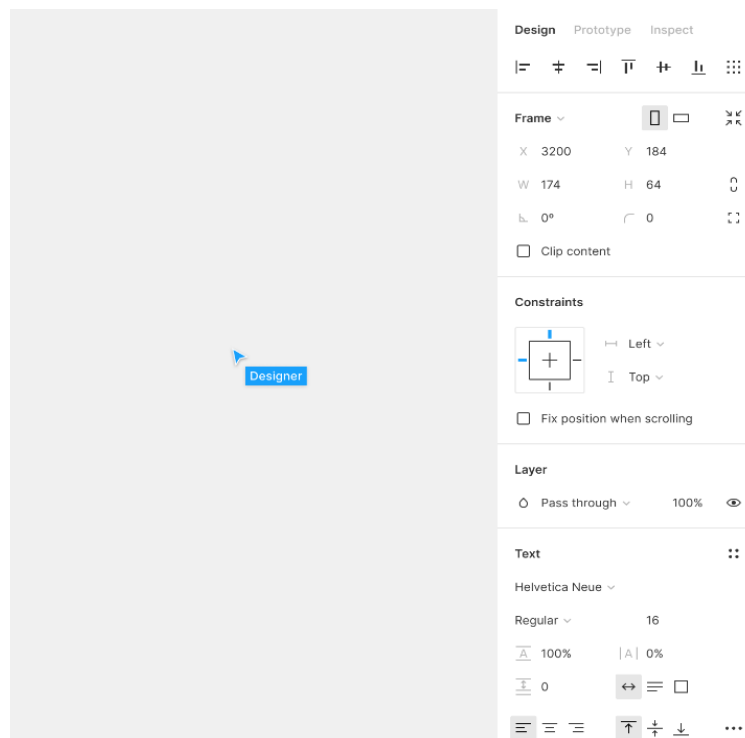


Рисунок 2.6 – Середовище Figma

2.2. Вибір системи керування версіями проєкту

Вибір системи керування версіями є ключовим етапом у розробці програмного забезпечення, оскільки він визначає, як розробники будуть працювати з кодом. На сьогоднішній день доступні різні системи керування версіями, такі як Git, Subversion, Mercurial та інші, кожна з яких має свої переваги та особливості. При виборі системи варто враховувати потреби команди розробників, характер проєкту, його масштаб та складність, а також можливості інтеграції з іншими інструментами розробки. Також важливо оцінити можливості системи щодо відгалуження, об'єднання змін та вирішення конфліктів.

GitHub є однією з найпопулярніших, безкоштовних та широко використовуваних платформ для спільної роботи над проєктами, використовуючи систему керування версіями Git. Вибір GitHub для керування версіями проєкту дозволяє команді розробників зручно співпрацювати, вносити зміни та відстежувати їх історію, а також використовувати різноманітні інструменти для управління проєктом. Зокрема, GitHub забезпечує можливість створення відгалужень (branches), подання запитів на об'єднання змін (pull requests) та вирішення конфліктів, що робить його оптимальним вибором для багатьох команд розробників[9].

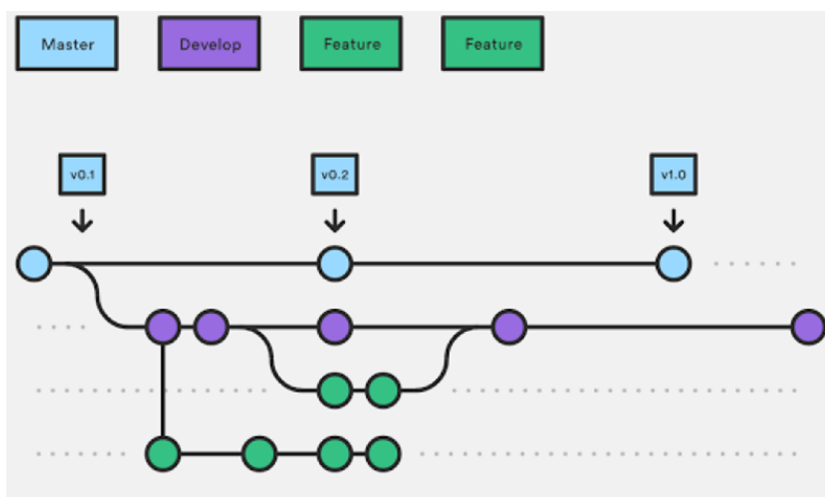


Рисунок 2.7 – Приклад роботи системи керуванням версіями Git

Як можна побачити з рисунка 2.7, основна гілка (часто називається "master" або "main") містить останню робочу версію проєкту. Гілка Developer призначена для створення нових можливостей та удосконалень. Всі заплановані зміни, що будуть включені у майбутні версії, розробляються саме тут. Коли хтось вносить зміни, вони спочатку створюють окрему гілку, а не одразу ж додають до основного стовбура. Тільки після того, як інші розробники перевірили, що зміни працюють правильно і не конфліктують з іншим кодом, гілку можна об'єднати з основним стовбуром. Гілки можна створювати паралельно, по черзі їх об'єднувати та порівнювати. Нові рядки коду в гілці будуть виділені іншим кольором у файлі, щоб візуально показати відмінності від основної версії стовбура. Тому використання цієї платформи для керування версіями проєкту є розумним та ефективним рішенням.

2.3. Розробка структури сайту

Структура вебсайту – це організація та ієрархія сторінок, що складаються з контенту, функцій та навігаційних елементів. Вона визначає, як різні сторінки та компоненти взаємодіють між собою та як користувачі можуть здійснювати переходи по сайту. Основна мета структури сайту – забезпечити зручну та логічну навігацію для користувачів. Ось кілька кроків, які можна виконати при розробці структури сайту:

- аналіз цілей аудиторії для розуміння потреб та очікувань цільової аудиторії, що допоможе створити структуру, яка відповідає їхнім потребам;
- створення карт сайту – це документ, який відображає ієрархію всіх сторінок сайту. Карта сайту допомагає визначити, які сторінки потрібно створити та як вони будуть пов'язані між собою;
- встановлення логічної ієрархії між різними сторінками сайту. Наприклад, які сторінки будуть головними, а які – підсторінками;
- завершення планування контенту для кожної сторінки. Це допоможе підготувати матеріали, які відповідають цілям кожної конкретної сторінки.



Рисунок 2.7 – Структура інтернет-магазину

На цьому етапі проектування є завершеним і можна переходити до практичної реалізації.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1. Створення дизайну сайту

Дизайн сайту – це не лише збірка графічних елементів, шрифтів та кольорів, яка має створювати гармонійну і виразну атмосферу для відвідувачів. Головна мета дизайну полягає в об'єднанні різних інформаційних елементів на сайті і в створенні такого візуального середовища, яке залишить приємне враження та забезпечить зручну навігацію. Розробка дизайну сайту відбувається у спеціалізованих графічних редакторах, що дозволяє налаштувати кожний деталь: від тіней та кольорів до точних розмірів, забезпечуючи таким чином максимальну гнучкість та індивідуальний підхід до кожного проекту.

У процесі створення дизайну сайту ми приділяли особливу увагу ряду ключових аспектів. Починаючи з аналізу цільової аудиторії, ми визначили їхні потреби та вподобання, щоб забезпечити їм максимальний комфорт взаємодії з сайтом. Враховуючи останні тенденції веб-дизайну, ми розробили естетично привабливий та функціональний інтерфейс, що забезпечує зручну навігацію для користувачів будь-якого рівня технічної обізнаності. Наш підхід до дизайну ґрунтується на принципах візуальної привабливості, легкості використання та відповідності бренду. Кожен елемент сайту був уважно пророблений з метою підвищення конверсії та забезпечення ефективної комунікації зі споживачами. В результаті ми здійснили успішне злиття естетики та функціональності, створивши дизайн, який відображає унікальний стиль.

Сайт матиме вісім сторінок, а саме: «Головна», «Каталог», «Сторінка товару», «Кошик», «Реєстрація», «Авторизація», «Особистий кабінет» та «Сторінка оформлення замовлення».

Результат створення дизайну головної сторінки ми можемо побачити на рисунках 3.1(а-г).

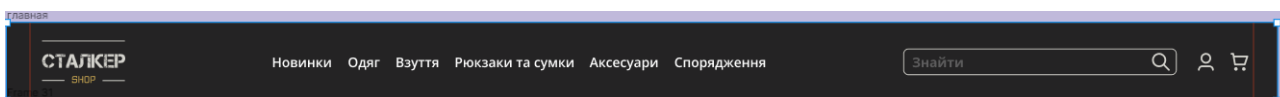


Рисунок 3.1а – Дизайн навігаційного меню

Як можна побачити на рисунку 3.1а, було розроблено меню не лише з урахуванням зручності та легкості навігації, а й з акцентом на ефективне використання простору та інтуїтивно зрозумілу структуру. Це дозволяє користувачам не лише швидко знаходити потрібну інформацію, а й швидко орієнтуватися в системі в цілому. Такий підхід сприяє покращенню загального досвіду користувача та зменшенню часу, необхідного для досягнення їх цілей. Крім того, було додано можливість пошуку, а також кнопки, які ведуть у особистий кабінет та в кошик, для ще більшого зручного використання.

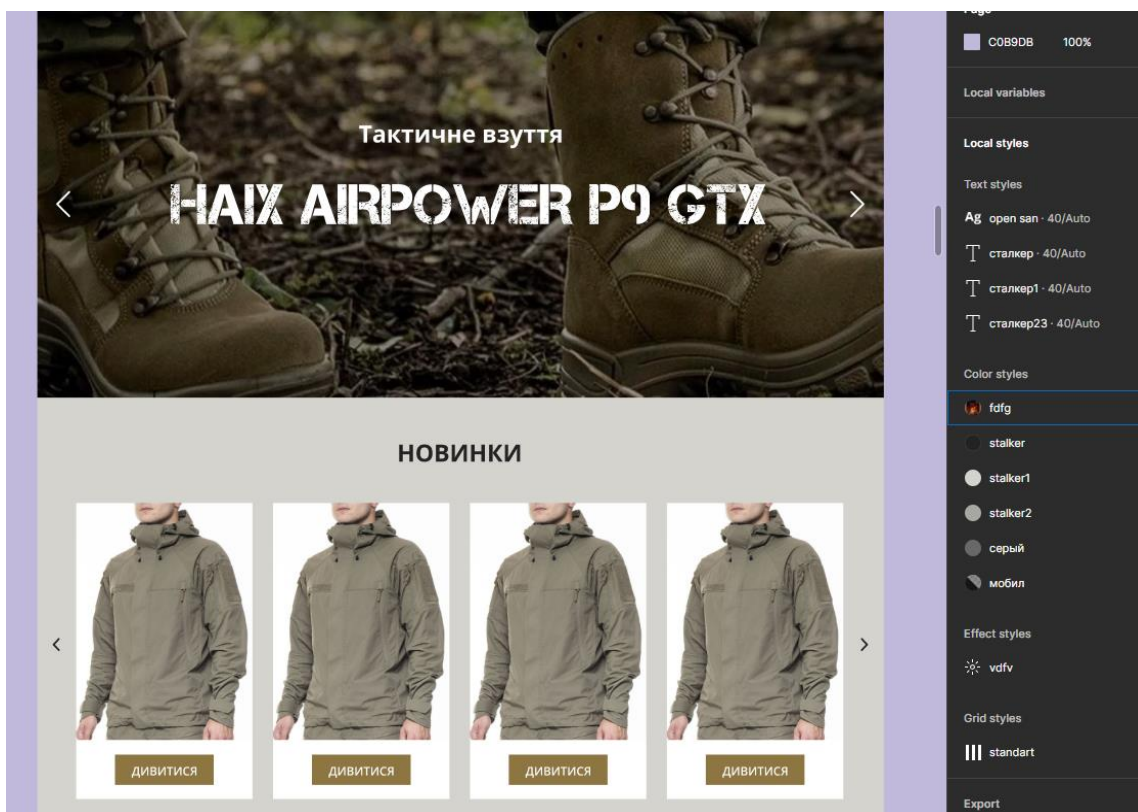


Рисунок 3.1б – Дизайн головної секції та секції «Новинки»

Дизайн ретельно розроблявся з урахуванням кількох ключових аспектів, включаючи головний екран та секцію "Новинки". Головна секція виконана у

формі слайдера, що дозволяє динамічно відображати ключові акції, нові товари або важливі повідомлення для відвідувачів. Секція "Новинки" виділена окремо, щоб користувачі могли легко ознайомитися з останніми поступленнями товарів та інформацією про їхні новинки. Всі ці елементи дизайну взаємодіють між собою, створюючи зручне та привабливе візуальне середовище для користувачів сайту.

