

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«ІНТЕРНЕТ-ПЛАТФОРМА ДЛЯ ПРОДАЖУ «РОЗУМНОГО
ПРИСТРОЮ» ІЗ ВИКОРИСТАННЯМ SWIPER»**

на здобуття освітнього ступеня бакалавр

за спеціальності 126 Інформаційні системи та технології

(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології

(назва)

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

(підпис)

Юрій МІЛЕНЬКИЙ

(ім'я, ПРІЗВИЩЕ здобувача)

Виконав:

здобувач вищої освіти

група ІСД-41

Юрій МІЛЕНЬКИЙ

(ім'я, ПРІЗВИЩЕ)

Керівник

Олег СЕНЬКОВ

к.т.н. доцент

(ім'я, ПРІЗВИЩЕ)

Рецензент:

(ім'я, ПРІЗВИЩЕ)

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІСТ

_____ Каміла СТОРЧАК

“ _____ ” _____ 2025 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Міленькому Юрію Сергійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Інтернет-платформа для продажу «розумного пристрою» із використанням Swiper

керівник кваліфікаційної роботи: Олег СЕНЬКОВ к.т.н., доцент

_____ *(ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “24” лютого 2025 р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.

3. Вихідні дані кваліфікаційної роботи: Готова інтернет-платформа з використаною бібліотекою Swiper.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз основних можливостей бібліотеки Swiper
2. Дослідження концепції повноекранного слайдеру
3. Проєктування та реалізація інтернет платформи.

5. Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	24.02-03.03.25	
2.	Аналіз основних можливостей бібліотеки Swiper	04.03-17.03.25	
3.	Дослідження концепції повноекранного слайдеру	18.03-31.03.25	
4.	Проєктування та реалізація інтернет платформи.	01.04-24.04.25	
5.	Висновки по роботі	25.04-16.05.25	
6.	Розробка демонстраційних матеріалів, доповідь.	19.05-20.05.25	
7.	Оформлення бакалаврської роботи	21.05-30.05.25	

Здобувач вищої освіти _____ **Юрій МІЛЕНЬКИЙ**
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи _____ **Олег СЕНЬКОВ**
(підпис) (ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття ступня магістр: 51 стор., 34 рис., 21 джерел.

Мета роботи – розробка адаптивної та оптимізованої інтернет-платформи для демонстрації та продажу розумного пристрою з використанням бібліотеки Swiper.js, з акцентом на зручний та привабливий інтерфейс.

Об'єкт дослідження – Інтернет-платформи для демонстрації та продажу розумних пристроїв.

Предмет дослідження – Адаптивна вебплатформа з використанням Swiper.js

Короткий зміст роботи. У кваліфікаційній роботі проведено аналіз теоретичних основ створення інтернет платформ, їх поняття, структуру та призначення. Було визначено особливості UX/UI для онлайн продажів. Спроековано особистої інтернет платформи, постановлення задач та подальша реалізація платформи та інтеграція бібліотеки Swiper в проект.

КЛЮЧОВІ СЛОВА: ВЕБ-ДИЗАЙН, АДАПТИВНІСТЬ, КОРИСТУВАЦЬКИЙ ДОСВІД, ІНТЕРФЕЙС, SWIPER, HTML, CSS, JS

ЗМІСТ

ВСТУП.....	6
1 ТЕОРИТИЧНІ ОСНОВИ СТВОРЕННЯ ІНТЕРНЕТ-ПЛАТФОРМИ.....	8
1.1 Поняття інтернет-платформи її структура і призначення.....	8
1.2 Огляд технологій для створення односторінкових сайтів.....	15
1.3 Особливості UX/UI для онлайн-продажів розумних пристроїв.....	18
1.4 Що таке Swiper.js: його призначення, можливості, приклади використання.....	22
2 ПРОЕКТУВАННЯ ІНТЕРНЕТ ПЛАТФОРМИ.....	27
2.1 Постановка задачі, вимоги до систем.....	27
2.2 Архітектура сайту.....	30
2.3 Структура сайту.....	35
3 РЕАЛІЗАЦІЯ ІНТЕРНЕТ ПЛАТФОРМИ.....	38
3.1 Інструменти реалізації, редактор, середовище, бібліотека.....	38
3.2 Реалізація основних елементів сайту.....	41
3.3 Скриншоти з поясненням коду.....	48
3.4 Як Swiper.js інтегровано в платформу.....	53
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59

ВСТУП

Актуальність теми. У сучасному цифровому світі інтернет-платформи стали невід’ємною частиною бізнес-процесів, особливо у сфері електронної комерції (E-commerce). Стрімкий розвиток технологій сприяв масовому переходу підприємств у онлайн-середовище, що забезпечує зручний доступ до товарів та послуг для широкого кола користувачів. Серед різноманітних веб-рішень особливої популярності набули односторінкові сайти (Single Page Applications, SPA) завдяки своїй простоті, швидкості та зручності використання. Такі платформи дозволяють ефективно презентувати продукт, забезпечуючи позитивний користувацький досвід і підвищуючи конверсію продажів.

Однак, незважаючи на переваги, створення якісного односторінкового сайту вимагає використання сучасних інструментів для забезпечення динамічності, адаптивності та візуальної привабливості. У цьому контексті актуальним є застосування бібліотеки Swiper.js, яка дозволяє реалізувати інтерактивні слайдери, повноекранний скрол та інші сучасні ефекти, покращуючи взаємодію з користувачем.

Мета роботи – розробити адаптивну та оптимізовану інтернет-платформу для демонстрації та продажу розумного пристрою з використанням бібліотеки Swiper.js, з акцентом на зручний та привабливий інтерфейс.

Для досягнення мети виконуються такі завдання:

1. Провести аналіз сучасних технологій розробки веб-платформ;
2. Дослідити можливості бібліотеки Swiper.js та сфери її застосування;
3. Спроекувати архітектуру інтернет-платформи;
4. Реалізувати основний функціонал сайту з використанням Swiper.js;
5. Провести тестування платформи та оцінити її зручність.

Об'єкт дослідження – процес створення інтернет-платформи для продажу розумного пристрою.

Предмет дослідження – розробка інтерфейсу з використанням сучасних веб-технологій, зокрема бібліотеки Swiper.js.

Методи дослідження. У роботі використано такі методи:

- теоретичний аналіз наукової літератури та інтернет-джерел;
- проектування інтерфейсу з використанням HTML, CSS та JavaScript;
- імітаційне моделювання інтерактивних елементів за допомогою Swiper.js;
- тестування готового рішення на зручність та адаптивність.

Наукова новизна роботи полягає в застосуванні бібліотеки Swiper.js для створення односторінкового сайту з ефектами повноекранного скролу та інтерактивними слайдами, що покращує візуальне сприйняття продукту та взаємодію з користувачем.

Практична значимість. Розроблена платформа може бути використана як готове рішення для презентації та продажу розумних пристроїв, забезпечуючи високу конверсію завдяки оптимізованому інтерфейсу.

Апробація результатів. Основні положення роботи були представлені на науково-практичних конференціях (за потреби вкажіть конкретні події).

Структура роботи. Дипломна робота складається з трьох розділів:

- Теоретичні основи побудови інтернет-платформ – аналіз сучасних технологій та підходів до розробки.
- Проектування інтернет-платформи – вибір інструментів, архітектура та дизайн.
- Реалізація та тестування – створення функціоналу, перевірка працездатності та користувацького досвіду.

1 ТЕОРИТИЧНІ ОСНОВИ СТВОРЕННЯ ІНТЕРНЕТ ПЛАТФОРМИ

1.1 Поняття інтернет-платформи її структура і призначення

Якщо звернути увагу на те, наскільки стрімко змінився наш світ за останні десятиліття, то перше, що спадає на думку — це повсюдне проникнення інтернету. Сьогодні важко уявити собі повноцінне життя без доступу до онлайн-сервісів: чи то купівля товарів, чи перегляд новин, навчання, спілкування або навіть просто планування подорожі. За всім цим стоять інтернет-платформи — сучасні веб-ресурси, які надають нам можливість взаємодіяти з інформацією, послугами, іншими людьми, не виходячи з дому.

На мою думку, інтернет-платформа — це не просто сайт. Це своєрідна цифрова екосистема, яка створена з тією метою, щоб забезпечити комфортну, зрозумілу і головне — ефективну взаємодію користувача з контентом. Вона може бути дуже різною за виглядом і складністю: від односторінкової візитки, що просто презентує компанію, до складного онлайн-магазину з десятками тисяч товарів.

Що цікаво — майже в кожній сфері життя сьогодні є своя платформа. Візьмемо, наприклад, освіту. Онлайн-курси, електронні щоденники, платформи дистанційного навчання — усе це стало звичним для мільйонів людей. Або медіа — новини тепер оновлюються в реальному часі, без необхідності чекати ранкової газети. Не менш важливою є роль платформ у розвагах — Netflix, Spotify, YouTube замінили традиційні кінотеатри, телебачення і навіть радіо. Ну і, звісно ж, електронна комерція — те, що мене особливо цікавить у цьому дипломному проєкті. Саме тут інтернет-платформи проявляють себе найбільш динамічно: інтерактивні каталоги, швидка оплата, доставка, система знижок — усе це стало звичним для сучасного користувача.

Окрім того, інтернет-платформи виконують низку важливих функцій:

Вони дозволяють поширювати інформацію практично миттєво — і це величезний плюс, особливо коли мова йде про актуальні новини чи оновлення

товару. Дається змога спілкуватись з великою аудиторією, зокрема через чати, форуми, коментарі. Вони дають можливість візуально представляти продукт чи послугу у найкращому світлі — галереї, відео, анімації, інтерактиви відіграють важливу роль. І що не менш важливо — забезпечують зручність: усього кілька кліків — і продукт у кошику, замовлення оформлено, лист підтвердження вже в поштовій скриньці.

Типи інтернет-платформ:

За період мого ознайомлення з різними видами веб-ресурсів я сформував власну уяву про типологію інтернет-платформ. Звісно, існує багато класифікацій, але я б виділив кілька основних типів, які часто зустрічаються у практиці веброзробника:

1. Статичні сайти

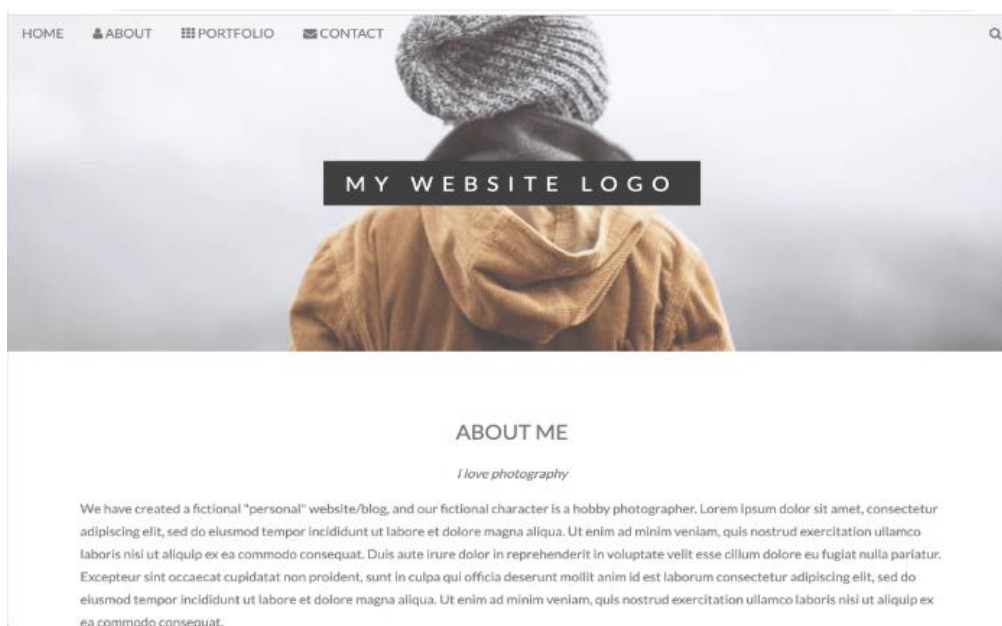


Рис. 1 Приклад статичного сайту

Це найпростіший тип платформи, де вміст зазвичай не змінюється і не передбачає активної взаємодії з користувачем. Я б сказав, що вони нагадують цифрові вивіски — інформативні, але не інтерактивні. Часто такі сайти створюють для невеликих компаній, які просто хочуть бути присутніми в інтернеті. Їхні особливості:

1. Контент оновлюється вручну через HTML-файли.

2. Інтерактивність — мінімальна або взагалі відсутня.
3. Вони швидко завантажуються, бо не залежать від серверної логіки.
4. Не потребують складного хостингу чи бази даних.

2. Сайти-візитки

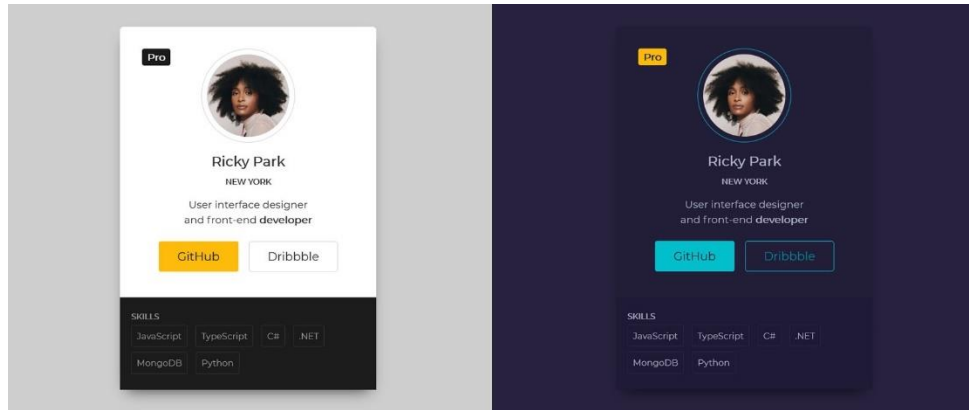


Рис 1 Приклад сайту візитки

Якщо уявити собі компанію або фахівця, який щойно розпочав свою діяльність і хоче мати власну «візитку» в інтернеті — то це і є сайт-візитка. Особисто я бачив багато таких проєктів, особливо серед фрилансерів та малих бізнесів. Вони прості, компактні, часто мають лише одну сторінку з інформацією про послуги, контакти, можливо — кілька фото. Їхні риси:

1. Мінімалістичний дизайн і структура.
2. Зазвичай одна або кілька секцій на одній сторінці.
3. Навігація може бути реалізована прокруткою.
4. Часто обходяться взагалі без JavaScript.