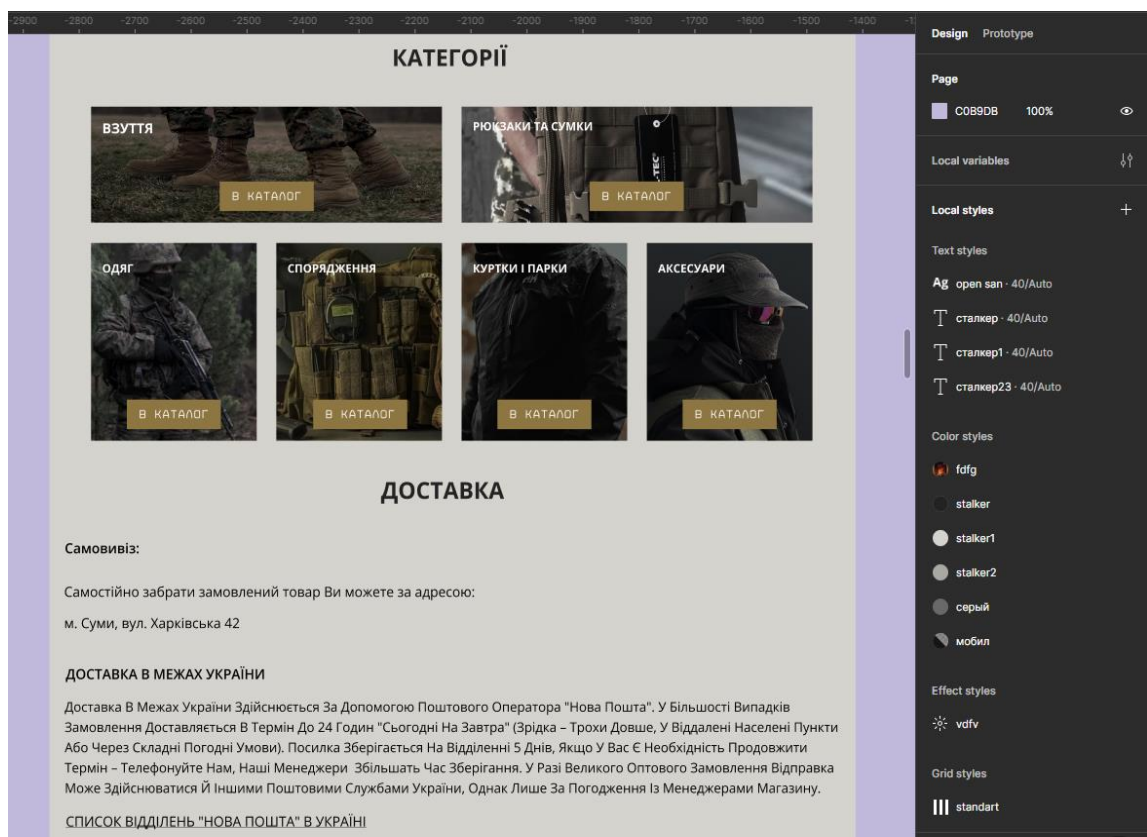


Рисунок 3.1в – Дизайн секцій «Категорії» та «Доставки»

На зображенні видно дві основні секції: «Категорії» та «Доставка». Кожна з цих секцій відображає важливі аспекти функціональності сайту та надає користувачам швидкий доступ до необхідної інформації. Секція «Категорії» дозволяє швидко переходити між різними категоріями товарів або послуг, тим самим полегшуючи навігацію та пошук потрібного продукту. У свою чергу, секція «Доставка» надає важливу інформацію про умови та варіанти доставки. Такий розподіл секцій сприяє зручності використання сайту та покращує загальний досвід користувача.

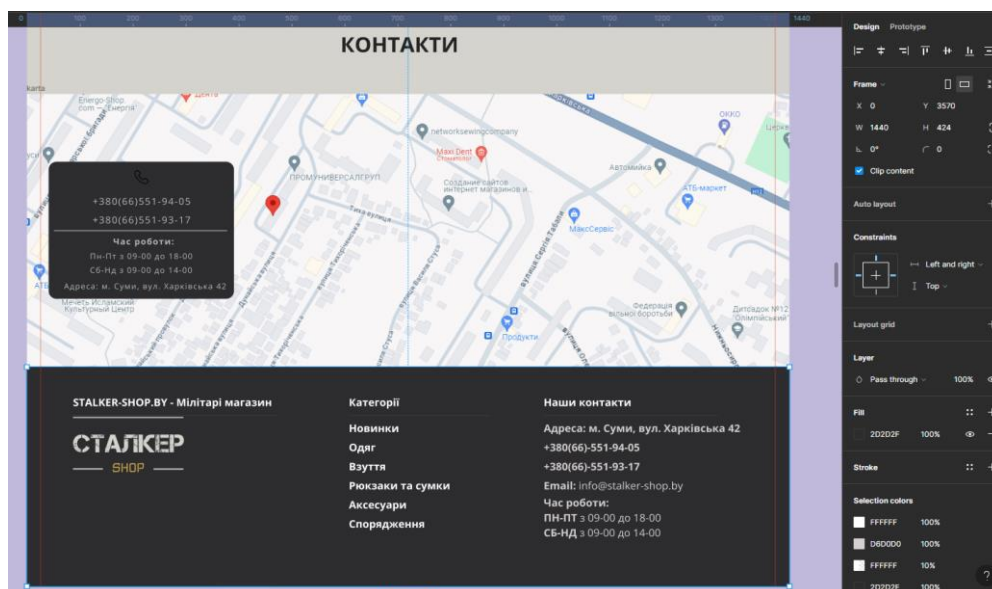


Рисунок 3.1г – Дизайн секцій «Контакти» та «Футеру»

Останнє зображення відображає дизайн секції «Контакти», де представлена вся необхідна інформація: адреса нашого магазину, години роботи та наш контактний номер телефону. У підвалі сторінки ми додатково розмістили наш логотип, розділи меню та контактні дані, щоб забезпечити легкий доступ до них для користувачів на будь-якій сторінці сайту. Дизайн сторінки каталог можна буде побачити на рисунка 3.2(а-б):

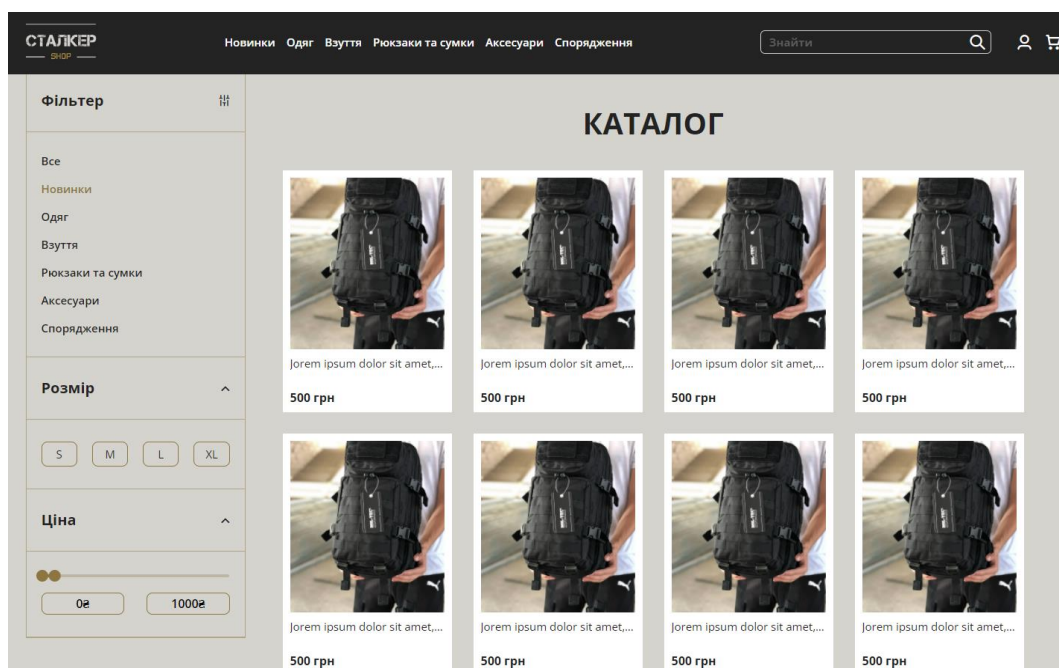


Рисунок 3.2а – Дизайн сторінки каталог та фільтру

На зображенні 3.2а прекрасно видно, що були створені прості карточки товарів, які легко розпізнати. На додаток до цього, впроваджено функцію фільтрації, що робить процес вибору товару ще більш зручним та швидким. За допомогою цього фільтра ми можемо впорядковувати товари за їхнім типом, ціною та розміром, надаючи користувачам можливість легко знаходити потрібний їм товар у всьому асортименті.