3. Блоги та інформаційні портали

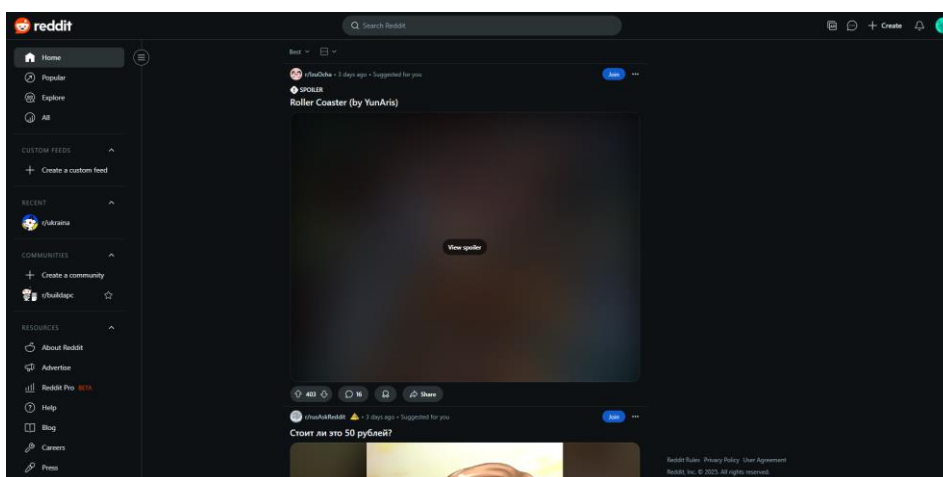


Рис. 2 Приклад інформаційного порталу

Я доволі часто користуюсь блогами — чи то з рецептами, порадами з кодування, чи новинами. Це динамічні платформи, де контент постійно оновлюється. Їх створюють як індивідуальні автори, так і великі медіаресурси. Тут вже не обійтись без бази даних, системи управління контентом (CMS), можливості коментування, рейтингу. Основні особливості:

1. Динамічний контент, який змінюється щодня.
2. Часто реалізуються на WordPress або подібних системах.
3. Підтримують мультимедіа (відео, аудіо, зображення).
4. Можуть містити коментарі, голосування, підписки.

4. Інтернет-магазини

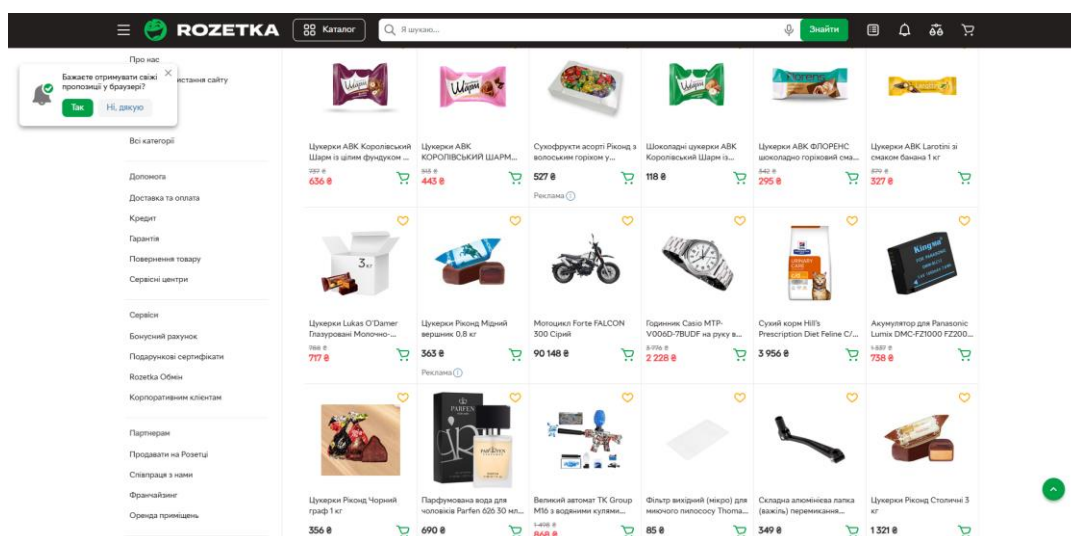


Рис. 3 Приклад інтернет-магазину

Саме цей тип платформ мене зацікавив найбільше. Онлайн-магазини — це складні й потужні системи, які мають десятки функцій, спрямованих на продаж. Я сам часто замовляю щось в інтернеті і щоразу помічаю, наскільки зручно, коли є фільтри, сортування, збереження товару в кошику, можливість переглянути кілька фото, дізнатися відгуки. Розробка таких платформ вимагає інтеграції з платіжними системами, авторизації користувачів, адмінпанелі тощо. Особливостями є :

1. Каталоги товарів з фільтрацією
 2. Система кошика та оплати товарів
 3. Особистий кабінет з особистою інформацією
 4. Адміністративна панель управління товарами
 5. Інтеграція з платіжними системами та службами доставки
 6. Динамічна взаємодія з веб-сайтом без перезавантажень
5. Маркетплейси

Багатофункціональні онлайн-ринки, де розміщуються товари, інформація про товари та послухи надаються від різних продавців. Особливостями є :

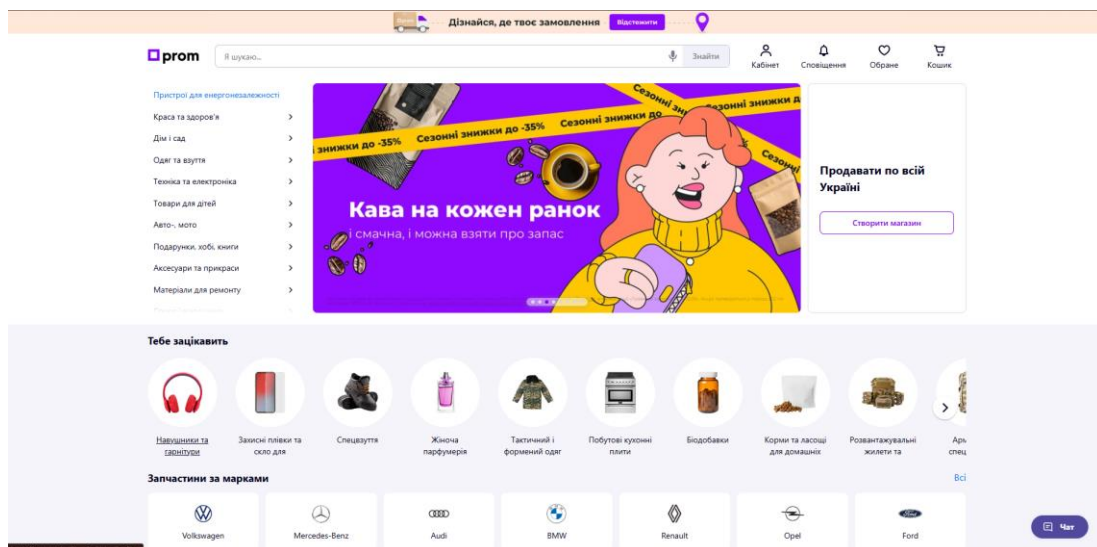


Рис 4. Приклад маркетплейсу

1. Величезна кількість товарів та продавців
2. Рейтинг продавців та відгуки
3. Розширені фільтрації товарів та послуг

4. Особистий кабінет для кожного продавця
5. Висока вимога для продуктивності платформи
6. Односторінкового сайти (SPA)

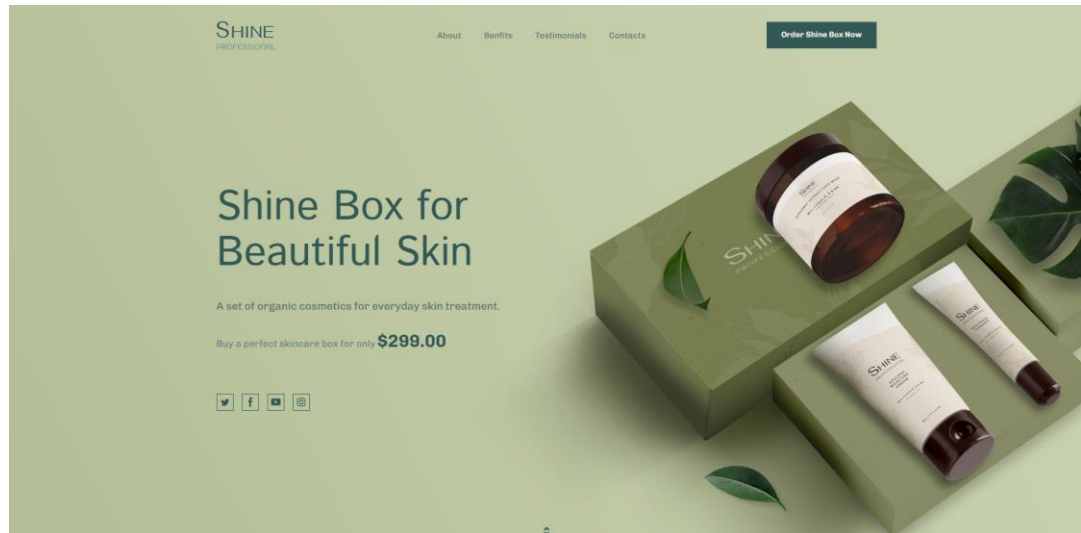


Рис 5. Приклад односторінкового сайти (SPA)

Сучасні платформи, де загальна взаємодія з користувачем знаходиться на одній сторінці та навіть без перезавантажень. Вони є легкими, зручними для користувачів, інтуїтивно зрозумілі. Особливостями є :

1. Висока швидкодія та UX
2. Вся логіка реалізована на стороні клієнта
3. Використання JavaScript- Фреймворків
4. Можливість створення повноцінних додатків

В нашому випадку, ми будемо створювати односторінковий сайт, який дозволяє ефективно презентувати розумний пристрій на функціональному сайті з візуальними ефектами, інтерактивними слайдами та повноекранним прокрутом.

Оптимальним рішенням для платформи є використання JavaScript- бібліотеки Swiper.js. Він допоможить створити інтуїтивно зрозумілий сайт з навігаційним меню, розділами та інтерактивними слайдами.

У межах цієї дипломної роботи було обрано формат сайту.

Це є оптимальним рішенням з кількох причин:

1. Простота реалізації
2. Швидкість роботи: користувач отримує швидкий відгук
3. Мобільна адаптивність: простіше зробити сайт для мобільних користувачів

Сайти особливо підходять для презентацій окремих продуктів, послуг, товарів, брендів тощо. В нашому випадку ми створюємо інтернет-платформу для продажу розумного пристрою.

Структура сайту складається з окремих візуальних блоків, які розміщені один за одним вертикально.

Кожен блок містить певну інформацію про товар (наприклад, опис функцій, переваги, галерею, технічні характеристики тощо) та змінюється за допомогою повноекранного прокруту – коли користувач перегортає сайт, як слайди на презентації

Для реалізації цього функціоналу ми будемо використовувати JavaScript-бібліотеку Swiper.js, яка дозволить нам реалізувати ефект повноекранного скролу(scroll), а також галерею варіацій пристроїв. Це забезпечить платформі сучасний вигляд, інтерактивність та зручну взаємодію з сайтом.

Ключові переваги використання інтернет – платформи :

1. Мінімалізм та увага на продукті – вся структура платформи зведена до демонстрації товару
2. Інтуїтивність інтерфейсу – користувач буде розуміти як взаємодіяти з сайтом
3. Візуальна привабливість – платформа не перенавантажена зайвими елементами
4. Адаптивність – можливість комфортного переглядати як на ПК, так і на смартфонах

Інтернет-платформа створена в межах цієї дипломної роботи, є прикладом сучасного та ефективного інструменту для просування товару та показу можливості створення сучасних веб-сайтів, роблячи їх інтерактивними, зрозумілими не роблячи їх повільними.

1.2 Огляд технологій для створення односторінкових сайтів

Для створення сайтів не обов'язково використовувати багато технологій. Такі сайти створюються як швидкі, зручні та невибагливі платформи для певних надання послуг та товарів. Їх головною перевагою є висока швидкість завантаження, простота навігація та інтуїтивно зрозумілий інтерфейс користувача.

У процесі розробки сучасний веб-платформ часто виникає потреба у створенні легких та ефективних у використанні веб-ресурсів. Односторінкові сайти (так звані Single Page Application або SPA) стали популярними завдяки своїй здатності швидко передавати контент користувачу, не вимагаючи багато ресурсів, та складних архітектурних рішень.

Основна мета таких платформ – забезпечити найбільш комфортну взаємодію з користувачем за допомогою швидкого доступу до інформації, простоти навігації та привабливого зовнішнього вигляду.

У рамках цієї роботи для створення односторінкової інтернет платформи було використано базовий стек технологій, якими є HTML/CSS/JavaScript. Незважаючи на простоту, комбінація цих інструментів дозволяє реалізувати рішення, що відповідає сучасним вимогам до веб – розробки.

HTML – Основа структури сайту.