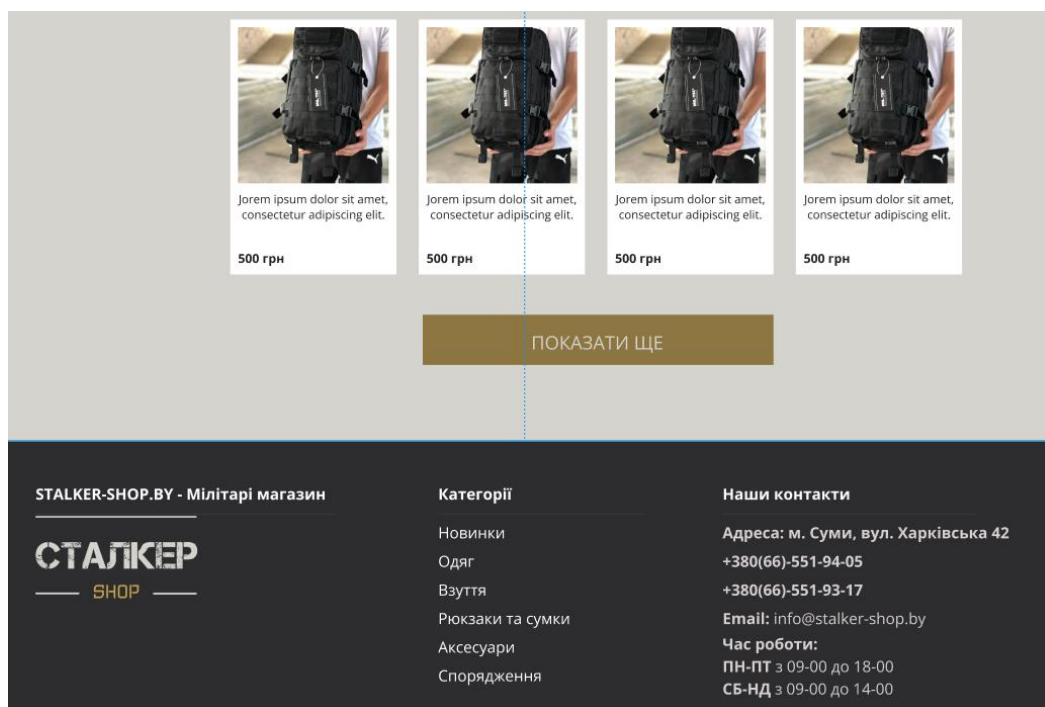


Рисунок 3.2б – Дизайн сторінки каталог та кнопки

На сторінку каталогу було додано елегантний та функціональний елемент – кнопку "Показати ще". Ця кнопка створена з метою забезпечення зручності користувачам. Просто натисніть на неї, і наступні карточки товарів з'являться перед вашими очима.

Крім того, було розроблено інші ключові сторінки сайту, які були описані раніше. Для створення дизайну використовувалась платформа Figma[11], яка не лише дозволяє створювати макети, але й надає широкий спектр інструментів для роботи з дизайном і прототипування. Використання цього інструменту дозволило нам ефективно та швидко втілити всі ідеї в життя, забезпечивши високу якість інтерфейсу для наших користувачів.

3.2. Підготовка до проєкту

Першим етапом підготовки до проєкту є завантаження середовища Node.js. Node.js дозволить нам використовувати потужний інструментарій, такий як Gulp, для автоматизації наших робочих процесів та створення зручної системи збірки. Для цього переходимо на офіційний сайт Node.js [рисунок 3.3](#) та натискаємо на кнопку «Download Node.js»

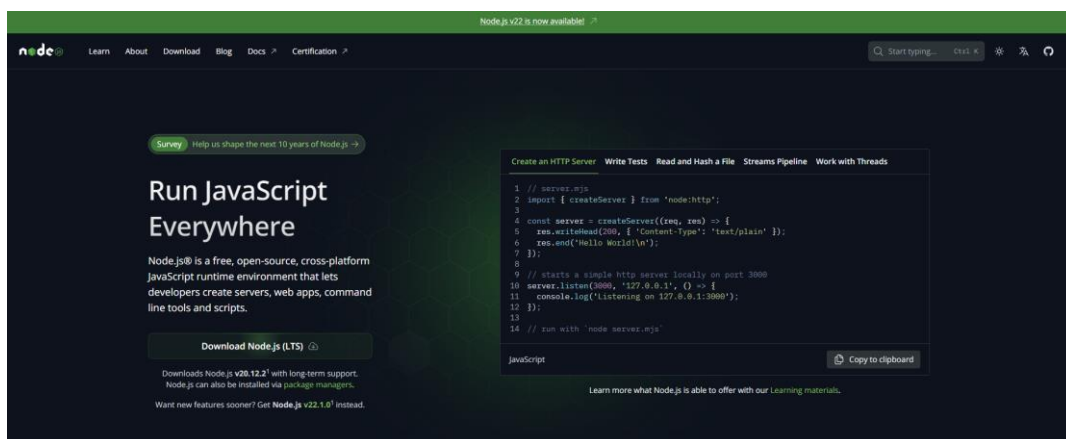


Рисунок 3.3 – Офіційний сайт Node.js

Після завантаження відкриваємо інсталятор Node.js та слідуємо інструкціям на екрані для встановлення програми на наш комп'ютер. Після того як встановили Node.js ми зможемо ефективно організувати наші задачі та оптимізувати робочий процес за допомогою Gulp.

Другий крок це встановлення Gulp глобально. Прописуємо команду `npm install -g gulp-cli` в терміналі редактору коду, щоб встановити Gulp глобально на нашому комп'ютері. Це дозволить нам використовувати команду `gulp` в будь-якій директорії. Як тільки встановлено Gulp глобально переходимо до встановлення його локально. Виконуємо команду `npm install gulp --save-dev`, щоб встановити Gulp як залежність нашого проєкту. Опція `--save-dev` додасть Gulp до нашого `package.json`.

Третій крок це після успішного встановлення Node.js та Gulp, встановлення розширення пакету Gulp за допомогою `npm` менеджера пакетів

Node.js. Наприклад, якщо потрібно встановлювати пакети або розширення для компіляції Sass, використовуємо команду `npm install gulp-sass --save-dev`. І так встановлюємо стільки розширень скільки треба для проєкту. Ці кроки допоможуть інтегрувати розширення у проєкт для подальшої автоматизації робочих процесів.

```

1 {
2   "name": "Stalker",
3   "version": "1.0.0",
4   "main": "gulpfile.js",
5   "scripts": {
6     "test": "echo \"Error: no test specified\" && exit 1"
7   },
8   "author": "",
9   "license": "ISC",
10  "devDependencies": {
11    "browser-sync": "^2.27.5",
12    "dart-sass": "^1.25.0",
13    "gulp": "^4.0.2",
14    "gulp-autoprefixer": "^6.1.0",
15    "gulp-clean-css": "^4.3.0",
16    "gulp-cli": "^2.3.0",
17    "gulp-htmlmin": "^5.0.1",
18    "gulp-imagemin": "^6.2.0",
19    "gulp-rename": "^1.4.0",
20    "gulp-sass": "^5.0.0",
21    "sass": "^1.43.2"
22  }

```

Рисунок 3.4 – Список пакетів Gulp для проєкту

Як тільки встановили всі розширення які були потрібні, переходимо до налаштування збірки Gulp. Це можна буде побачити на рисунку 3.5(а-б).

Перші рядки коду підключають необхідні модулі для Gulp та його розширення, такі як `gulp`, `browser-sync`, `gulp-sass`, `gulp-clean-css`, `gulp-autoprefixer`, `gulp-rename`, `gulp-imagemin`, та `gulp-htmlmin`. Задача `'server'` запускає локальний сервер за допомогою `BrowserSync`, який спрощує розробку вебсайтів. Сервер встановлюється в папці `dist`, і при зміні файлів HTML, сторінка автоматично перезавантажується. Задача `'styles'` компілює файли Sass у CSS, додає вендорні префікси, мініфікує та зберігає їх у папці `dist/css`. Після цього вона також оновлює сторінку у браузері завдяки `BrowserSync`. Задача `'watch'` стежить за змінами у файлах Sass, HTML, JavaScript, шрифтах, і зображеннях, викликаючи відповідні завдання Gulp при зміні.

```

1  const gulp      = require('gulp');
2  const browserSync = require('browser-sync');
3  const sass = require('gulp-sass')(require('sass'));
4  const cleanCSS = require('gulp-clean-css');
5  const autoprefixer = require('gulp-autoprefixer');
6  const rename = require("gulp-rename");
7  const imagemin = require('gulp-imagemin');
8  const htmlmin = require('gulp-htmlmin');
9
10 gulp.task('server', function() {
11
12     browserSync({
13         server: {
14             baseDir: "dist"
15         }
16     });
17
18     gulp.watch("src/*.html").on('change', browserSync.reload);
19 });
20
21 gulp.task('styles', function() {
22     return gulp.src("src/sass/**/*.+(scss|sass)")
23         .pipe(sass({ outputStyle: 'expanded' }).on('error', sass.logError))
24         .pipe(rename({suffix: '.min', prefix: ''}))
25         .pipe(autoprefixer())
26         .pipe(cleanCSS({compatibility: 'ie8'}))
27         .pipe(gulp.dest("dist/css"))
28         .pipe(browserSync.stream());
29 });
30
31 gulp.task('watch', function() {
32     gulp.watch("src/sass/**/*.+(scss|sass|css)", gulp.parallel('styles'));
33     gulp.watch("src/*.html").on('change', gulp.parallel('html'));
34     gulp.watch("src/js/**/*.js").on('change', gulp.parallel('scripts'));
35     gulp.watch("src/fonts/**/*").on('all', gulp.parallel('fonts'));
36     gulp.watch("src/icons/**/*").on('all', gulp.parallel('icons'));
37     gulp.watch("src/img/**/*").on('all', gulp.parallel('images'));
38 });

```

Рисунок 3.5а – Налаштування збірки Gulp

```

1  gulp.task('html', function () {
2      return gulp.src("src/*.html")
3          .pipe(htmlmin({ collapseWhitespace: true }))
4          .pipe(gulp.dest("dist"));
5  });
6
7  gulp.task('scripts', function () {
8      return gulp.src("src/js/**/*.js")
9          .pipe(gulp.dest("dist/js"))
10         .pipe(browserSync.stream());
11 });
12
13 gulp.task('fonts', function () {
14     return gulp.src("src/fonts/**/*")
15         .pipe(gulp.dest("dist/fonts"))
16         .pipe(browserSync.stream());
17 });
18
19 gulp.task('icons', function () {
20     return gulp.src("src/icons/**/*")
21         .pipe(gulp.dest("dist/icons"))
22         .pipe(browserSync.stream());
23 });
24
25 gulp.task('mailer', function () {
26     return gulp.src("src/mailer/**/*")
27         .pipe(gulp.dest("dist/mailer"));
28 });
29 gulp.task('json', function () {
30     return gulp.src("src/json/**/*")
31         .pipe(gulp.dest("dist/json"));
32 });
33
34 gulp.task('images', function () {
35     return gulp.src("src/img/**/*")
36         .pipe(imagemin())
37         .pipe(gulp.dest("dist/img"))
38         .pipe(browserSync.stream());
39 });
40
41 gulp.task('default', gulp.parallel('watch', 'server', 'styles', 'scripts', 'fonts', 'icons', 'mailer', 'json', 'html', 'images'));

```

Рисунок 3.5б – Налаштування збірки Gulp

На рисунку 3.5б можна побачити фрагменти коду на JavaScript з використанням Gulp, який визначає різні завдання для автоматизації робочих процесів у нашому проєкті. `html` це завдання оптимізує HTML-файли за допомогою плагіна `htmlmin` (мініфікація HTML) та переміщує їх у папку `dist`. `Scripts` це завдання просто копіює JavaScript-файли з папки `src/js` до папки `dist/js` та оновлює переглядач за допомогою `browserSync.stream()`. Такі самі дії виконуються з `fonts`, `icons`, `mailer`, `json`. `Images` це завдання оптимізує зображення за допомогою `imagemin` та переміщує їх у папку `dist/img`, оновлюючи при цьому переглядач за допомогою `browserSync.stream()`. `Default` це основне завдання, яке викликається за замовчуванням. Воно виконується одночасно з іншими завданнями, такими як `watch`, `server`, `styles`, `scripts`, `fonts`, `icons`, `mailer`, `json`, `html`, `images`. Ці завдання використовуються для автоматизації різних аспектів робочого процесу, таких як мініфікація HTML, оптимізація зображень, копіювання ресурсів тощо. Вони допомагають спростити розробку та забезпечити послідовність та якість роботи.