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Моя тестова сторінка</title>
  </head>
  <body>
    
  </body>
</html>

```

Рис. 6 Приклад коду HTML

Виконує роль “скелета” веб – сторінки. Це мова гіпертекстової розмітки за допомогою якої створюється структура документу та ієрархія елементів – визначаються заголовки, абзаци, списки зображення, посилання тощо. Браузер інтерпретує HTML-документ та відображає його в вигляді графічного інтерфейсу. Кожен елемент сторінки має відповідний тег, це логічно організувати зміст і забезпечити доступність ресурсу для пошукових систем, а також адаптацію під мобільний пристрій

У нашій інтернет – платформі HTML виконує функцію побудови каркасу інтерфейсу, де окремі блоки відповідають за конкретні розділи: «шапка» сайту, головна секція, опис продукту, галерея, контактна інформація, тощо.

CSS – Оформлення візуального вигляду

```

h1 {
  font-size: 60px;
  text-align: center;
}

p,
li {
  font-size: 16px;
  line-height: 2;
  letter-spacing: 1px;
}

```

Рис. 7 Приклад коду CSS

CSS – це мова стилів або таблиця стилів, яка відповідає за зовнішній вигляд інтернет – платформи. За допомогою цієї таблиці стилів задаються:

1. Кольори;
2. Шрифти;
3. Відступи;
4. Положення елементів;
5. Реалізація анімацій;
6. Трансформація;
7. Медіа-запити.

Саме за допомогою каскадних таблиць стилів ми створимо сучасний привабливий дизайн, що забезпечує позитивний досвід користування. В межах цієї роботи CSS буде використовуватися для візуалізації кожної секції сторінки, налаштування елементів слайдеру, оформлення кнопок, кольорових акцентів і тд.

Адаптивна верстка гарантує можливість перегляду на всіх типів пристроїв – від смартфонів до планшетів і моніторів з великою роздільною здатністю. За відповідатимуть медіа-запити які будуть змінювати стилі при різних умовах.



Рис. 8 Приклад адаптивного рішення сайту

JavaScript -динаміка та інтерактивність.

```
var myHeading = document.querySelector("h1");  
myHeading.textContent = "Hello world!";
```

Рис. 9 Приклад коду JavaScript

JavaScript є мовою програмування, він дозволяє додати інтернет платформі інтерактивності, логіку та поведінку. На відміну від HTML та CSS, які лише задають структуру й зовнішній вигляд, JavaScript забезпечує динамічну взаємодію з відвідувачем платформи: обробку дій, зміну контенту без перезавантажень сторінки, реалізацію слайдерів, меню, модальних вікон, форм зворотного зв'язку тощо.

В цій роботі JavaScript виступає ключовим інструментом для інтеграції бібліотеки Swiper.js яка реалізує повноекранний скрол та слайди продукту. За допомогою JavaScript здійснюється контроль за переходи між слайдами, активація меню, адаптація анімацій до розміру екрану та інші інтерфейсні взаємодії. Все це підвищить інтуїтивність роботи платформи та покращить враження користувача

Це основний набір технологій, з яким можна створити повноцінну функціональну інтернет-платформу. У сукупності ці три технології дозволяють створити функціональну, адаптивну та привабливу інтернет платформу, яка буде зручною у використанні на різних пристроях і відповідатиме сучасним вимогам веб-розробки.

Саме така концепція лежить в основі реалізації платформи для демонстрації й просування розумного пристрою що й стало об'єктом дослідження.

1.3 Особливості UX/UI для онлайн-продажів розумних пристроїв

У сучасному цифровому просторі, де конкуренція серед онлайн – продавців є дуже високою, тому і якість користувацького досвіду (UX) та зовнішнього інтерфейсу (UI) мають вони ключову роль у забезпеченні ефективного зв'язку між платформою та потенційним покупцем, зверніть увагу це стосується сфери продажу розумних пристроїв, де продукт сам по собі є технологічним, тому

інтернет в такому стилі повинна мати відповідний вигляд до товару який вона просуває.

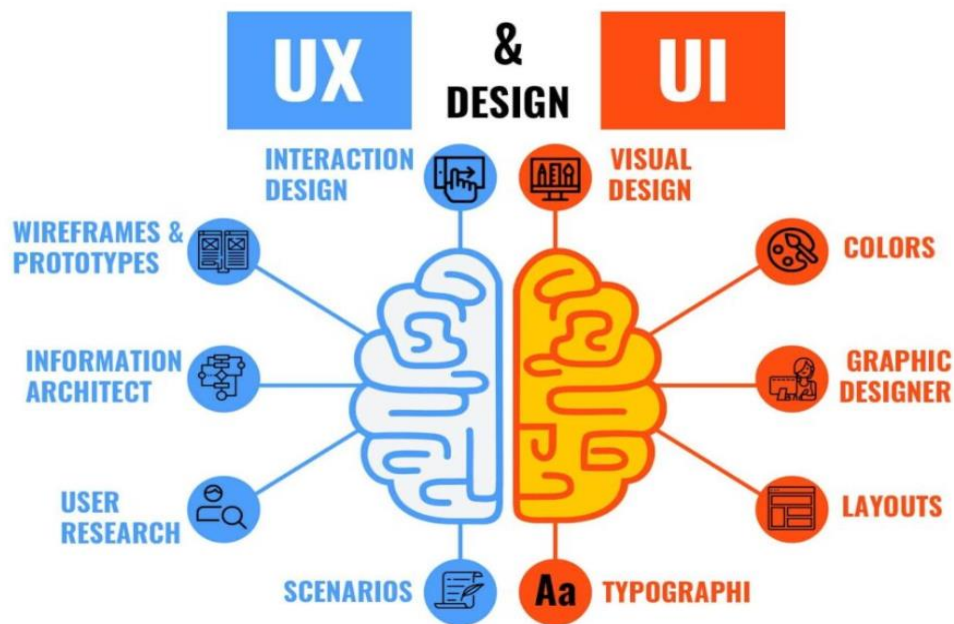


Рис.10 Різниця UX/UI

UX (User Experience) Це загальний досвід, який користувач отримує під час користування сайтом, це сукупність вражень, зручності, які користувач відчуває при взаємодії з сайтом. Головна мета UX – дизайну – зробити цю взаємодію інтуїтивно зрозумілою та результативною і взагалі має бути приємною. У сфері онлайн продажів, на мою думку, UX має дуже великий вплив на кількість користувачів, які перейшли від перегляду до здійснення покупок.

Особливістю UX дизайну є створення зручного та ефективного для користувачів дизайн сайту. Це є зручність навігації, швидкість завантаження сторінки, простота оформлення замовлення.

UI (User Interface) це інструмент, через який користувачі будуть взаємодіяти із сайтом. Саме завдяки UI ми формуємо перше враження про сайт та пристрій. У випадку розумних пристроїв. Інтерфейс має бути чистим, технологічним, стильним та логічним. На мою думку все що стосується розумних пристроїв має виглядати технологічно.

Особливості UX	Особливості UI
----------------	----------------

Користувацькі дослідження: розуміння трендів та цільової аудиторії завдяки глибокому аналізу. Це дозволяє будувати логіку сайту з урахуванням очікувань і досвіду користувача.	Передбачуваність: елементи сайту мають бути на очевидних місцях.
Зручність використання: Інтерфейс має бути розроблений зручним та інтуїтивно зрозумілим, з легкою подачею інформації, дизайн повинен створювати приємний досвід, включаючи людей з обмеженими можливостями	Візуальна ієрархія: важливі елементи (ціна, опис, зображення продукту, кнопка «Купити» мають бути виділяючими.
Інтуїтивність: користувач не повинен думати як користуватись сайтом. Всі кнопки, дії та переходи мають бути очевидними з першого погляду.	Естетика: вся інтернет платформа повинна бути в одному графічному стилі. Візуально, сайт повинен імпонувати користувачу, щоб він розумів що відбувається на екрані,
Швидкість взаємодії: надмірні завантаження, зайві дії та складні процеси реєстрації чи замовлення знижують якість UX.	Уніфікований стиль: Кольори, шрифти, відступи – все має бути зроблено в одному стилі.
Спрощення: Мінімізації процесів виконання задачі. Щоб не було зайвих кнопок, картинок, неважливих функцій та контенту	Доступність: Дизайн сайту повинен враховувати всі можливі потреби та різні здібності за для доступності контенту або просуванню товарів та послуг
Доступність: сайт повинен бути зручним для всіх категорій	Адаптивність: інтерфейс має бути розроблений під якомога більшу

користувачів, включаючи людей з порушення зору чи моторики	кількість пристроїв для охоплення більшої аудиторії
--	---

Онлайн – продаж розумних пристроїв, таких як фітнес – браслети, розумні годинник та інші смарт гаджети тощо, вимагає від нас особливого підходу до створення UX/UI дизайну. Сам пристрій є технологічним, який обіцяє користувачу зручність, функціонал, і стиль. Тому сайт, що його презентує та рекламує, та показувати своїм видом його цінності – через динаміку, логіку, естетику.

Наприклад, реалізація повноекранного скролу(scroll) між розділами сайту яка здійснено за допомогою бібліотеки, створює ефект презентації продукту, схожого на слайд- шоу. Для користувачів це буде візуально привабило та зможе акцентувати увагу на характеристиках продукту, чого і вимагає UI/UX

Переваги якісного UX/UI у контексті онлайн-продажів:	Потенційні недоліки або виклики при реалізації UX/UI:
1. Збільшення довіри користувача — чистий інтерфейс та логічна навігація викликають відчуття надійності.	1. Перевантаження дизайну — надмірна кількість ефектів, анімацій або інформації може відволікати і дратувати.
2. Зменшення показника відмов — зручний сайт утримує відвідувачів довше, що підвищує шанси на покупку.	2. Зловживання шаблонами - банальні рішення без урахування цільової аудиторії знижують
3. Оптимізація шляху користувача - перегляду до замовлення. продумана	3. унікальність платформи.

структура дозволяє швидше пройти шлях від	
4. Позитивні емоції — приємне візуальне оформлення та зрозумілий функціонал формують позитивне враження від бренду.	4. Неврахування доступності — відсутність адаптації під людей з особливими потребами може
5. Адаптивність до трендів — сучасні анімації, плавні переходи, інтерактивні слайди показують, що продукт і платформа «в ногу з часом».	

Важливою перевагою реалізованої платформи є використання бібліотеки Swiper.js, що поєднує ефективність і візуальну привабливість. Завдяки її можливостям, як-от повноекранний скрол(scroll), адаптивні слайди, анімації та кастомізація інтерфейсу, користувач отримує не лише інформативну, а й емоційно привабливу презентацію продукту, що є просто незвичайним ходом. Та ще вона значно спрощує життя розробникам

UX/UI — це не лише візуал чи технічна зручність, це цілісна стратегія, спрямована на задоволення різних потреб та підвищення ефективності розробленої платформи. На мою думку, у випадку з розумними пристроями, саме грамотне поєднання продуманого UX та естетичного UI дозволяє донести до користувача цінність товару і переконати його у доцільності покупки. Чого ми і будемо добиватись в нашій практичній роботі

1.4 Що таке Swiper.js: його призначення, можливості, приклади використання

Swiper.js – це сучасна безкоштовна JavaScript-бібліотека для творення динамічних елементів, такі як слайдер, повноекранні презентації, галереї, на веб-

сайтах, веб-додатках, мобільних додатках. Swiper.js є одною з найпопулярніших бібліотек у сфері веб-розробки. Бібліотека добре оптимізована для користування на мобільних сенсорних телефонах та планшетах, але і не меншою ефективністю функціонує та виглядає на настільному комп'ютері

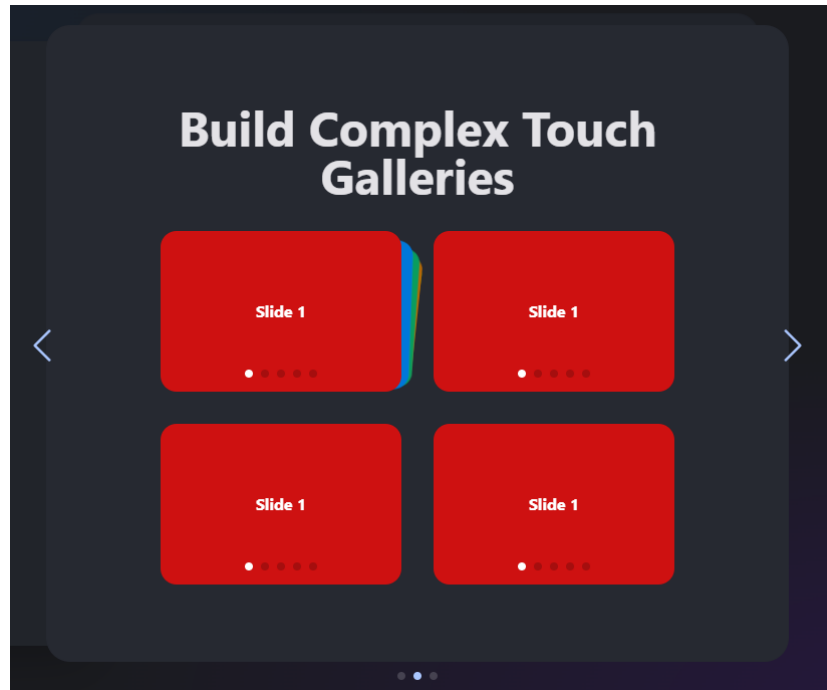


Рис. 11 Приклад роботи Swiper.js

У сучасній веб розробці звичайно важливим є створення інтерфейсів, які були б функціональними та привабливими з точки зору потенційного покупця, саме з такою метою активно використовуються інтерактивні елементи на сайті, які дозволяють покращити загальну навігацію, оптимізувати подачу необхідної інформації та зробити інтернет – платформу динамічною та живою.

Одним із таких потужних інструментів для реалізації таких багатофункціональних рішень є JavaScript – бібліотека Swiper.js. – сучасна, легка, високопродуктивна та дуже популярна, для створення слайдерів, галерей, презентацій.

Основне призначення Swiper.js – надання зручного використання реалізації навігації у вигляді свайпів або слайдів що є дуже ефективним на сенсорних пристроях. Swiper.js відповідає сучасним очікуванням користувача: швидка

взаємодія: плавні анімації: мінімалістичний дизайн та повна адаптивність під різні розміри екранів.

Завдяки цьому інструменту розробники мають можливість не тільки покращити UX/UI, а й значно підвищити привабливість та зекономити місце на платформі.

Swiper.js широко застосовується:

1. Галерея зображень
2. Карусель товарів
3. Вертикальний чи горизонтальний скрол(scroll) між блоками сайту
4. Повноекранних переходів між сторінками у мобільних пристроях

Завдяки зручному API та можливостям бібліотеки, він використовується у дуже багатьох проєктів.

Бібліотека пропонує широкий спектр можливостей, вбудованих функцій, за допомогою яких можна налаштовувати слайдер без обмежень:

1. Горизонтальна та вертикальна навігація слайдів – Можливість створення як класичних горизонтальних слайдерів, так і вертикальних для побудови ефекту сторінкового скролу(scroll);
2. Необмежена кількість слайдів – немає жорстоких лімітів щодо кількості слайдів, що розміщені на сторінці;
3. Підтримка адаптивного дизайну – автоматичне масштабування компонентів під різний розмір пристроїв на якому переглядається контент.
4. Динамічне додавання/ видалення слайдів – гнучка структура DOM, дає можливість змінювати інформацію на сайті, без перезавантажень сторінки
5. Автоматичне прокручування – дає можливість встановити автоматичну зміну слайдів з налаштованим таймером
6. Безшовна прокрутка – прокрутка слайдів у безкінечному циклі

7. Підтримка навігації за допомогою клавіатури – підтримка взаємодії за допомогою стрілок клавіатури, що покращує доступність
8. Вбудовані візуальні ефекти – вбудовані візуальні ефекти, які додають естетичності при зміні слайди
9. Модулі пагінації стрілок, масштабування – такі як пагінація, стрілки, масштабування, зум, мініатюри, а також можливість повної кастомізації зовнішнього вигляду через CSS та JavaScript

Також Swiper.js легко інтегрується з фреймворками або використовується у значайному JS.

Swiper.js може використовуватись у різних ситуаціях:

Вид сайту	Сценарії використання
Інтернет магазини	1. Галереї товарів 2. Слайдери характеристик або відгуки покупців 3. Візуалізація товарних варіацій (колір, форма, модель)
Портфоліо	4. Перегляд зображень робіт з плавним гортанням 5. Вітальні слайди з динамічними описами 6. Відображення процесів роботи крок за кроком
Мобільні сайти	7. навігація між екранами або вкладками свайпом. 8. Слайдер між вкладками без перезавантажень 9. Інтерактивні меню та сторінки презентацією контенту
Новинні сайти	10. Слайдери з останніми публікаціями 11. Горизонтальні стрічки новин 12. Актуальні статті дня або тижня

Серед переваг які роблять Swiper.js таким популярним серед веб-розробників варто виділити наступні:

1. Висока продуктивність та оптимізація – бібліотека написана без якихось залежностей, що забезпечує високу продуктивність та стабільну роботу сайту
2. Повна адаптивність – правильна робота на будь-якому пристрої чи розміру екрану це дає можливість охопити більшу аудиторію
3. Легкість у використанні – зрозуміла та доступна документація до бібліотеки та приклади, які дозволяють швидко в ньому орієнтуватись

У межах цієї роботи Swiper.js використовуватиметься для розробки односторінкового сайту та повноекранним скролу(scroll) між секціями сайту, прокрутки зображень варіації товару, динамічним описом до кожного зі слайдів.

2 ПРОЕКТУВАННЯ ІНТЕРНЕТ ПЛАТФОРМИ

2.1 Постановка задачі, вимоги до систем

У сучасному інтернет середовищі, де темп розвитку електронної комерції швидко зростають, важливими критерієм для розробки є якість, функціональність та зручність у використанні інтернет платформ для продажу товарів та технологічної продукції.

У зв'язку з цим постає необхідність у створенні ефективного програмного рішення, яке б забезпечило швидкий доступ до інформації про продукт, зручну навігацію, інтерактивну взаємодію користувача з інтерфейсом, а також безперебійну роботу систем на різних пристроях.

У межах цієї роботи розглядається створення односторінкової інтернет – платформи для продажу розумного пристрою із використанням Swiper для реалізації сучасного інтерфейсу взаємодії з користувачем або потенційним покупцем, а також організації повноекранного скролу(scroll) між блоками сайту.

Основна мета – розробити інформаційно – презентаційний сайт, що поєднує в собі інноваційний візуальний підхід, зручність використання для користувача і технічну ефективність поставлених функціональних завдань.

Розробка інтернет – платформи передбачає вирішення ряду важливих задач, які відносяться як функціональна складова системи, так і зовнішній вигляд, адаптивність та інтерактивність.

Постановка задач ґрунтується на необхідності створення ресурсу, який би відповідав сучасним вимогам до електронної торгівлі, а також враховував би факт презентації саме розумного пристрою(smart device).

Створення адаптивного, зручної, інформативної, інтерактивної інтернет – платформи для продажу розумного пристрою з використанням інструментів сучасної фронтенд – розробки.

Основні задачі, які необхідно вирішити в процесі розробки:

1. Розробити загальну архітектуру веб – платформи.
2. Створити привабливий інтерфейс користувача (UI), який ефективно представлятиме функції та переваги пристрою.
3. Реалізувати плавну та інтуїтивно зрозумілу навігацію між секціями сайту за допомогою повноекранного скролу.
4. Впровадити динамічні слайдери зображень, описів, характеристик, та варіантів пристрою, використовуючи бібліотеку Swiper.js.
5. Забезпечити кросбраузерну та кросплатформену сумісність веб – сайту.
6. Передбачити адаптивність інтерфейсу для мобільних пристроїв.
7. Реалізувати функції зміни опису під різні варіації девайсу
8. Оптимізувати сайт для швидкого завантаження та мінімального використання ресурсів.

Вимоги до розроблюваної системи можна поділити на кілька основних категорій: функціональні, нефункціональні, технічні, інтерфейсні, Кожна з категорій охоплює різні аспекти, що забезпечують повноцінну роботу інтернет – платформи

Функціональні вимоги:

1. Виведення головного екрану з базовою інформацією про пристрій.
2. Перегляд слайдів з детальними характеристиками товару
3. Підключення повноекранного скролу(scroll) між секціями сайту.
4. Реалізація інтерактивного слайдера різними варіантами моделі.
5. Динамічний текстовий супровід кожного слайду.
6. Наявність секцій з відгуками.
7. Кнопка переходом до умов покупки або контактна форма.
8. Реалізація плавних анімацій та переходів між секціями контенту

Нефункціональні вимоги:

1. Висока швидкодія – швидке завантаження платформи та влвна взаємодія з контентом

2. Надійність – стабільна робота на різних пристроях
3. Естетичність – привабливий дизайн, зусереджений на одному візуальному стилі
4. Адаптивність – повна підтримка мобільних пристроїв: від смартфонів до широкоформатних моніторів
5. Незалежність від серверної частини – сайт має працювати без потреби в бекенді.

Технічні умови:

1. HTML5, CSS3 – основні сучасні технології верстки та стилізації нашого контенту
2. JavaScript – Реалізація інтерактивності та логіки сайту
3. Бібліотека Swiper.js – основа для побудови слайдерів та повноекранного скролу.
4. Підтримка ES6+ - використання сучасного синтаксису Javascript.
5. Можливість локального запуску через звичайний браузер без додаткових налаштувань.

Вимоги до користувацького інтерфейсу:

1. Панель навігації
2. Повноекранний скрол(scroll)
3. Слайдер з візуальними елементами
4. Динамічні текстові блоки
5. Кнопки СТА, які спонукають до перегляду, замовлення або зв'язку.

У результаті реалізації поставленої задачі ми повинні отримати повністю функціонуючу односторінкову інтернет платформу, яка дозволяє ефективно презентувати розумний пристрій кінцевому споживачу. За рахунок інтерактивного слайдера, динамічним описом, сучасним дизайном, користувач зможе зручно переглядати контент. Знайомитись з перевагами пристрою. А також здійснювати дії що приведуть його до придбання.

2.2 Архітектура сайту

Архітектура сайту — це сукупність структурних, функціональних та логічних компонентів, які забезпечують ефективне функціонування веб-ресурсу, підтримку взаємодії між користувачем і системою, а також реалізацію поставлених цілей проекту.

Розглядаючи архітектуру веб-сайту, слід зазначити, що вона представляє собою складну систему взаємопов'язаних компонентів, які в сукупності забезпечують ефективну роботу ресурсу. У нашому випадку, при створенні платформи для продажу розумного пристрою, архітектура грає вирішальну роль не лише у технічній реалізації, а й у формуванні якісного користувацького досвіду.

У контексті розробки інтернет-платформи для продажу розумного пристрою архітектура виступає, не лише технічним каркасом, а й основою зручності використання, стабільності, масштабованості, адаптивності ресурсу.

Сучасні вимоги до архітектури таких сайтів є не лише високі стандарти користувацького досвіду (UX/UI), а й необхідністю швидкої роботи з контентом, ефективної інтеграції з аналітичними системами та зручного керування.

Загальні підходи до побудови архітектури односторінкового сайту

Односторінковий додаток (SPA) відрізняється від багатосторінкових ресурсів тим, що вся взаємодія з користувачем відбувається в межах одного HTML-документа.

Під час навігації сторінка частково оновлюється за допомогою асинхронних запитів, що дозволяє уникнути повного перезавантаження сторінки, забезпечуючи більш плавний і швидкий користувацький досвід.

Архітектура SPA передбачає тісну взаємодію між фронтом і динамічним контентом. У нашому випадку платформа є односторінковою, вона

буде розроблена з використанням HTML, CSS, JavaScript та бібліотеки Swiper.js, яка дозволяє створювати динамічні інтерактивні компоненти, такі як повноекранна прокрутка та слайдери продуктів.

Основні компоненти архітектури сайту

Архітектура сайту включає декілька ключових складових:

У рамках реалізації проекту інтернет-платформи передбачено побудову логічної, чіткої та лаконічної структури, що міститиме такі основні секції:

Головний екран (main section)

- зображення з назвою пристрою та слоганом

Про товар (Product Section)

- детальна інформація про функціональні можливості розумного пристрою.

Галерея / Візуальні варіації (Swiper Section)

- Інтерактивний слайдер товару

Форма замовлення або зворотного зв'язку (Call to Action)

- інтерактивна форма для збору контактних даних.

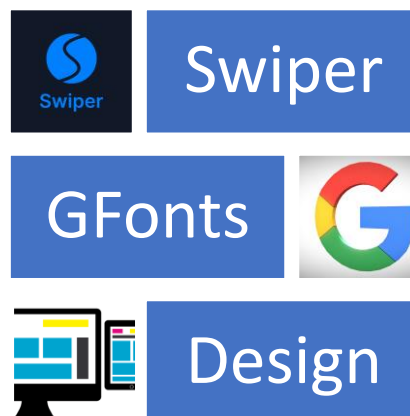
Контакти / Футер (Footer)

- контактна інформація, соціальні мережі, політика конфіденційності.

Фронтенд архітектура базується на класичних веб-технологіях:



- HTML5 — створення семантичної структури та елементів сторінки.
- CSS3 (Flexbox, Grid, Media queries) — реалізація адаптивного дизайну, стилізація всіх компонентів.
- JavaScript (Vanilla JS) — програмна логіка для обробки подій, анімацій, взаємодії зі Swiper.js.

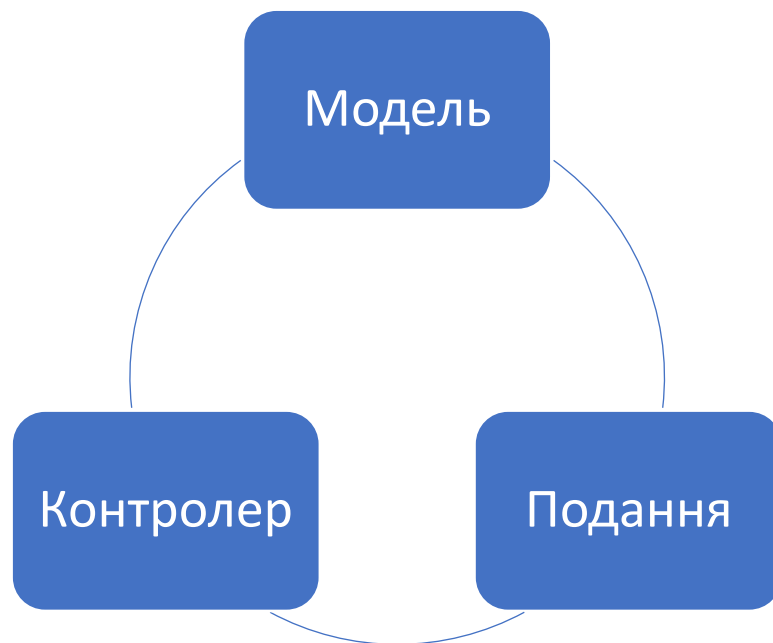


- Swiper.js — реалізація динамічного повноекранного скролу та каруселей з варіантами товару.
- Google Fonts, FontAwesome — для покращення типографіки та візуальних іконок.
- Responsive Design — підтримка масштабування інтерфейсу відповідно до розмірів екрану.

-

Архітектурна модель взаємодії компонентів

У нашій системі застосовується принцип MVC (Model-View-Controller) у спрощеному вигляді. Хоча класична реалізація MVC притаманна складнішим фреймворкам, навіть у проєктах без серверної частини цей принцип дозволяє логічно відокремити:



- Модель (Model) — структура даних, наприклад, інформація про товар, ціни, характеристики.
- Подання (View) — візуальні елементи, що відображаються користувачу (слайдери, тексти, кнопки).
- Контролер (Controller) — JavaScript-логіка, яка реагує на дії користувача, наприклад, свайпи, натискання кнопок, прокрутка.

Переваги обраної архітектури:	Певні обмеження, які варто враховувати:
Реалізація сайту на основі SPA та використання Swiper.js має низку суттєвих переваг:	Початкова залежність від JavaScript — при вимкненому JS сайт може частково втратити функціональність.
Швидкість завантаження та реакції — завдяки роботі без повного перезавантаження сторінки.	Проблеми з SEO — SPA складніше індексуються пошуковими системами, якщо не реалізовано SSR.
Адаптивність та кросбраузерність — сайт коректно відображається на будь-яких пристроях та браузерах.	Можливе збільшення початкового часу завантаження — у разі великої кількості вбудованих зображень або скриптів.
Зручність навігації — реалізована логічна структура з переходами між секціями свайпом або скролом.	Обмежена інтеграція з серверною логік — складніше реалізовувати динамічний контент без бекенду.
Простота в обслуговуванні — модульність дозволяє легко змінювати окремі блоки без впливу на решту сайту.	Початкова залежність від JavaScript — при вимкненому JS сайт може частково втратити функціональність.
Можливість подальшого розширення — архітектура дозволяє додати інтеграцію з CMS, аналітичними сервісами, CRM.	

Таким чином, архітектура розроблюваного сайту ґрунтується на принципах простоти, доступності та зручності. Обрана структура дозволяє ефективно презентувати товар, зберігаючи при цьому високий рівень інтерактивності та адаптивності.

2.3 Структура сайту

Використання сучасних технологій, таких як JavaScript та Swiper.js, забезпечує плавну навігацію між розділами, привабливий зовнішній вигляд та позитивний користувацький досвід. Архітектура платформи є гнучкою, тому ви можете адаптувати її до змін у функціоналі, дизайні чи потребах бізнесу в майбутньому.

Структура сайту - це логічне та функціональне розташування інформаційного контенту, елементів інтерфейсу та навігації, які забезпечують ефективну взаємодію між користувачем та веб-платформою. Добре організована структура є одним з основних факторів, що впливає на зручність використання, швидкість орієнтації та візуальне сприйняття. У нашому випадку структура орієнтована на просування та продаж одного конкретного смарт-пристрою, тому вона має бути лаконічною, логічно завершеною, інформативною та візуально привабливою водночас.

Основні принципи побудови структури сайту

1. Під час розробки структури веб-сайту було враховано наступні принципи:
2. Принцип ієрархії — кожен елемент має своє логічне місце у загальній структурі.
3. Принцип цілісності — усі елементи сайту виконують спільну функцію й утворюють єдину платформу.
4. Принцип лаконічності — мінімум відволікаючих факторів, фокус на основному контенті.
5. Принцип мобільності — структура повинна коректно відображатися на різних пристроях.

6. Принцип повторюваності патернів — послідовна структура блоків і секцій сприяє легшому сприйняттю.

Тип структури

Обрана структура відповідає концепції односторінкового сайту (SPA), де вся інформація подається на одній веб-сторінці, з логічним поділом на секції. Переміщення між секціями здійснюється за допомогою:

1. повноекранного скролу (реалізованого за допомогою бібліотеки Swiper.js);
2. навігаційного меню (якірні посилання, які ведуть до відповідних блоків сторінки).

Логічні блоки сайту

Сайт містить кілька основних блоків, які розміщуються вертикально один за одним. Кожен блок має свою інформаційну та функціональну мету:

1. Головний екран (Main Section)
2. Назва пристрою: SmartHub Air
3. Слоган або основне повідомлення: короткий опис призначення пристрою
4. Заклик до дії: кнопка «Замовити зараз»
5. Зображення пристрою або відеофон
6. Іконка прокрутки (наприклад, стрілка вниз або swipe-індикатор)
7. Галерея пристрою (Gallery Section)
8. Вбудований Swiper-слайдер із зображеннями пристрою з різних ракурсів
9. Можливість перемикання слайдів кнопками або свайпом
10. Короткі описи або назви під кожним зображенням
11. Приклад реалізації: зображення стандартної, міні- та професійної версії пристрою
12. Можливість гортання свайпом або кнопками
13. Реалізовано за допомогою бібліотеки Swiper.js
14. Форма зворотного зв'язку / покупки (Contact / Order Section)
15. Стандартна форма з полями введення:

- 16.Ім'я
- 17.Електронна пошта
- 18.Номер телефону
- 19.Кнопка дії: «Замовити» або «Отримати консультацію»
- 20.Футер (Footer)
- 21.Контактна інформація: email, телефон, адреса

Навігаційне меню для швидкого переходу до ключових секцій сайту. Навігація в межах SPA реалізована як горизонтальне фіксоване меню, яке дозволяє швидко переходити між секціями сторінки. Якірні посилання ведуть до конкретного блоку. Меню автоматично адаптується до ширини екрана.

Структура HTML-документа

У технічному плані структура сайту має таку загальну послідовність тегів: Кожна секція логічно виділена, має свій унікальний ідентифікатор та стилі.

Мультимедійна структура

Для посилення візуального впливу та залучення уваги користувача до ключових блоків застосовано: Адаптивні зображення (роздільні для мобільних і десктопних пристроїв); Інтерактивні переходи між блоками (анімації при прокрутці); Іконки у векторному форматі (SVG) для збереження якості при масштабуванні.

Таким чином, структура сайту, що розробляється в межах даного проекту, є чіткою, функціональною, логічно завершеною та адаптованою до потреб сучасного користувача.

Вона дозволяє ефективно презентувати продукт, забезпечити зручну навігацію, візуально привабити потенційного покупця та сприяти формуванню позитивного враження від взаємодії з платформою. Такий підхід до структурування контенту сприяє підвищенню рівня конверсії, довіри до бренду та покращенню загального UX.

3 РЕАЛІЗАЦІЯ ІНТЕРНЕТ ПЛАТФОРМИ

3.1 Інструменти реалізації, редактор, середовище, бібліотека

Для реалізації інтернет-платформи продажу розумного пристрою Я обрав найпопулярніші інструменти для розробки, які забезпечують мені ефективне створення, тестування та оптимізацію веб-додатку. У цьому розділі детально розглядаються:

1. Середовище розробки (Visual Studio Code),
2. Браузер для тестування (Google Chrome),
3. Ключові бібліотеки (Swiper.js),

Додаткові плагіни (CSS Peek, Live Server тощо).

1. Visual Studio Code як основний редактор коду

Visual Studio Code (VS Code) — це легкий, але функціональний редактор коду з відкритим вихідним кодом, розроблений Microsoft. Він підтримує роботу з HTML, CSS, JavaScript та іншими мовами, що робить його ідеальним вибором нашої задачі.

CSS Extractor - ще одне корисне розширення для VS Code, яке дозволяє зручно переносити вбудовані стилі або стилі, написані в HTML, у зовнішні CSS-файли. Це значно спрощує структуру коду, забезпечує розділення логіки і стилів (що відповідає принципам чистої архітектури) і дозволяє повторно використовувати CSS класи. Завдяки цьому плагіну можна перенести назви класів написані в HTML, до файлу в CSS, не перевіряючи правильність написаних назв класів. Що значно скорочує час написання проекту

Live server – Дозволяє запускати розроблений проєкт на локальному сервері, це дозволяє працювати над проєктом в декілька людей, або своєю командою. З інших плюсі, при кожному зміні на сайті, не треба перезавантажувати сторінку, щоб побачити зміни, плагін робить це автоматично як тільки ми зберігаємо зміни в файлі проєкту

Переваги VS Code для даного проекту:

1. Інтегрований термінал — дозволяє запускати сервери, встановлювати залежності (наприклад, через npm) без перемикання між вікнами.
2. Підтримка Emmet — прискорює написання HTML/CSS за допомогою шаблонів (наприклад, `div.slider>ul.slides>li*5`).
3. Вбудований Git — полегшує контроль версій без додаткових програм.
4. Розширення (плагіни) — значно розширюють функціонал

Налаштування для проекту:

1. Використання тем оформлення (наприклад, Dark+) для зручності роботи.
2. Включення форматування коду на збереження (Prettier) для дотримання єдиного стилю.

2. Google Chrome для тестування та налагодження

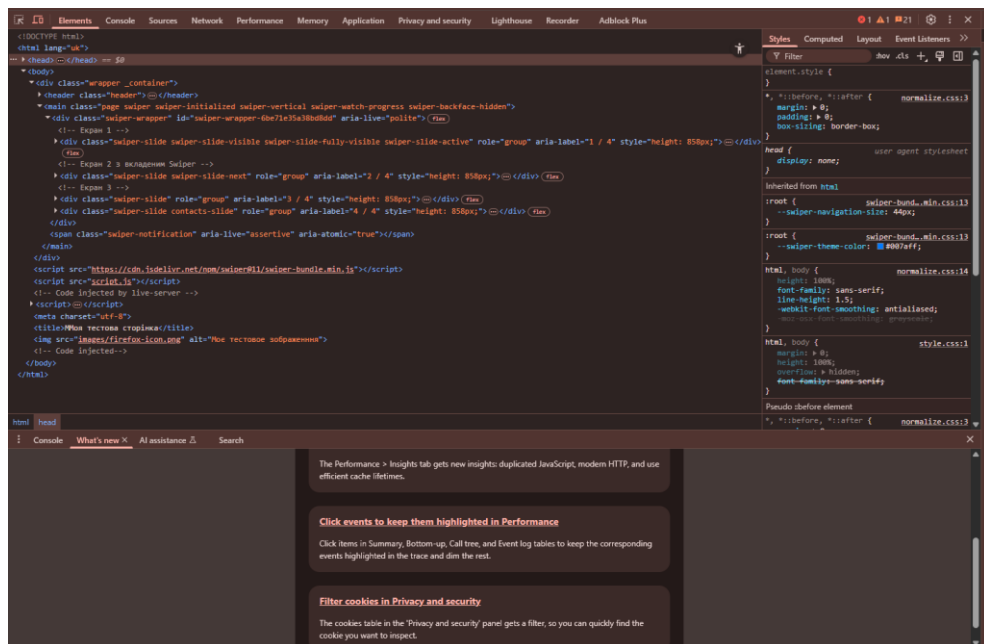


Рис. 3.1 Приклад використання DevTools

Google Chrome — основний браузер для тестування платформи через:

1. Інструменти розробника (DevTools) — дозволяють аналізувати HTML-структуру, CSS-стилі, помилки JavaScript, мережеві запити тощо.

2. Режим адаптивного дизайну — перевірка відображення на різних пристроях (мобільні, планшети, десктопи).
3. Логування консолі — відстеження помилок JS та результатів виконання скриптів.

3. Бібліотека Swiper.js для створення слайдерів

Swiper.js — сучасна JavaScript-бібліотека для реалізації інтерактивних слайдерів, яка була обрана для:

1. Повноекранного скролу — плавна навігація між розділами сайту.
2. Адаптивних слайдів — демонстрація товару з ефектами переходу.
3. Тач-підтримки — коректна робота на мобільних пристроях.

Приклад ініціалізації Swiper у проєкті:

```
const swiper = new Swiper('.swiper-container', {
  direction: 'vertical', // Вертикальний скрол
  loop: true, // Безкінечний цикл слайдів
  pagination: {
    el: '.swiper-pagination', // Точки навігації
    clickable: true,
  },
});
```

Рис 3.2 Приклад ініціалізації Swiper.

Переваги Swiper.js:

1. Модульність — можна підключати лише необхідні функції (наприклад, пагінацію або кнопки навігації).
2. Документація — чіткі приклади та API для швидкої інтеграції.
4. Додаткові плагіни для підвищення продуктивності

Для прискорення розробки було використано такі розширення VS Code:

1. Live Server - Запускає локальний сервер з автоматичним перезавантаженням сторінки при зміні коду. Доступ через IP — можливість тестування на інших пристроях в одній мережі.

2. CSS Peek. Швидкий перехід до стилів — при наведенні на клас у HTML-файлі відкриває відповідний CSS. Підтримка SCSS/LESS — корисна для складних стилів.
3. Auto Rename Tag -Автоматичне перейменування парних тегів (наприклад, `<div>` → `<section>`).
4. Prettier. Форматування коду — автоматичне вирівнювання HTML/CSS/JS за єдиним стандартом.

Обраний стек інструментів забезпечив ефективний процес розробки:

1. VS Code як центральне середовище,
2. Swiper.js для динамічного інтерфейсу,
3. Google Chrome для налагодження,
4. Плагіни для автоматизації рутинних задач.

Ці інструменти дозволили створити сучасну, адаптивну платформу з мінімальними витратами часу на технічні аспекти.

3.2 Реалізація основних елементів сайту

Працюючи над інтернет-платформою Smart Device для мого проєкту AIR SMARTHUB, я вирішив створити саме односторінковий веб-сайт, оскільки цей підхід, на мою думку, найкраще поєднує простоту використання, швидкість доступу до інформації та сучасний вигляд. Сторінка вийшла компактною, але водночас достатньо гнучкою, аби презентувати головну ідею проєкту і дати користувачу повне уявлення про пристрій.

Одним з ключових завдань було зробити платформу зручною та доступною, у тому числі й для людей з обмеженими можливостями. Я завжди вважав, що справжній цифровий продукт має бути інклюзивним. Мені хотілося, щоб користувач, незалежно від пристрою чи можливостей, зміг легко знайти потрібну інформацію, без зайвих дій і зусиль.

У розробці я використав HTML5, CSS3 та бібліотеку Swiper.js, яка дозволила створити повноекранний макет-слайдер. Я вперше працював зі Swiper

і був приємно здивований, наскільки легко з її допомогою будувати розділи, що плавно змінюються один за одним. Кожен блок на сторінці став окремим слайдом. Це не тільки розділило вміст на логічні частини, але й створило ефект безперервного потоку, коли користувач “ковзає” через розділи, не втрачаючи контексту. Особисто для мене, як для розробника, важливо було не просто вивести контент, а викликати відчуття залученості — і Swiper у цьому дійсно допоміг.

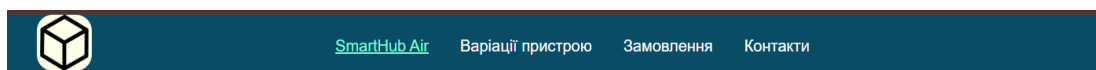


Рис. 3.3 Шапка, логотип, елементи навігації сайту

Особливу увагу я приділив UI/UX-дизайну. Я переконаний, що зовнішній вигляд сайту має не тільки приваблювати, але й пояснювати — де що розташовано, куди натискати, як швидко знайти інформацію. Саме тому я обрав мінімалістичний, але логічно структурований дизайн, де кожна кнопка, кожен відступ, кожен елемент має свою функцію. В основі сторінки — контейнер-обгортка, в який я помістив усі основні елементи: заголовок, меню, контентні блоки та навігацію.

На самому верху сайту я розмістив логотип компанії — елемент, який несе не лише візуальну функцію, а й формує брендову впізнаваність. Поряд розташоване меню, яке включає такі розділи, як "Smarthub Air", "Варіації пристроїв" і ще один, який я залишив на майбутнє. Для мене було важливо, щоб навігація працювала просто: нічого зайвого, без складних багаторівневих списків. Людина бачить — натискає — переходить.

Також я подбав про адаптивність — сьогодні більшість користувачів заходять на сайти з мобільних, і я не хотів, щоб вони відчували дискомфорт або потребу “приближати” текст. Тому верстка автоматично підлаштовується під будь-яку ширину екрана, а елементи перетікають у компактний вигляд без втрати функціональності.

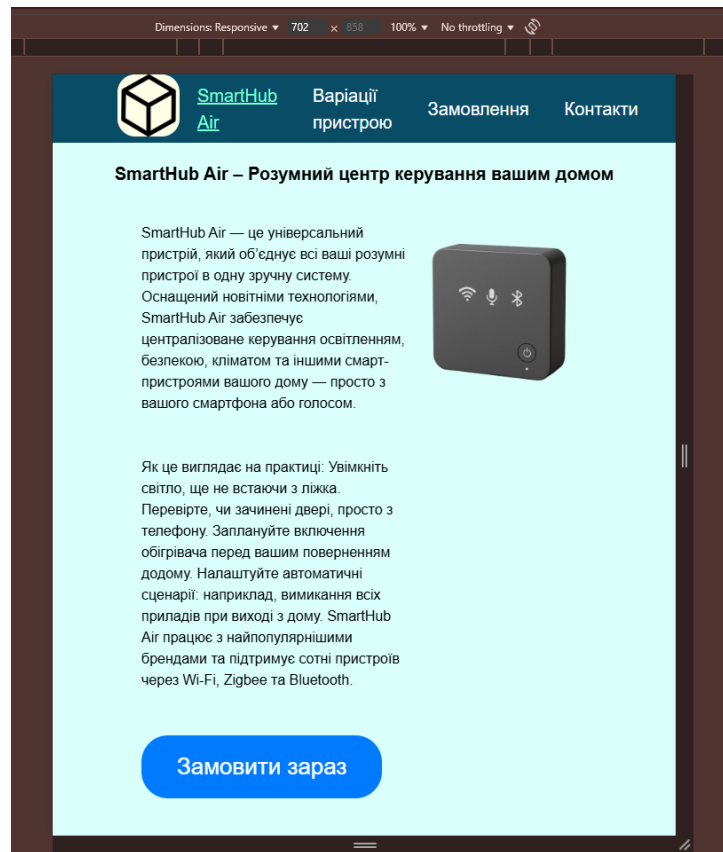


Рис. 3.4 Адаптована версія сайту

Можливо, найбільше задоволення мені принесла робота над анімованими переходами між слайдами. Це ніби дрібниця, але саме такі речі формують відчуття “живого” продукту. Коли все рухається плавно, без ривків — користувач не напружується, а просто насолоджується процесом.

Підсумовуючи, я можу сказати, що створення цієї платформи стало для мене не лише технічним завданням, а й можливістю реалізувати своє бачення сучасного, легкого у сприйнятті веб-продукту. Я хотів не просто створити черговий сайт, а щось цікаве, інтуїтивне і людське — і, як мені здається, це вдалося. Основний вміст веб-сайту організований за допомогою слайдів Swiper, кожен слайд містить унікальну функціональну та змістовну інформацію. Початковий слайд вводить основну модель гаджета AIR SMARTHUB, що відображає текст, який пояснює функції пристрою, його передбачуване використання та практичні приклади додатків.

Зображення з пристроєм розміщується біля тексту, що дозволяє вам об'єднати те, що ви бачите з письмовим поясненням.

У нижній частині слайда ви знайдете кнопку "замовлення зараз". Ця кнопка є прямим запрошенням для користувачів для запуску процесу замовлення. Основний вміст вебс-айту реалізовано у вигляді Swiper-слайдів, кожен з яких має свій функціональний і семантичний зміст.

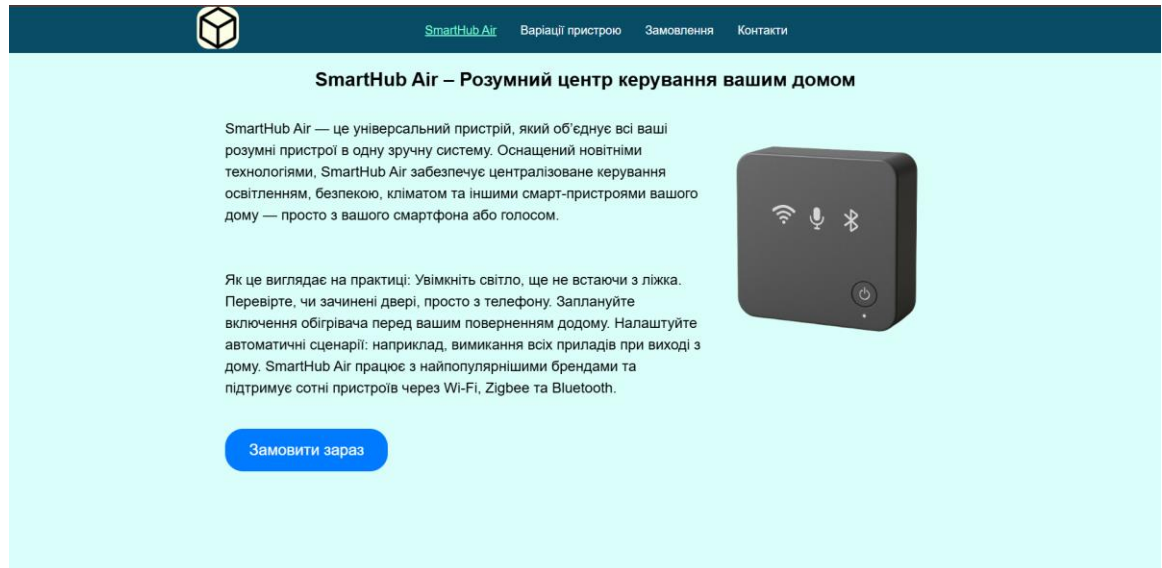


Рис 3.5 Перша секція сайту

Перший слайд — це презентація основної моделі пристрою SmartHub Air, де користувач отримує текстову інформацію про можливості пристрою, його призначення, а також приклади практичного використання.

Візуальний контент (зображення пристрою) розташовано поряд із текстом, що дозволяє поєднати візуальне сприйняття з текстовим описом. У нижній частині цього слайду розміщено кнопку «Замовити зараз», яка є активним закликом до дії (СТА — Call to Action) та веде користувача до процесу оформлення замовлення.

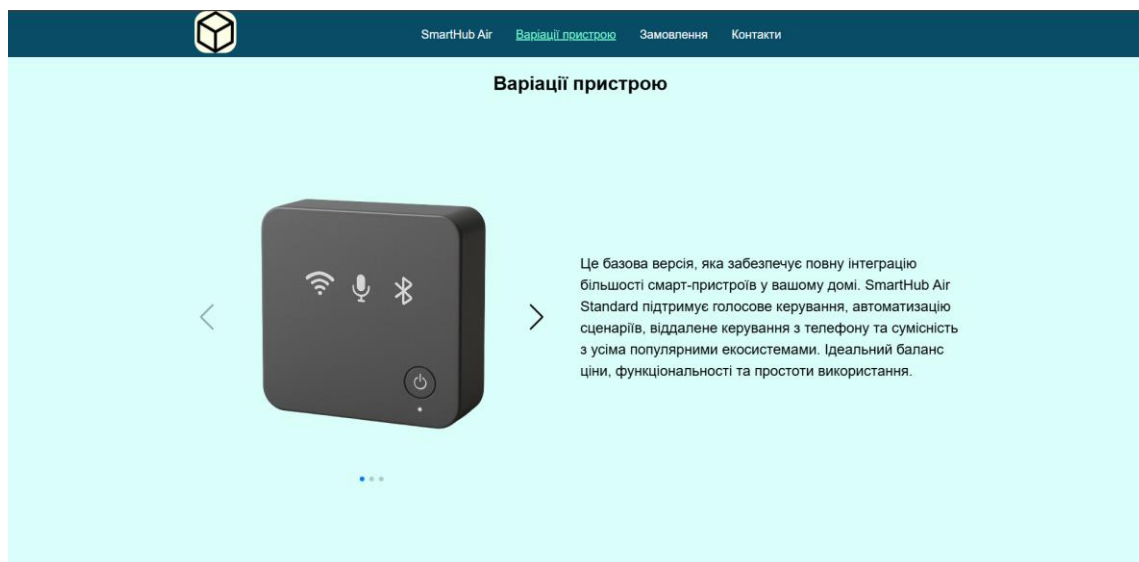


Рис. 3.6 Друга секція сайту

На другому слайді я представляю різні моделі SmartHub Air, такі як міні, стандарт та професіонал. Щоб додати цю функцію, я використовував Swiper, по суті, карусель зображень, безпосередньо інтегрована у слайд. Ця установка дозволила мені спостерігати за змінами пристрою, не виходячи з поточного екрану.

Цей метод покращить взаємодію користувачів та дозволить вам спостерігати за змінами в інтерактивній формі. Swiper дозволяє користувачам орієнтуватися за допомогою клавіш зі стрілками, і функцією спливаючого вікна, пропонуючи гнучкий спосіб взаємодії.

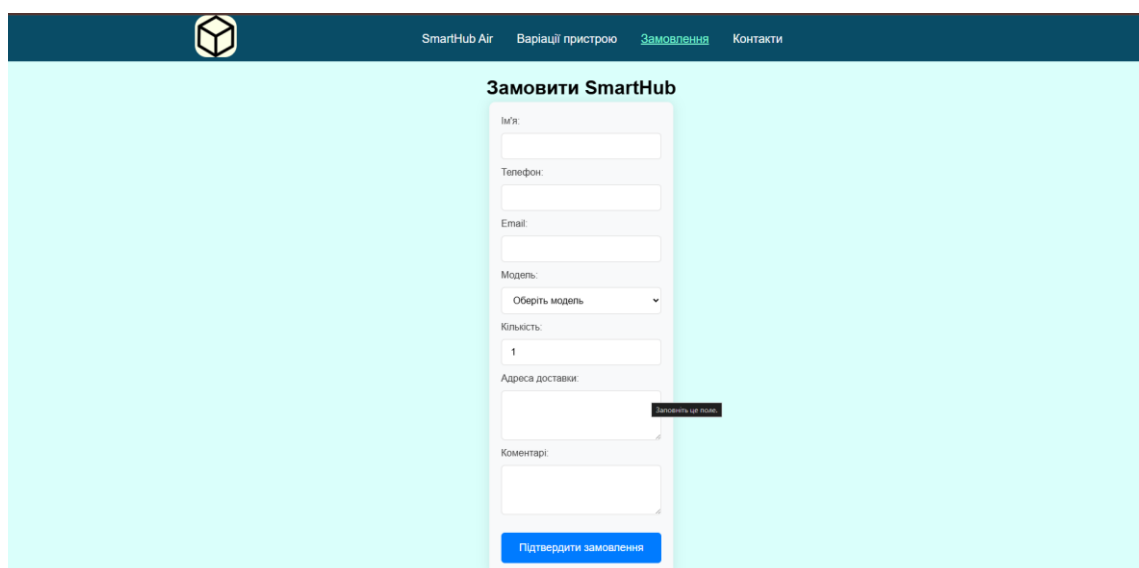


Рис. 3.7 Третя секція сайту

Третій слайд на поточному етапі виконує роль замовлення товару у вигляді заповнення форми, що досить сильно полегшує процес замовлення товару та економить час для користувача.

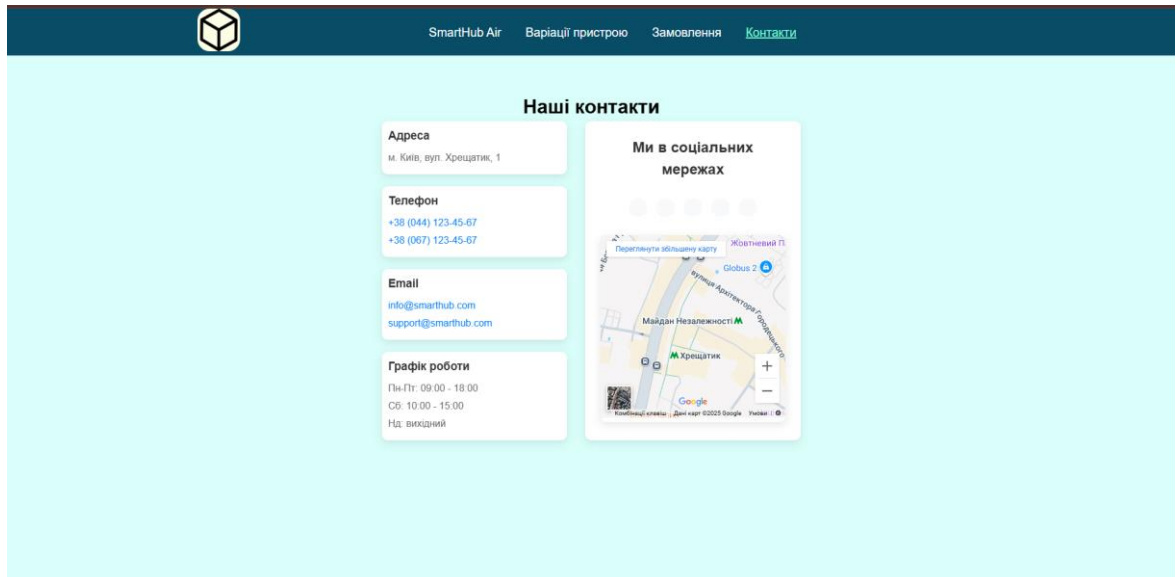


Рис 3.8 Четверта секція сайту

На цій секції сайту я вирішив розмістити контактні данні, що є корисним та зручним для користувачів, які не хочуть чи не можуть замовляти онлайн, додана вся необхідна інформація, додана мапа для зручності

Сторінка реалізована з розрахунку на (адаптивну) верстки. Всі блоки масштабуються відповідно до ширини екрана, що забезпечує коректне відображення як на настільних пристроях, так і на мобільних пристроях.