У середовищі Visual Studio Code, задаємо структуру папок для нашого проєкту (рис. 3.6). Розглянемо більш детально:

- тека `css` – містить в собі файли з стилями;
- тека `fonts` – зберігає всі шрифти які використовуються в проєкті;
- тека `icons` – зберігає іконки;
- тека `img` – зберігає всі необхідні фото;
- тека `js` – містить код для роботи з функціоналом сайту;
- тека `sass` – містить декілька папок і файлів для зручного задавання стилів;
- файл `index.html` – містить розмітку головної сторінки сайту;
- файл `_common.html` – містить компоненти, які найчастіше використовуються;
- всі інші файли пов'язані з Gulp;

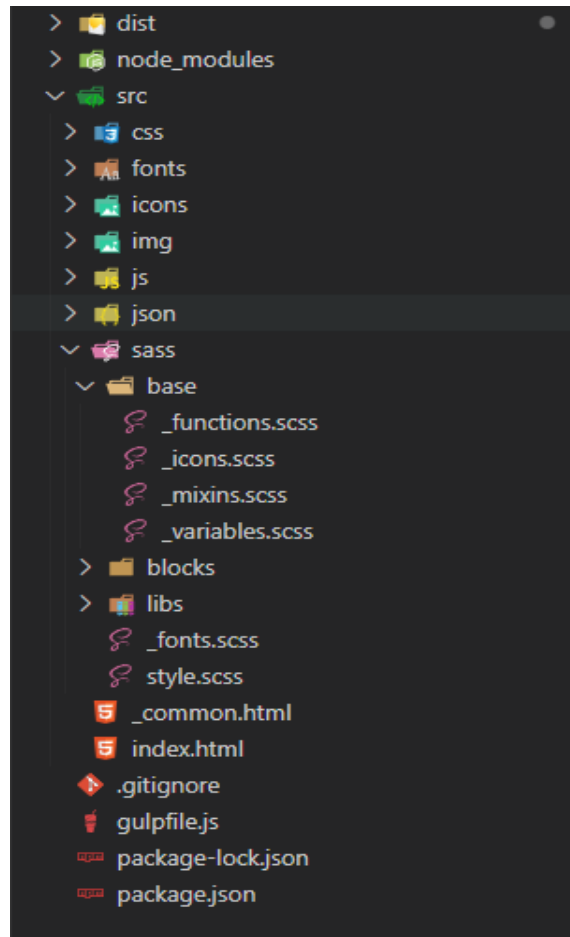


Рисунок 3.6 – Структура папок проєкту

3.3. Верстка інтернет-магазину

На основі розробленого дизайну ми можемо переходити до програмної розробки. За допомогою верстки розмістимо наші елементи дизайну згідно макету (рис.3.7).

Звідси ми бачимо, що була створено шапка та перша секція головної сторінки. У шапці містяться всі необхідні пункти меню та кнопки, які були в макеті, а також був доданий пошук по сайту. У першій секції екрану реалізовано слайдер за допомогою вже наявного плагіну Swiper[15]. Реалізацію головної секції слайдер можна буде побачити у додатку Б.

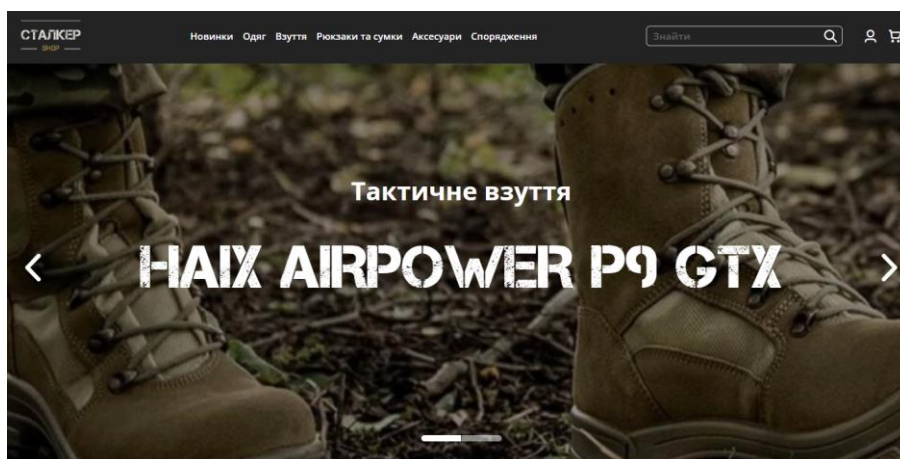


Рисунок 3.7 – Шапка та перша секція головної сторінки



Рисунок 3.8 – Секція новинки

Додано розділ "Новинки", щоб забезпечити користувачам можливість швидко оглядати останні поступлення товарів на сайті. Цей розділ також реалізований у формі слайдера, щоб забезпечити зручний інтерфейс прокрутки. Використано Swiper для створення цього слайдера, що дозволяє ефективно організувати та відображати нові товари для користувачів сайту.

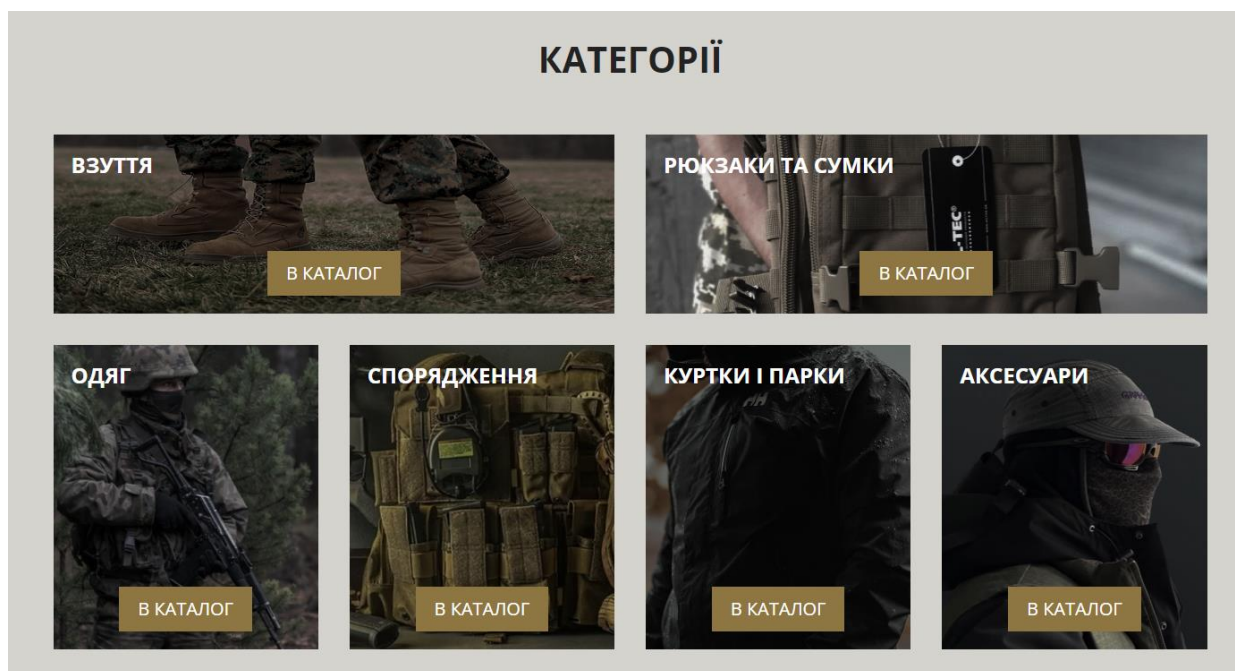


Рисунок 3.8 – Секція категорії

Було додано розділ під назвою «Категорії», щоб користувачі могли зручно переглядати асортимент магазину та швидко знаходити потрібні товари. Тепер вони можуть легко переходити між різними категоріями, щоб знайти саме те, що їм потрібно.

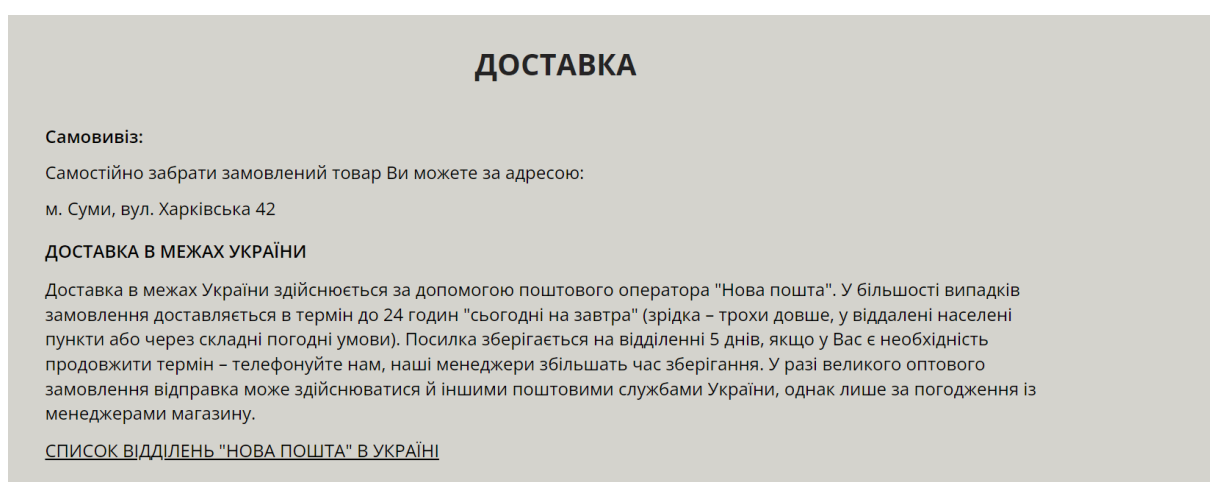


Рисунок 3.9 – Секція доставка

Розділ «Доставка» створено з метою надання користувачеві інформації про правила доставки та області, куди проводиться доставка. Також у розділ було

включено посилання на всі відділення Нової пошти, щоб користувач міг легко знайти найближче до себе.