```

1  *,
2  *::before,
3  *::after {
4    margin: 0;
5    padding: 0;
6    box-sizing: border-box;
7  }
8
9  a{
10   color: inherit;
11   text-decoration: none;
12 }
13
14 html, body {
15   height: 100%;
16   font-family: sans-serif;
17   line-height: 1.5;
18   -webkit-font-smoothing: antialiased;
19   -moz-osx-font-smoothing: grayscale;
20 }
21
22 ul, ol, li {
23   list-style: none;
24 }
25
26 .wrapper{
27   min-width: 100%;
28 }
29
30

```

Рис 3.9 Код для скидання стандартних стилів

Для оптимізації зовнішнього вигляду `normalize.css`, яка дозволяє забезпечити єдине стилістичне відображення елементів у різних браузерах.

Семантична розмітка HTML включає базові теги `header`, `main`, `nav`, `div`, `img`, `h1`, `p`, `button`, що відповідає стандартам сучасної веб-розробки та сприяє SEO-оптимізації.

Стилізація виконана за допомогою каскадних таблиць стилів (CSS), що дозволило задати конкретні параметри для кожного елемента: колір, розмір шрифту, інтервали, положення блоків, а також елементи анімації та взаємодії при наведенні курсору.

Особливу увагу при розробці було приділено елементам взаємодії з користувачем. Наприклад, кнопка «Замовити зараз» змінює свій колір при наведенні, що виконує функцію зворотного зв'язку та демонструє інтерактивність інтерфейсу.

Шрифти, кольори та відступи підібрані з урахуванням принципів зручності читання, що підвищує доступність ресурсу для широкого кола користувачів.

Таким чином, реалізація основних елементів сайту базується на поєднанні інформативного наповнення, візуальної привабливості та технологічної ефективності.

Усі компоненти ресурсу — від структури до стилізації — спрямовані на створення позитивного досвіду взаємодії користувача з інтернет-платформою. Застосовані підходи демонструють можливості сучасних вебтехнологій у контексті ефективної онлайн-презентації розумних пристроїв нового покоління.

Коли я приступив до розробки цього проекту, то хотів створити не просто ще один лендінг, а інтерактивну платформу з плавними переходами та зручною навігацією. У цьому розділі я детально розберу ключові елементи коду, поясню свої архітектурні рішення та поділюся тими "фішками", які роблять цей сайт особливим

3.3 Скриншоти з поясненням коду