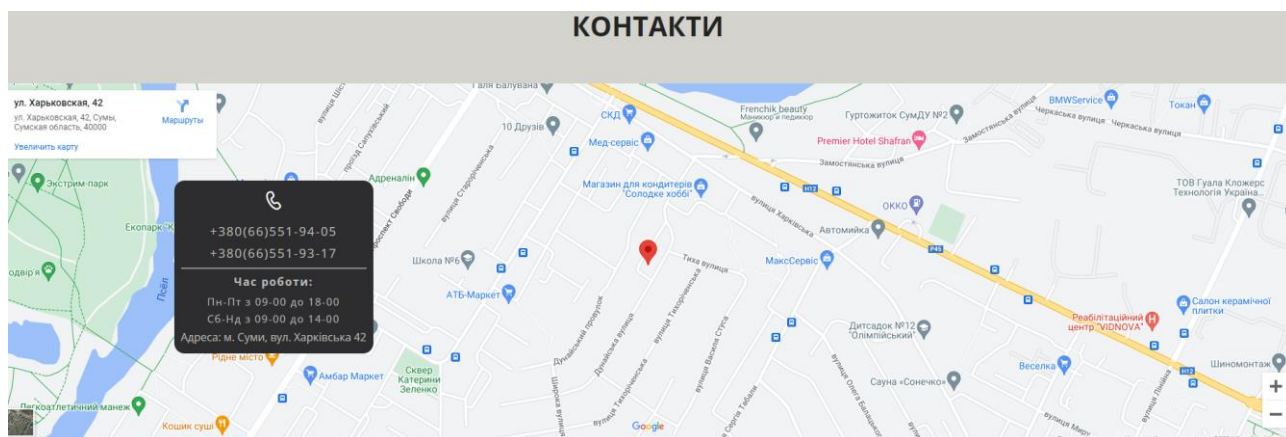


Рисунок 3.10 – Секція контакти

Зворотний зв'язок розділу «Контакти» призначено для зручності користувачів, щоб вони могли швидко знайти адресу магазину, графік його роботи та контактний номер телефону для зв'язку з ним.

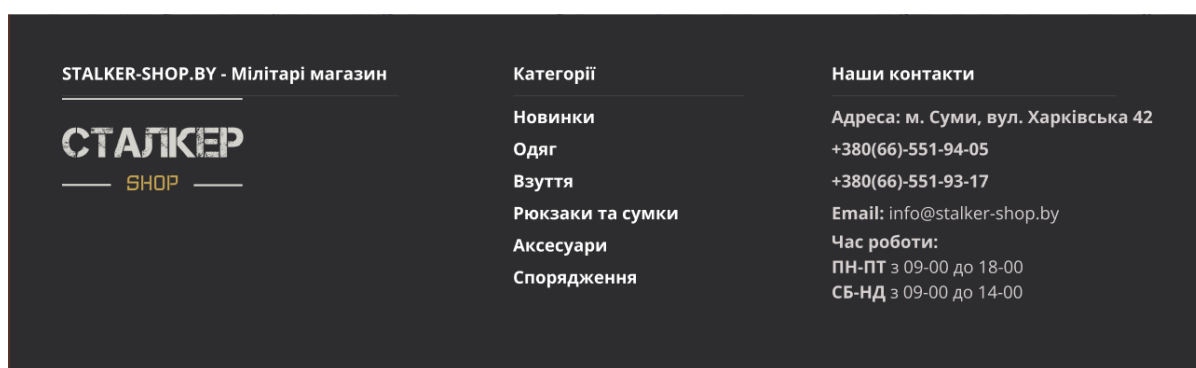
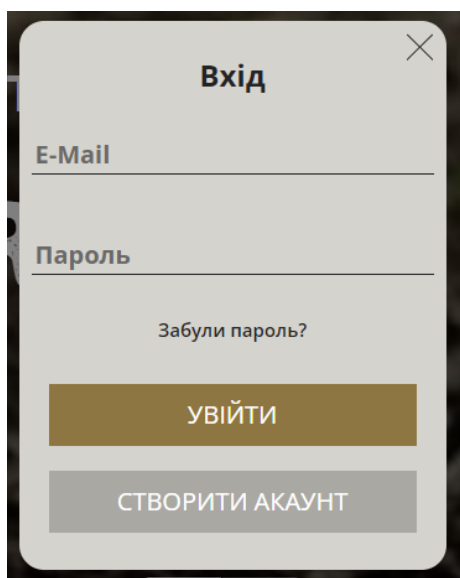


Рисунок 3.11 – Підвал сайту

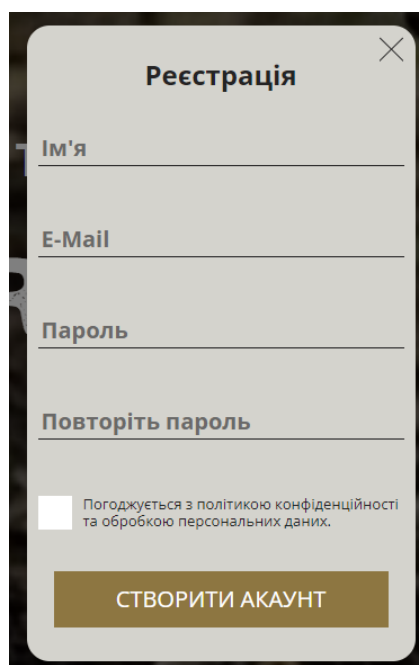
Підвал сайту був створений для подвоєння наявних категорій і забезпечення зручного способу знаходження магазину. Отже, таким чином було розроблено головну сторінку відповідно до створеного макету.

На рисунку 3.12 представлена форма для авторизації, що містить поля для введення електронної пошти та пароля. Також присутні кнопки для входу, створення акаунту і посилання для відновлення пароля у випадку його втрати.



The screenshot shows a mobile application window titled "Вхід" (Login). It features a close button (X) in the top right corner. Below the title, there are two input fields: "E-Mail" and "Пароль" (Password). A link "Забули пароль?" (Forgot password?) is positioned below the password field. At the bottom, there are two buttons: a prominent brown button labeled "УВІЙТИ" (Login) and a grey button labeled "СТВОРИТИ АКАУНТ" (Create Account).

Рисунок 3.12 – Модульне вікно авторизації



The screenshot shows a mobile application window titled "Реєстрація" (Registration). It features a close button (X) in the top right corner. Below the title, there are four input fields: "Ім'я" (Name), "E-Mail", "Пароль" (Password), and "Повторіть пароль" (Repeat password). Below the "Пароль" field, there is a checkbox with the text "Погоджуються з політикою конфіденційності та обробкою персональних даних." (I agree with the privacy policy and processing of personal data.). At the bottom, there is a brown button labeled "СТВОРИТИ АКАУНТ" (Create Account).

Рисунок 3.13 – Модульне вікно реєстрації

Як видно на рисунку 3.13, форма реєстрації включає поля для введення імені, електронної пошти, паролю та підтвердження паролю, а також радіокнопку для згоди з політикою конфіденційності. Наступна сторінка для розробки це «Каталог»:

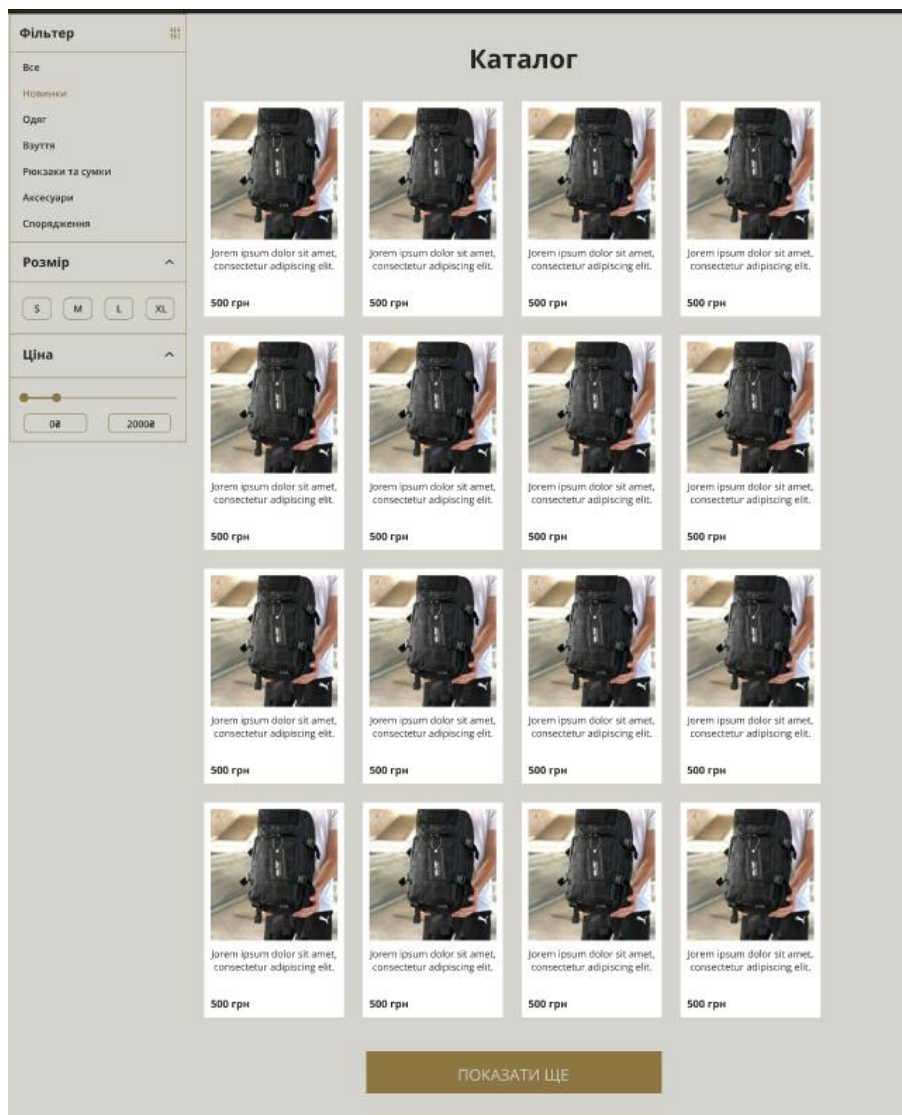


Рисунок 3.14 – Сторінка «Каталог»

Як видно, сторінка каталогу була зверстана відповідно до макету з додаванням функціоналу кнопки "Показати ще", фільтром та картками товарів, щоб усі заплановані дії виконувалися без помилок.