```
<div class="wrapper _container">
  <header class="header">
    <div class="header_container">
      <div class="logo">
        
      </div>
      <nav class="header__menu menu">
        <a href="" class="menu__link">SmartHub Air</a>
        <a href="" class="menu__link">Варіації пристрою</a>
        <a href="" class="menu__link">Відгуки</a>
      </nav>
    </div>
  </header>

  <main class="page swiper">
    <div class="swiper-wrapper">
```

Рис.3.10 HTML розмітка для навігаційного меню

На цьому скриншоті ми бачимо реалізацію навігаційного меню, створеного у вигляді звичайної «шапки» сайту. Меню не буде перекидувати нас на інші сторінки, так як це односторінковий сайт. Тому він буде нас перекидувати до вибраної нами секції на сайті, є дуже зручним рішенням для подібної платформи.

1. На сайті використовуються теги: `header`, `div`, `nav` та `img` для вставлення логотипу в шапку сайту
2. `Header` – стандартний тег для визначення шапки сайту, використовується покращення SEO.
3. `Div` – використовувався для завернення контенту в блок для кращої маніпуляції в CSS
4. `Nav` – використовувався оптимізації SEO та для зручного використання елементів в JavaScript для реалізації робочого навігаційного меню

`Img` тег для втсавлення фото на сайт, в нашому випадку – в шапку сайту

```

let menuLinks = document.querySelectorAll('.menu__link');

function menuSlider() {
  if (menuLinks.length > 0) {
    menuLinks[pageSlider.realIndex].classList.add('_active');
    for(let index = 0; index < menuLinks.length; index++) {
      const menuLink = menuLinks[index];
      menuLink.addEventListener("click", function(e){
        menuSliderRemove();
        pageSlider.slideTo(index, 800);
        menuLink.classList.add('_active');
        e.preventDefault();
      });
    }
  }
}

function menuSliderRemove() {
  let menuLinkActive = document.querySelector('.menu__link._active');

  if(menuLinkActive){
    menuLinkActive.classList.remove('_active');
  }
}

```

Рис. 3.11 Реалізація навігаційного меню

Цей фрагмент коду відповідає за синхронізацію навігаційного меню зі слайдами нашого сайту. Ми знаходимо всі посилання меню за допомогою класу `.menu__link`. Це дає нам колекцію елементів, з якими ми будемо працювати.

Функція починається з перевірки, чи взагалі існують елементи меню. Це важливо, щоб уникнути помилок, якщо меню на сторінці немає.

При кліку на пункт меню:

- Викликаємо `menuSliderRemove()` (функція, яка видаляє клас `_active` з усіх пунктів)
- Переходимо до відповідного слайду за 1000 мс
- Додаємо клас `_active` до поточного пункту меню
- Відміняємо стандартну поведінку посилання

Чому це важливо?

Цей код забезпечує:

1. Двосторонній зв'язок між меню і слайдером
2. Плавні переходи між розділами
3. Візуальне підсвічування поточного розділу
4. Зручну навігацію для користувачів

Проблеми, які ми вирішили:

1. Без `realIndex` в режимі `loop` активний пункт меню міг визначатися неправильно
2. Без `preventDefault()` сторінка могла "стрибати" при кліку на посилання
3. Без видалення попереднього активного класу могло бути кілька активних пунктів одночасно

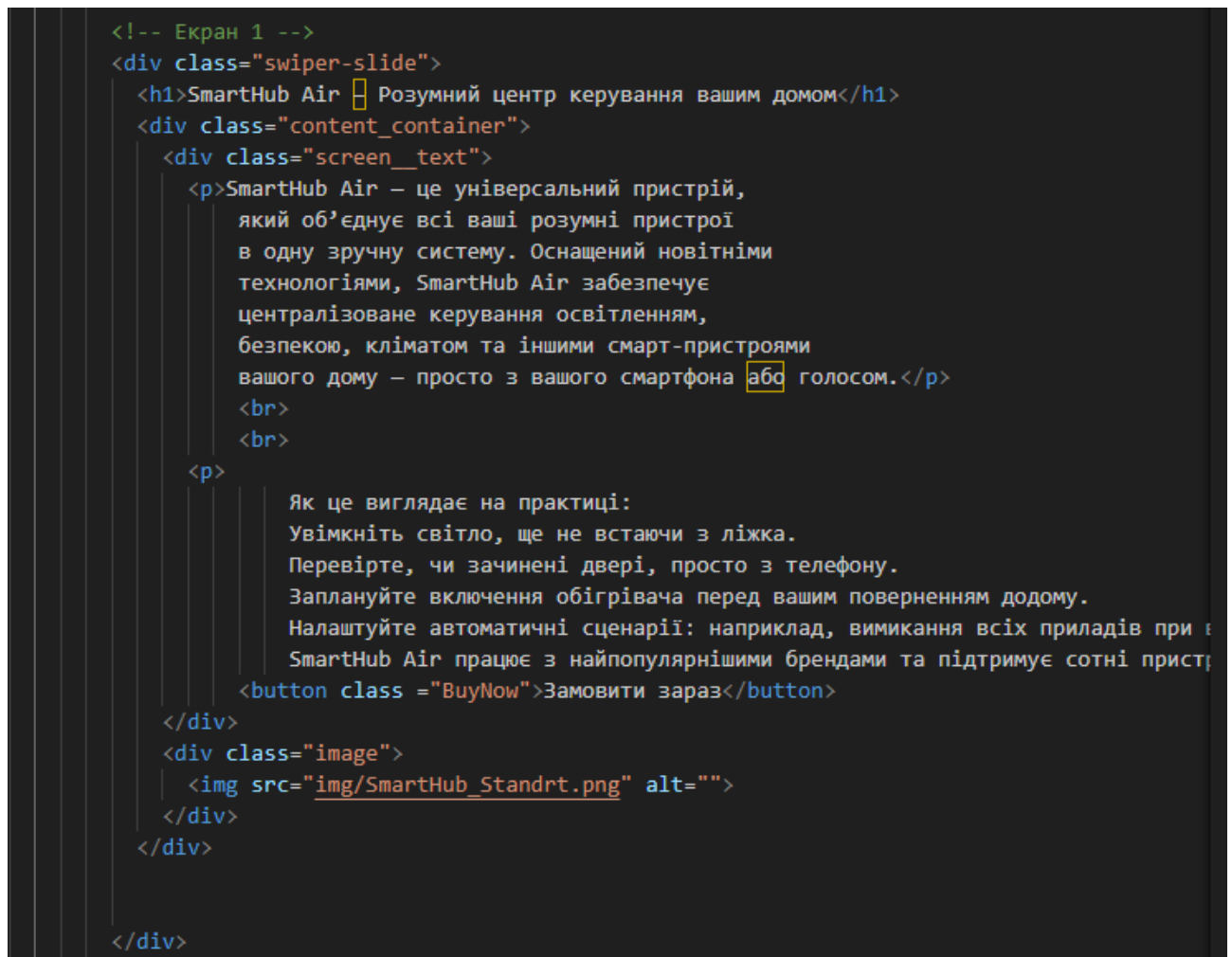


Рис 3.12 Реалізація головного екрану

На цьому скриншоті показаний екран в якому є фото та опис розумного пристрою в мінімалістичному стилі. Додана кнопка «Купити зараз», за

допомогою тегу `<button>`, використовувались теги `<br>` для переносу тексту на новий рядок та `<h1>` для заголовку сторінки

```

<!-- Екран 2 з вкладеним Swiper -->
<div class="swiper-slide">
  <h1>Варіації пристрою</h1>
<div class="text_variation_container">
  <div class="slide-text active" data-slide="0">
    <p>Це базова версія, яка забезпечує повну інтеграцію більшості смарт-пристроїв у вашому домі. SmartHub Air Standard підтримує голос
  </div>
  <div class="slide-text" data-slide="1">
    <p>SmartHub Air Mini
    Компактний контроль — максимальна зручність.
    SmartHub Air Mini — це ідеальне рішення для невеликих приміщень, квартир або для тих, хто лише починає занурення у світ розумного дому.
  </div>
  <div class="slide-text" data-slide="2">
    <p>SmartHub Air Pro — флагманська версія для тих, хто хоче отримати повний контроль над усім будинком. Підтримка до 300 пристроїв,
  </div>

<!-- Swiper -->
<div class="inner-swiper swiper">
  <div class="swiper-wrapper">
    <div class="swiper-slide"></div>
    <div class="swiper-slide"></div>
    <div class="swiper-slide"></div>
  </div>
  <div class="swiper-button-prev"></div>
  <div class="swiper-button-next"></div>
  <div class="swiper-pagination"></div>
</div>
</div>

```

Рис. 3.13 Реалізацію другого екрану з вкладеним Swiper

На цьому скриншоті реалізований екран в якому є вкладений свайпер, пагінація, та стрілки керування. Ще на цьому екрані є опис, який буде змінюватись при гортанні варіації пристрою

```

},
on: {
  slideChange: function () {
    const allTexts = document.querySelectorAll('.slide-text');
    allTexts.forEach(text => text.classList.remove('active'));

    const activeText = document.querySelector(`.slide-text[data-slide="${this.activeIndex}"]`);
    if (activeText) {
      activeText.classList.add('active');
    }
  }
}

```

Рис 3.14 Реалізація зміни опису при свайпі

На цьому скриншоті показано реалізацію зміни тексту при свайпі варіацій розумного пристрою, що дозволяє економити місце на сайті.

```

html, body {
    margin: 0;
    height: 100%;
    overflow: hidden;
    font-family: sans-serif;
}

[class*="_container"]{
    max-width: 1440px;
    margin: 0px auto;
    padding: 0px 0px;
}

.wrapper {
    height: 100%;
}

```

Рис. 3.15 Розширення секції на повний екран

На цьому слайді показаний стилізація сайту CSS, яка розширює видиму частину екрану на весь екран по висоті, що дозволяє адаптивно відображати коректно екрани на майже всіх настільних пристроях.

```

// Обробка форми замовлення
document.getElementById('orderForm').addEventListener('submit', function(e) {
    e.preventDefault();

    // Тут можна додати логіку відправки форми на сервер
    const formData = {
        name: document.getElementById('name').value,
        phone: document.getElementById('phone').value,
        email: document.getElementById('email').value,
        model: document.getElementById('model').value,
        quantity: document.getElementById('quantity').value,
        address: document.getElementById('address').value,
        comments: document.getElementById('comments').value
    };

    console.log('Дані форми:', formData);
    alert('Ваше замовлення прийнято! Ми зв\'яжемося з вами найближчим часом.');
```

Рис. 3.16 Обробка форми замовлення

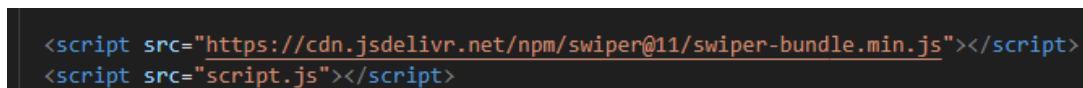
На цьому скриншоті показано обробку форми намовлення для секції з замовленням розумного пристрою, я вважаю це необхідним рішенням для того щоб всі данні які надходять від користувачів, зберігались в змінних для подальших процедур.

3.4 Як Swiper.js інтегровано в платформу

Коли я вперше побачив Swiper.js у дії на чужих сайтах, то відразу зрозумів – саме ця бібліотека стане "двигуном" мого проекту.

Розповім про всі етапи інтеграції – від базового підключення до тонких налаштувань, які зробили мій сайт справді особливим.

Бібліотека Swiper.js стала основним інструментом для реалізації інтерактивної навігації на нашому сайті. Розглянемо ключові аспекти її інтеграції.