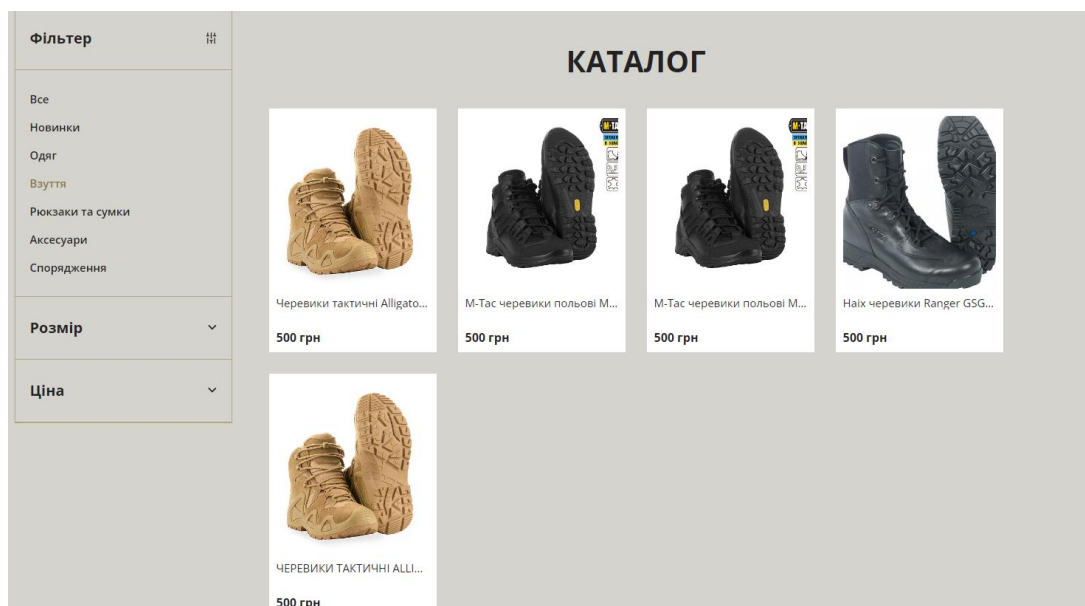


Рисунок 3.15 – Результат роботи фільтру

На рисунку 3.15 показано результат роботи фільтра для вибраної категорії. Як видно, коли користувач обирає певну категорію фільтра, відображається відповідна товарна картка, яку обрав користувач. Сторінку товару можна побачити на рисунку 3.16.

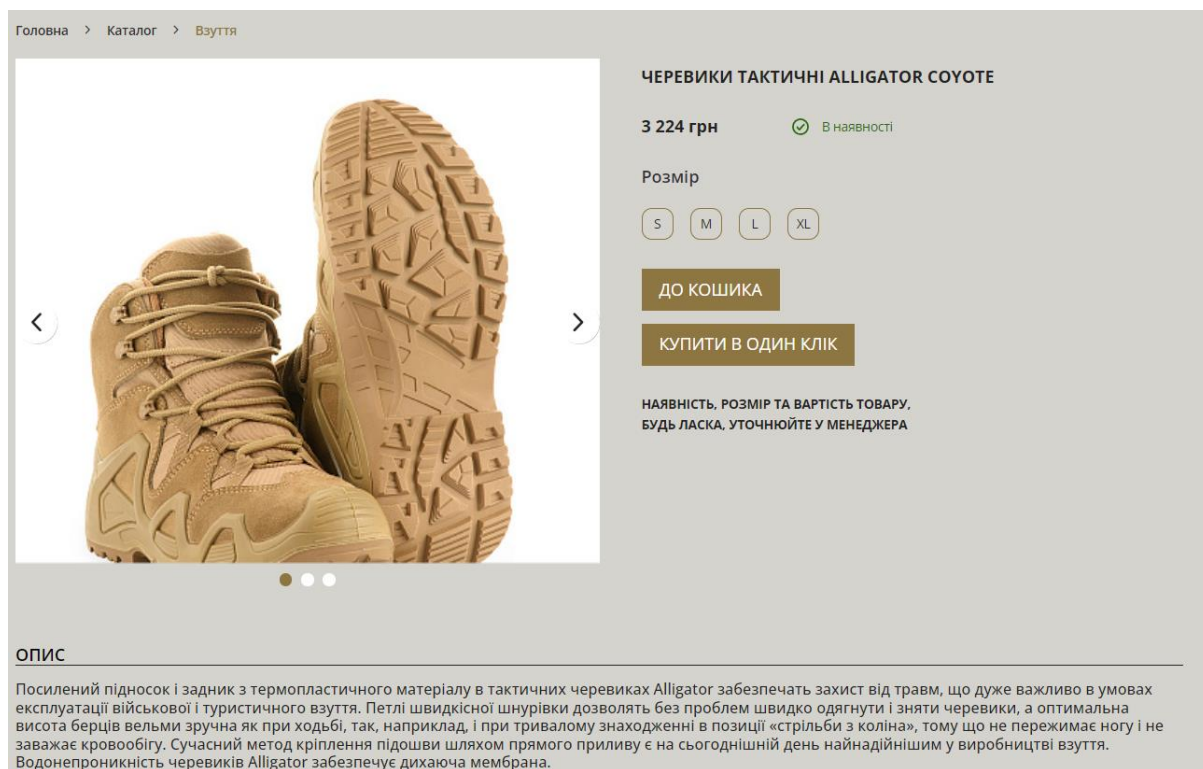


Рисунок 3.16 – Сторінка товару

На сторінці товару представлені кнопки «До кошика» та «Купити в один клік». Можна обрати необхідний розмір, побачити ціну товару, його опис і дізнатися про наявність. Також при телефонному виклику можна дізнатися розмір і наявність товару. На рисунку 3.17 можна побачити як виглядає кошик:

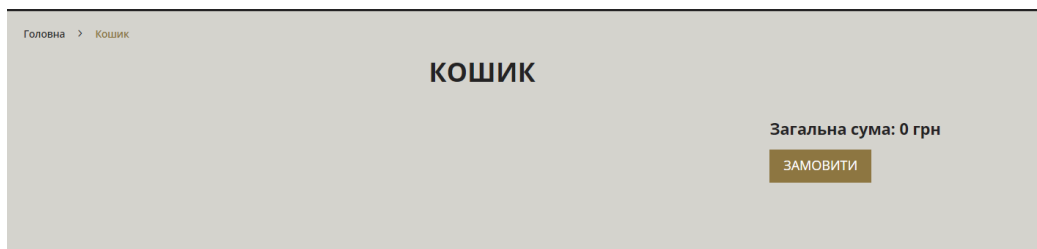


Рисунок 3.17 – Кошик

Коли кошик порожній, у ньому відображається лише кнопка «Замовити» та загальна сума. Заповнений кошик можна буде побачити на рисунку 3.18. Розмітку сторінки можна побачити у додатку А, а реалізацію як додається товар до кошика у додатку Б:

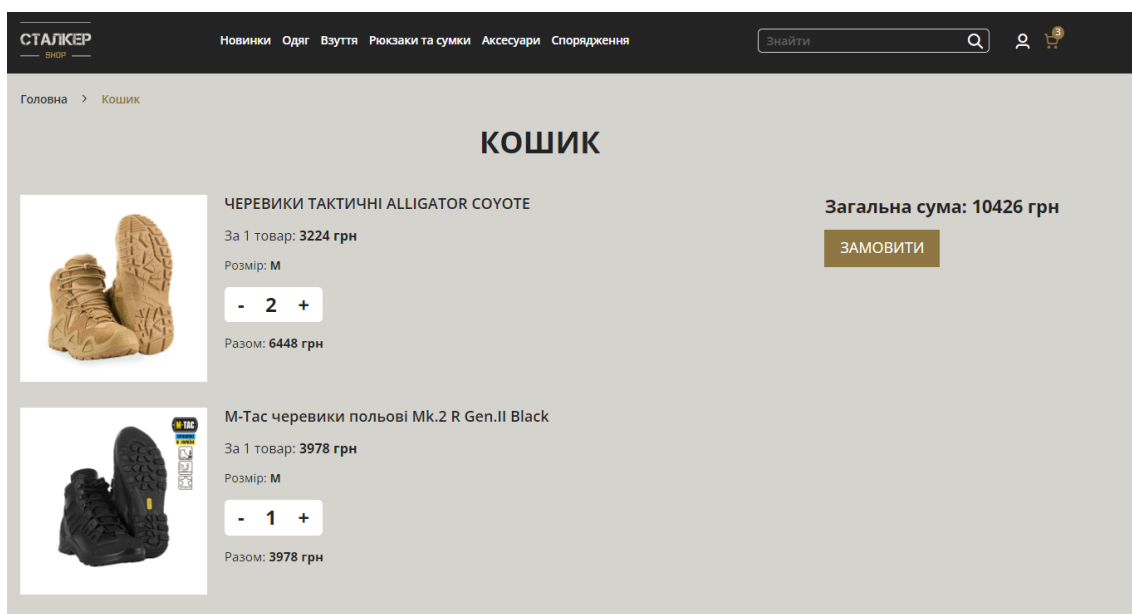


Рисунок 3.18 – Заповнений кошик

Згідно з рисунком 3.18, при додаванні товару до кошика з'являється назва товару, його зображення, ціна за одиницю, кількість товару (яку можна

збільшувати або зменшувати), вартість за обрану кількість товарів, а також загальна сума за всі товари, які знаходяться в кошику. Також можна побачити в шапці сайту кількість товарів в кошику. На рисунку 3.19 можна побачити сторінку «Мої дані».

The screenshot shows the 'Мої дані' (My Data) page on the STALKER website. The page has a dark header with the logo and navigation links. Below the header, there is a breadcrumb trail: 'Головна > Аккаунт > Мої дані'. The main content area is divided into two columns. The left column is titled 'МОЇ ДАНІ' and contains a 'Замовлення' (Orders) section with a 'Вихід' (Exit) link. Below this are five input fields: 'Ім'я' (Name), 'Прізвище' (Surname), 'По батькові' (Patronymic), '+380' (Phone prefix), and 'E-Mail'. A 'ЗБЕРЕГТИ' (SAVE) button is at the bottom of this section. The right column is titled 'АДРЕСА ДОСТАВКИ' (Delivery Address) and contains a '+ДОДАТИ АДРЕСУ' (+ADD ADDRESS) button. Below this is the 'ІНФОРМАЦІЯ ПРО ДОСТАВКУ' (DELIVERY INFORMATION) section with three radio button options: 'Самовивіз товару' (Self-pickup), 'Кур'єром' (Courier, selected), and 'Нова Пошта' (Nova Poshta). The 'Кур'єром' option has a sub-note: 'Доставка здійснюється при замовленні товару на суму понад 1500 грн.' Below this are two more radio button options: 'Укр Пошта' (Ukrainian Post) and 'Нова Пошта' (Nova Poshta). The 'Укр Пошта' option has a sub-note: 'Вартість доставки 50 грн'. The 'Нова Пошта' option has a sub-note: 'Вартість доставки 70 грн'. At the bottom of the right column is the 'СПОСІБ ОПЛАТИ' (PAYMENT METHOD) section with two radio button options: 'Готівкою' (Cash, selected) and 'Накладний платіж' (Post payment). The 'Готівкою' option has a sub-note: 'Оплата готівкою кур'єру'. The 'Накладний платіж' option has a sub-note: 'Оплата здійснюється після отримання товару у відділенні пошти'.

Рисунок 3.19 – Сторінка «Мої дані»

На цій сторінці ви можете вказати свої дані, зокрема ім'я, прізвище, по батькові, номер телефону та електронну пошту. Також тут можна додати інформацію про доставку: свою адресу, спосіб доставки товару і метод оплати, щоб користувачеві було зрозуміло, як буде здійснена доставка товару.

3.4. Перевірка адаптивності

Адаптивність сторінок є критично важливою в сучасному вебсередовищі. З поширенням різноманітних пристроїв для доступу до Інтернету, від комп'ютерів до смартфонів та планшетів, створення вебсторінок, які ефективно працюють на будь-яких пристроях, стає необхідністю.

У перевірці використовувався інструмент "Панель розробника" у браузері Google Chrome. Ця панель дозволяє вибрати різні розміри екрана крім того, можна обирати горизонтальну або вертикальну орієнтацію пристрою. Для перевірки на адаптивність перевіримо тільки головну сторінку сайту під розмір 320px.

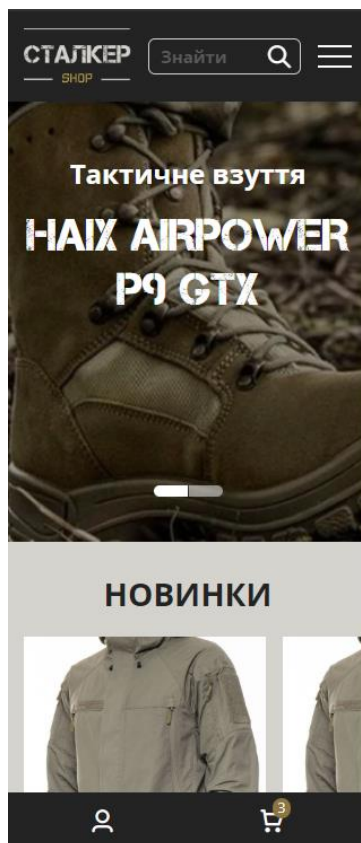
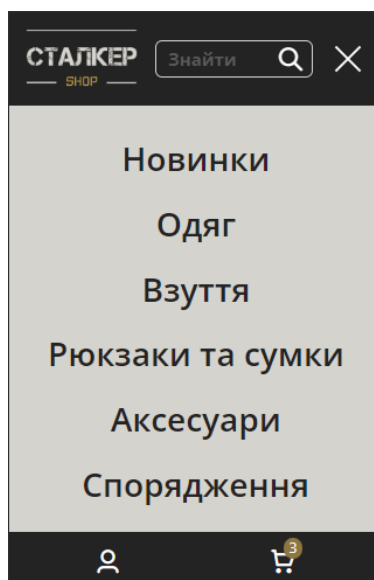


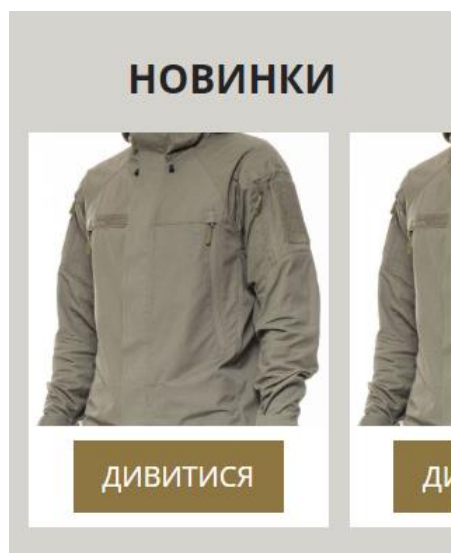
Рисунок 3.20 – Адаптивний головний екран та шапка

Пункти меню були приховані у бургер-меню, а кошик та обліковий запис переміщені вниз, щоб покращити зручність користування та візуальне сприйняття сайту, забезпечивши їх легшу доступність для користувача. На рисунку 3.21 можна побачити адаптивне меню для



Рисунку 3.21 – Адаптивне меню

Адаптивне меню було створено, щоб уникнути перекриття пунктів та невідавкового натискання на непотрібний пункт. Користувач може легко відкривати та закривати пункти для зручного використання.



Рисунку 3.22 – Адаптивна секція «Новинки»

Щоб полегшити використання слайдера та забезпечити зрозумілість, вирішено сховати стрілки через обмежений простір, проте залишили половину наступної карточки видимою, щоб користувач міг зрозуміти, що це слайдер і як його прокрутити. Це можна побачити у додатку Б.

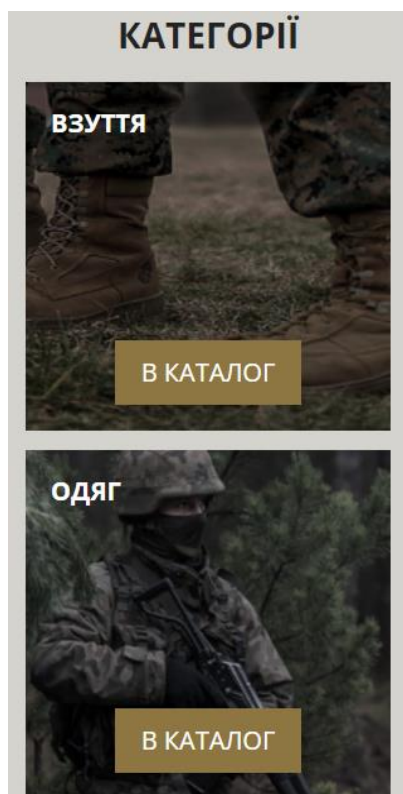


Рисунок 3.23 – Адаптивна секція «Категорії»

Як показано на рисунку 3.23, було вирішено перетворити цю секцію на стовпчики, щоб вони виглядали однаково і розташовувалися один під одним.

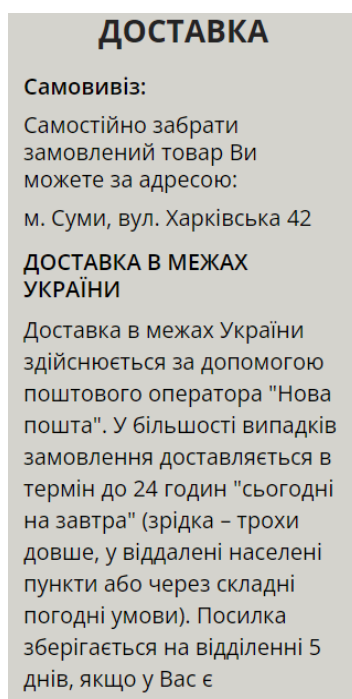


Рисунок 3.24 – Адаптивна секція «Доставка»

Секція залишилася майже незмінною, оскільки вона містить лише текст, який автоматично підлаштовується під розміри.

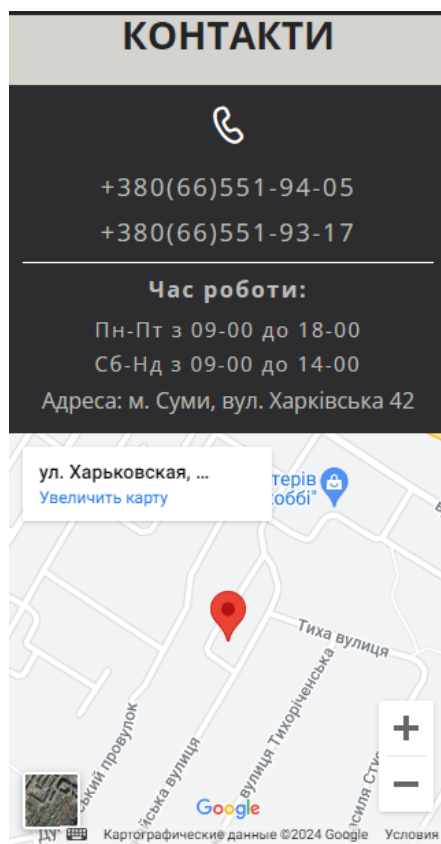


Рисунок 3.25 – Адаптивна секція «Контакти»

На зображенні 3.25 видно, що блок з контактними даними був переміщений вище карти. Раніше цей блок накладався на карту, але зміни були зроблені для того, щоб на маленькому екрані карта залишалася видимою і не перекривалася.

На рисунку 3.26 зображено адаптивний футер, який виконано у вигляді стовпчиків. Це рішення було прийнято для забезпечення естетичного вигляду на маленьких екранах.

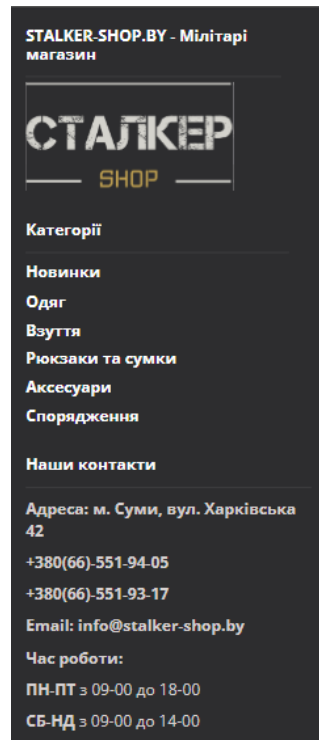


Рисунок 3.26 – Адаптивний футер сайту

Як можна побачити на зображеннях, було успішно розроблено адаптивну головну сторінку сайту і по такій тактиці буде розроблено інші адаптовані сторінки інтернет магазину.

Увесь код розробленого інтернет-магазину знаходиться за посиланням на GitHub <https://github.com/M1m1k23/Stalker>

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено адаптивний інтернет-магазин військового спорядження. Було розглянуто та обрано оптимальні інструменти для виконання поставленої мети. Також, під час розробки було визначено вимоги до продукту.

У ході виконання роботи було виконано завдання:

- Аналіз програмних засобів та інструментів.
- Опанування векторного онлайн-сервісу розробки інтерфейсів та прототипування Figma.
- Створення дизайну.
- Опанування обраних технологій для розроблення сайту.
- Розробка сайту.
- Протестовано адаптивності дизайну під різні пристрої.

Можна зробити висновок, що створений інтернет-магазин задовольняє потреби користувачів, надаючи унікальний користувацький досвід, це важливо. Забезпечуючи адаптивність сторінок під різні пристрої, магазин дозволяє користувачам зручно переглядати та придбати товари навіть на мобільних пристроях. Все це сприяє позитивному користувацькому досвіду та підвищує ймовірність повторного відвідування магазину і здійснення покупок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Учасники проєктів Вікімедіа. Visual Studio Code – Вікіпедія. Вікіпедія.
URL: https://uk.wikipedia.org/wiki/Visual_Studio_Code#:~:text=Visual%20Studio%20Code%20–%20засіб%20для,Windows,%20Linux%20і%20OS%20X.&text=Компанія%20Microsoft%20представила%20Visual%20Studio,2015%20на%20конференції%20Build%202015.
2. Учасники проєктів Вікімедіа. HTML – Вікіпедія. Вікіпедія.
URL: <https://uk.wikipedia.org/wiki/HTML>.
3. Що таке html?. Український веб-довідник.
URL: https://css.in.ua/article/shcho-take-css_3.
4. Що таке CSS. Український веб-довідник.
URL: https://css.in.ua/article/shcho-take-html_10.
5. JavaScript - Клієнтською мовою програмування, що робить сторінки сайту інтерактивними. Browser check, please wait ...
URL: <https://astwellsoft.com/uk/blog/tehnology/javascript.html>.
6. Що таке Sacc?. Web Developer in UA. URL: <https://web-developer.in.ua/assets/articles/sass/what-is-sass/what-is-sass.html>.
7. Що таке JSON | Навіщо потрібен цей формат. *Apix-Drive*.
URL: <https://apix-drive.com/ua/blog/useful/scho-take-json>
8. Омельчук Є. Що таке Figma: функції, інструменти та переваги - академія Wezom. URL: <https://wezom.academy/ua>
9. Що таке GitHub і як з ним працювати | Онлайн-курси від компанії QATestLab. *Онлайн-курси від компанії QATestLab | Головна сторінка*. URL: <https://training.qatestlab.com/blog/technical-articles/what-is-github-and-how-to-work/>
10. Етапи створення сайту| 9 етапів від ідеї до запуску | Блог WEBCASE. *Webcase*. URL: <https://webcase.com.ua>
11. Figma: The Collaborative Interface Design Tool. *Figma*. URL:

<https://www.figma.com/>

12. Збірка з Gulp. *Htmlbook* - довідник HTML і CSS. URL: <https://htmlbook.online/page/dev-building-with-gulp>

13. Node.js – Run JavaScript Everywhere. *Node.js – Run JavaScript Everywhere*. URL: <https://nodejs.org/en>

14. Quick Start | gulp.js. *gulp.js*. URL: <https://gulpjs.com/docs/en/getting-started/quick-start>

15. Swiper - The Most Modern Mobile Touch Slider. *Swiper*. URL: <https://swiperjs.com/>

16. Воєнторг. Армійський, військовий магазин Мілітарист у Києві, Харкові, Одесі, Дніпрі, Запоріжжі - *Militarist.ua*. *Militarist*. URL: <https://militarist.ua/ua/>

17. Воєнторг Київ - Армійський магазин військового одягу, Військовий магазин Мілітарі | *AGRESSOR*. *Воєнторг Київ - Армійський магазин військового одягу, Військовий магазин Мілітарі | AGRESSOR*. URL: <https://agressor.com.ua/ua>

Додаток А

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swiper@11/swiper-
bundle.min.css"/>
  <link rel="stylesheet" href="css/style.min.css">
  <title>Stalker || Cart</title>
</head>
<body>
  <div class="wrap">
    <header class="header">
      <div class="header__container">
        <a href="index.html" class="header__logo">
          
        </a>
        <div class="header__menu menu">
          <nav class="menu__body">
            <ul class="menu__list">
              <li class="menu__item">
                <a href="catalog.html"
class="menu__link">Новинки</a>
              </li>
              <li class="menu__item">
                <a href="#" class="menu__link">Одяг</a>
              </li>
              <li class="menu__item">
                <a href="#" class="menu__link">Взуття</a>
              </li>
              <li class="menu__item">
                <a href="#" class="menu__link">Рюкзаки та сумки</a>
              </li>
              <li class="menu__item">
                <a href="#" class="menu__link">Акcesуари</a>
              </li>
              <li class="menu__item">
                <a href="#" class="menu__link">Спорядження</a>
            </ul>
          </nav>
        </div>
      </div>
    </header>
  </div>

```



```

        </li>
    </ul>
</nav>
</div>
<form action="#" class="header__search search-form">
    <input type="text" class="search-form__input"
placeholder="Знайти">
    <button class="search-form__button _icon-search"></button>
</form>
<div class="header__action action-header">
    <a href="#" data-popup="#popup" class="action-
header__item _icon-user"></a>
    <a href="cart.html" class="action-header__item action-
header__item--active _icon-cart">
        <span class="action-header__quantity">0</span>
    </a>
</div>
<button class="icon-menu"><span></span></button>
</div>
</header>
<main class="page">
    <div class="page__cart cart">
        <section class="cart">
            <div class="cart__container">
                <ul class="cart__breadcrumbs breadcrumbs">
                    <li class="breadcrumbs__item _icon-ch-right">
                        <a href="index.html"
class="breadcrumbs__link">Головна</a>
                    </li>
                    <li class="breadcrumbs__item">
                        <span class="breadcrumbs__current">Кошик</span>
                    </li>
                </ul>
                <h1 class="cart__title title">Кошик</h1>
                <div class="cart__wrapper">
                    <ul class="cart__list">
                    </ul>
                    <div class="cart__total">
                        <div class="cart__total"></div>

```

```

        <button class="cart__button
button">Замовити</button>
    </div>
</div>
</div>
</section>
</div>
</main>
<footer class="footer">
    <div class="footer__container">
        <div class="footer__body">
            <h4 class="footer__title">STALKER-SHOP.BY - Мілітарі
магазин</h4>
            <a href="#" class="footer__logo-big">
                
            </a>
        </div>
        <div class="footer__menu menu-footer">
            <div class="menu-footer__block">
                <h4 class="menu-footer__title">Категорії</h4>
                <ul class="menu-footer__list">
                    <li class="menu-footer__item">
                        <a href="#" class="menu-footer__link">Новинки</a>
                    </li>
                    <li class="menu-footer__item">
                        <a href="#" class="menu-footer__link">Одяг</a>
                    </li>
                    <li class="menu-footer__item">
                        <a href="#" class="menu-footer__link">Взуття</a>
                    </li>
                    <li class="menu-footer__item">
                        <a href="#" class="menu-footer__link">Рюкзаки та
сумки</a>
                    </li>
                    <li class="menu-footer__item">
                        <a href="#" class="menu-footer__link">Акcesуари</a>
                    </li>
                    <li class="menu-footer__item">
                        <a href="#" class="menu-

```

```

footer__link">Спорядження</a>
        </li>
    </ul>
</div>
</div>
<div class="footer__info info-footer">
    <h4 class="info-footer__title">Наши контакти</h4>
    <ul class="info-footer__list">
        <li class="info-footer__item">
            Адреса: м. Суми, вул. Харківська 42
        </li>
        <li class="info-footer__item">
            <a href="tel:+380(66)-551-94-05" class="info-
footer__link">+380(66)-551-94-05</a>
        </li>
        <li class="info-footer__item">
            <a href="tel:+380(66)-551-94-05" class="info-
footer__link">+380(66)-551-93-17</a>
        </li>
        <li class="info-footer__item">
            Email: <a href="mailto:info@stalker-shop.by"
class="info-footer__link">info@stalker-shop.by</a>
        </li>
        <li class="info-footer__item">Час роботи:</li>
        <li class="info-footer__item">ПН-ПТ <span>з 09-00 до 18-
00</span></li>
        <li class="info-footer__item">СБ-НД <span>з 09-00 до 14-
00</span></li>
    </ul>
</div>
</div>
</footer>
</div>
<script src="js/script.js"></script>
</body>
</html>

```

Додаток Б

```
function displayProduct(product) {
```

```

const descriptionElement = document.querySelector('.description-product__info
p');

// Заповнюємо дані на сторінці товару з отриманими даними
document.querySelector('.main-product__title').innerText = product.name;
document.querySelector('.main-product__image--one').src =
product.image_sliderOne;
document.querySelector('.main-product__image--two').src =
product.image_sliderTwo;
document.querySelector('.main-product__image--three').src =
product.image_sliderThree;
document.querySelector('.main-product__price').innerText = product.price + '
грн';
document.querySelector('.description-product__info p').innerText =
product.description;
// Додати подію для кнопки "До кошика"
document.querySelector('.action-
product__button.button').addEventListener('click', function() {
const selectedSize = document.querySelector('.sizes-product__input:checked');
if (selectedSize) {
const size = selectedSize.value;
addToCart(product, size);
} else {
alert('Будь ласка, виберіть розмір');
}
});
}
function addToCart(product, size) {
let cart = JSON.parse(localStorage.getItem('cart')) || [];
const existingProduct = cart.find(item => item.id === product.id && item.size
=== size);

if (existingProduct) {
existingProduct.quantity += 1;
} else {
const productToAdd = {...product, quantity: 1, size};
cart.push(productToAdd);
}
}

```

```

localStorage.setItem('cart', JSON.stringify(cart));
alert('Товар додано до кошика');
updateCartQuantityDisplay();
}
document.addEventListener('DOMContentLoaded', function() {
  if (window.location.pathname.endsWith('cart.html')) {
    displayCart();
  }
});
function displayCart() {
  const cart = JSON.parse(localStorage.getItem('cart')) || [];
  const cartList = document.querySelector('.cart__list');
  const cartTotal = document.querySelector('.cart__total');

  cartList.innerHTML = '';
  let total = 0;

  cart.forEach(product => {
    const productElement = document.createElement('li');
    productElement.className = 'cart__item';
    productElement.innerHTML = `
      <div class="cart__image"></div>
      <div class="cart__body">
        <div class="cart__name">${product.name}</div>
        <div class="cart__price">За 1 товар: <span> ${product.price}
грн</span></div>
        <div class="cart__size">Розмір: <span>${product.size}</span></div>

        <div class="cart__quantity">
          <button class="cart__quantity-btn" data-id="${product.id}" data-
size="${product.size}" data-action="decrease">-</button>
          <span class="cart__quantity-value">${product.quantity}</span>
          <button class="cart__quantity-btn" data-id="${product.id}" data-
size="${product.size}" data-action="increase">+</button>
        </div>
        <div class="cart__total-price">Разом: <span>${product.price *
product.quantity} грн </span></div>
      </div>

```

```

    `;
    cartList.appendChild(productElement);
    total += product.price * product.quantity;
  });

  cartTotal.innerHTML = `Загальна сума: ${total} грн`;

  // Додаємо обробники подій для кнопок збільшення/зменшення кількості
  document.querySelectorAll('.cart__quantity-btn').forEach(button => {
    button.addEventListener('click', function() {
      const productId = parseInt(this.dataset.id);
      const size = this.dataset.size;
      const action = this.dataset.action;
      updateCart(productId, size, action);
    });
  });
  updateCartQuantityDisplay();
}

function updateCart(productId, size, action) {
  let cart = JSON.parse(localStorage.getItem('cart')) || [];
  const product = cart.find(item => item.id === productId && item.size === size);

  if (product) {
    if (action === 'increase') {
      product.quantity += 1;
    } else if (action === 'decrease') {
      product.quantity -= 1;
      if (product.quantity <= 0) {
        cart = cart.filter(item => !(item.id === productId && item.size ===
size));
      }
    }
  }

  localStorage.setItem('cart', JSON.stringify(cart));
  displayCart();
}

function updateCartQuantityDisplay() {

```

```

const cart = JSON.parse(localStorage.getItem('cart')) || [];
const totalQuantity = cart.reduce((sum, product) => sum + product.quantity, 0);
const quantityElement = document.querySelector('.action-header__quantity');

if (quantityElement) {
  quantityElement.textContent = totalQuantity;
}
}

document.addEventListener('DOMContentLoaded', displayCart);
document.addEventListener('DOMContentLoaded', updateCartQuantityDisplay);

```

Слайдер

```

const newSlider = document.querySelector('.new__slider');
if(newSlider) {
  new Swiper('.new__slider', {
    // Optional parameters
    loop: true,
    autoHeight: true,
    speed: 800,
    spaceBetween: 35,
    slidesPerView: 4,
    // Navigation arrows
    navigation: {
      nextEl: '.new__arrow--right',
      prevEl: '.new__arrow--left',
    },
    breakpoints: {
      // when window width is >= 320px
      320: {
        slidesPerView: 1.4,
        spaceBetween: 15
      },
      480: {
        slidesPerView: 1.7,
        spaceBetween: 25
      },
      500: {
        slidesPerView: 2,
        spaceBetween: 25
      }
    }
  });
}

```

```
    },  
    // when window width is >= 480px  
    650: {  
      slidesPerView: 2,  
      spaceBetween: 30  
    },  
    // when window width is >= 640px  
    991: {  
      slidesPerView: 4,  
      spaceBetween: 35  
    }  
  }  
});  
}
```