```
<script src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-bundle.min.js"></script>
<script src="script.js"></script>
```

Рис. 3.17

На цьому скриншоті додалено два скрипта, за допомогою якого і можливе підключення Swiper.js так як він працює по API принципу. Перший скрипт це посилання на API бібліотеки, а друге це файл з налаштуваннями, яке показано на [Рис.3.18]

```

let pageSlider = new Swiper('.page', {
  direction: 'vertical',
  speed: 1000,
  watchOverflow: true,

  keyboard: {
    enabled: true,
    onlyInViewport: true,
    pageUpDown: true,
  },

  mousewheel: {
    sensitivity: 2,
  },

  pagination: {
    el: '.page_pagination',
    type: 'bullets',
    clickable: true,
    bulletClass: "page_bullet",
    bulletActiveClass: "page_bullet_active",
  },

  scrollbar: {
    el: '.page_scroll',
    draggable: true,
    dragClass: "page_drag-scroll",
  },

  parallax: true,
  observer: true,
  observerParents: true,
  observeSlideChildren: true,

  init: false,

  on: {
    init: function () {
      menuSlider();
    },
    slideChange: function () {
      menuSliderRemove();
      menuLinks[pageSlider.realIndex].classList.add('_active');
    }
  }
});

```

Рис.3.18 Ініціалізація повноекранного скролу.

Вертикальний скрол як основа навігації

Я свідомо обрав вертикальний скрол замість традиційного горизонтального, оскільки він:

1. Створює ефект "розгортання" контенту
2. Надає сайту преміальний вигляд
3. Дозволяє краще контролювати послідовність перегляду

Швидкість анімації підібрана емпіричним шляхом - 1000 мс виявилося ідеальним балансом між плавністю та швидкістю реакції.

Розширені можливості керування

Для зручності користувачів реалізовано кілька альтернативних способів керування:

1. Клавіатура - стрілки ввєрх/вниз
2. Миша - колесо прокрутки з оптимальною чутливістю
3. Тач-скрол на мобільних пристроях
4. Кастомний скроллбар з можливістю перетягування

Інтерактивна пагінація

Стандартні точки навігації замінено на стилізовані елементи, які:

1. Чітко відображають поточну позицію
2. Реагують на наведення
3. Дозволяють швидко переходити до будь-якого розділу

Вкладені слайдери для контенту

Для демонстрації різних варіантів продукту використано вкладені горизонтальні слайдери, які:

1. Працюють незалежно від основного скролу
2. Мають власні навігаційні елементи
3. Плавнo анімовані
4. Оптимізовані для тач-управління

Синхронізація з меню

Реалізовано двосторонній зв'язок між головним меню та слайдером:

1. Активний пункт меню автоматично оновлюється
2. Клік по меню переходить до відповідного розділу
3. Підсвічування допомагає орієнтуватись у контенті
4. Адаптація під різні пристрої
5. Особлива увага приділена:
6. Оптимальній роботі на тач-пристроях

7. Коректному відображенню на екранах різних розмірів
8. Плавній роботі при зміні орієнтації екрана

Оптимізація продуктивності

Для забезпечення плавної роботи:

1. Використано механізм лінивого завантаження
2. Обмежено кількість одночасно завантажуваних елементів
3. Застосовано техніки для мінімізації перерозрахунків

Цей підхід до інтеграції Swiper.js дозволив створити сучасну, зручну та продуктивну систему навігації, яка покращує загальний користувацький досвід на нашій платформі.

ВИСНОВОК

У ході виконання даної дипломної роботи було розроблено сучасну інтернет-платформу для продажу розумного пристрою з використанням технологій односторінкового додатку (SPA). Основна мета роботи - створення зручного, адаптивного та візуально привабливого інтерфейсу - була успішно досягнута.

Головним інструментом реалізації стала бібліотека Swiper.js, яка дозволила створити ефективну систему повноекранного вертикального скролу з плавними переходами між розділами. Особливу увагу було приділено оптимізації користувацького досвіду (UX), що включало:

1. Реалізацію інтуїтивної навігації
2. Адаптивний дизайн для різних пристроїв
3. Високий рівень інтерактивності елементів
4. Легка в інтеграції на сайт
5. Великі можливості кастомізації
6. Безліч сценаріїв використання
7. Регулярна підтримка бібліотеки
8. Велика та докладна документація

Технічна частина проекту базувалася на сучасному підході до веб-розробки з використанням семантичної HTML-розмітки, CSS3 для стилізації та чистого JavaScript для логіки роботи. Особливістю реалізації стало використання вкладених слайдерів для демонстрації різних варіантів продукту.

Практична цінність роботи полягає у створенні готового рішення, яке може бути використане для ефективної презентації та продажу розумних пристроїв. Запропонований підхід до розробки довів свою ефективність і може бути застосований для подібних проектів у майбутньому.

HTML Пропонує безліч тегів на всі випадки, вони як за логічним сенсом підходять до написання, так і оптимізують SEO, що є корисним для просування

сайту в пошукових системах, сподобалась ієрархічна система розташування тегів.

Було важливим навчитись створювати та реалізовувати проект з нуля, вчитись на помилках та розвивати ідеї які приходили до мене, скористатись різними технологіями так стеками, Swiper.js показала себе як досить потужна бібліотека, за допомогою якої можна не тільки створити галерею товару, або варіацію одного товару, а й використати як повноцінний повноекранний слайдер на весь сайт що є досить вражаючим результатом.

Окремо було корисно проектувати інтернет платформу не тільки з точки зору архітектури сайту, а і з точки зору дизайну. CSS в наш час дає можливість реалізувати будь-які фантазії дизайнерів, зараз CSS може не тільки «Красити кнопки» а навіть самостійно створювати анімації і все це лише за допомогою таблиці стилів, що теж є досить вражаючим.

Окремо можна сказати за адаптивність, яка досягається лише декількома медіа-запитами та зміни стилів вже всередині запита, це не важке рішення, але дає багато користі та забезпечує великий охоплення користувачів для сайту і дозволяє користуватись не лише настільними ПК, а й мобільними пристроями, такі як смартфон, планшет.

JavaScript. Він має безкінечний потенціал як і всі мови програмування, для мене реалізація слайдерів на ньому було самим задоволенням, легкий синтаксис, необов'язкові крапка з комой, безліч документації робить цю мову програмування дуже корисною, затребуваним та найпопулярнішою серед новачків і не тільки. На сайті за допомогою JavaScript було інтегровано слайдер як повноекранний так і вкладений, було створене навігаційне меню, та динамічний опис продукту при зміні варіації пристрою

Таким чином, всі поставлені завдання були успішно виконані, що підтверджує досягнення мети дипломної роботи. Отримані результати демонструють можливість створення сучасних, зручних та ефективних інтернет-рішень з використанням актуальних веб-технологій.

ПЕРЕЛІК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Белякова В. Где применяется JavaScript: от веб-разработки до мобильных приложений. Популярные фреймворки и ключевые особенности языка программирования JavaScript. *Free Online Programming Courses: HTML, CSS, JavaScript | CodeBasics*. URL: https://codebasics.com/ru/blog_posts/javascript-gde-ispolzuesya-i-kak-rabotaet.
2. CSS: Cascading Style Sheets | MDN. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
3. Flanders C. Understanding the Basics of UI/UX Design: A Beginner's Guide. *Medium*. URL: <https://medium.com/design-bootcamp/understanding-the-basics-of-ui-ux-design-a-beginners-guide-11752310317a>.
4. Hashemi-Pour C., Churchville F. What Is a User Interface (UI)? | Definition from TechTarget. *Search App Architecture*. URL: [https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI#:~:text=The%20user%20interface%20\(UI\)%20is,the%20appearance%20of%20a%20desktop](https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI#:~:text=The%20user%20interface%20(UI)%20is,the%20appearance%20of%20a%20desktop).
5. HTML: HyperText Markup Language | MDN. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
6. JavaScript | MDN. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
7. Swiper Slider. Установка и настройка - Как создать сайт. *Как создать сайт*. URL: <https://webdiz.com.ua/swiper-slider-ustanovka-i-nastroyka/>.
8. Swiper - The Most Modern Mobile Touch Slider. *Swiper*. URL: <https://swiperjs.com/>.
9. What is User Experience (UX) Design? – updated 2025. *The Interaction Design Foundation*. URL: <https://www.interaction-design.org/literature/topics/ux-design?srltid=AfmBOoq8gp9-QtiTmIIg8yQ0awWgte6Tv4TmkmrI6LShNWisrlndhe1N>.

10. Wilkins J. 40 JavaScript Projects for Beginners – Easy Ideas to Get Started Coding JS. *freeCodeCamp.org*.
URL: <https://www.freecodecamp.org/news/javascript-projects-for-beginners/>.
11. Информационная UX-архитектура сайта: как создавать в 2025. *HighTime Media*. URL: <https://hightime.media/ux-arkhitektura-sayta/>.
12. Alici E. What is Swiper and how does it work?. *Medium*.
URL: <https://medium.com/@alicienes99/what-is-swiper-and-how-does-it-work-e729645ed5e8>.
13. Applying the principles of UX design to your website. *Brand is a Verb*.
URL: https://www.brandisaverb.org/applying-the-principles-of-ux-design-to-your-website?gad_source=1&gad_campaignid=13906902408&gbraid=0AAAAABtSRZL0nq3cDHS9BtmecVVZ46q6g&gclid=Cj0KCQjwiqbBBhCAARIsAJSfZkZiox_t0CAqR0amifA_dyIUi609jSPLv4SEMi0O9u_RBnZ4dWeycHEaAqBiEALw_wcB.
14. Best Architectural Design Websites. *Awwwards - Website Awards - Best Web Design Trends*. URL: <https://www.awwwards.com/websites/architecture/>.
15. Contributors to Wikimedia projects. Adaptive website - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Adaptive_website.
16. CSS Tutorial - GeeksforGeeks. *GeeksforGeeks*.
URL: <https://www.geeksforgeeks.org/css-tutorial/>.
17. Додаток 3/4. Нормативні документи. Додаток 3/4. Нормативні документи. Додаток 3/4. Нормативні документи. *Ranktracker: The all-in-one platform for effective SEO*.
URL: <https://www.ranktracker.com/uk/seo/glossary/site-architecture/#:~:text=Архітектура%20сайту%20-%20це%20структура%20та,вони%20пов'язані%20між%20собою>.
18. Hashemi-Pour C., Churchville F. What Is a User Interface (UI)? | Definition from TechTarget. *Search App Architecture*.
URL: <https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI>.

- 19.Migrating from ion-slides to Swiper.js | Ionic Framework. *Ionic Framework - The Cross-Platform App Development Leader*.
URL: <https://ionicframework.com/docs/angular/slides>.
- 20.Responsive Design vs. Adaptive Design: What's the Best Choice for Designers?. *Studio* by *UXPin*.
URL: <https://www.uxpin.com/studio/blog/responsive-vs-adaptive-design-whats-best-choice-designers/>.
- 21.Swiper JS: Ultimate Guide to Responsive Touch Slider for Websites. *The IoT Academy Blogs - Best Tech, Career Tips & Guides*.
URL: <https://www.theiotacademy.co/blog/swiper-js/>.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Презентація до кваліфікаційної роботи

на тему:

«ІНТЕРНЕТ ПЛАТФОРМА ДЛЯ ПРОДАЖУ «РОЗУМНОГО ПРИСТРОЮ» ІЗ
ВИКОРИСТАННЯ SWIPER»

Виконав: Здобувач вищої освіти гр.ІСД-41

Юрій МІЛЕНЬКИЙ

Керівник: к.т.н доцент каф. ІСТ

Олег СЕНЬКОВ

МЕТА ТА ЗАВДАННЯ

Мета роботи – розробка адаптивної та оптимізованої інтернет-платформи для демонстрації та продажу розумного пристрою з використанням бібліотеки Swiper.js, з акцентом на зручний та привабливий інтерфейс.

Інтернет платформа розрахована на користувачів та покупців яким більш привабливий мінімалізм, привабливий більш зручний формат замовлення у вигляді форми

Для досягнення мети виконуються такі завдання:

1. Провести аналіз сучасних технологій розробки веб-платформ;
2. Дослідити можливості бібліотеки Swiper.js та сфери її застосування;
3. Спроектувати архітектуру інтернет-платформи;
4. Реалізувати основний функціонал сайту з використанням Swiper.js;
5. Провести тестування платформи та оцінити її зручність

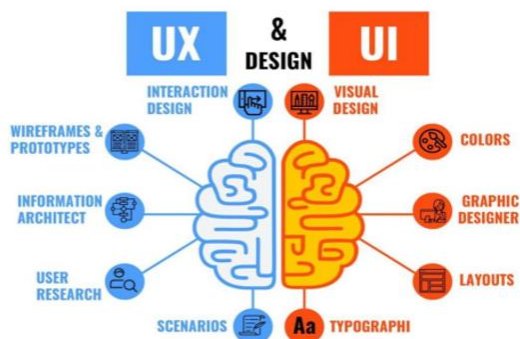
ТЕОРИТИЧНІ ОСНОВИ СТВОРЕННЯ ІНТЕРНЕТ ПЛАТФОРМИ

Для створення односторінкової інтернет платформи було використано базовий стек технологій, якими є HTML/CSS/JavaScript. Незважаючи на простоту, комбінація цих інструментів дозволяє реалізувати рішення, що відповідає сучасним вимогам до веб – розробки.



3

ОСОБЛИВОСТІ UX/UI ДЛЯ ОНЛАЙН-ПРОДАЖІВ РОЗУМНИХ ПРИСТРОЇВ

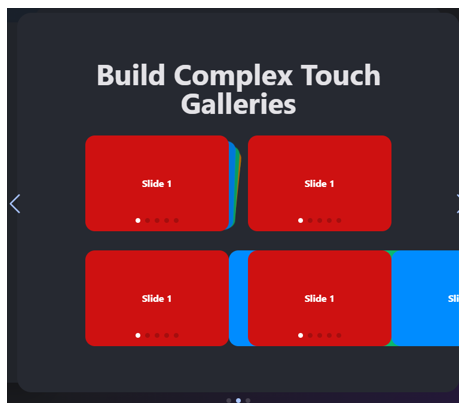


- UI (User Interface) це інструмент, через який користувачі будуть взаємодіяти із сайтом. Саме завдяки UI ми формуємо перше враження про сайт та пристрій.
- UX (User Experience) Це загальний досвід, який користувач отримує під час користування сайтом, це сукупність вражень, зручності, які користувач відчуває при взаємодії з сайтом

4

ЩО TAKE SWIPER.JS: ЙОГО ПРИЗНАЧЕННЯ, МОЖЛИВОСТІ, ПРИКЛАДИ ВИКОРИСТАННЯ

Swiper.js – це сучасна безкоштовна JavaScript-бібліотека для творення динамічних елементів, такі як слайдер, повноекранні презентації, галереї, на вебсайтах, веб-додатках, мобільних додатках. Swiper.js є одною з найпопулярніших бібліотек у сфері веб-розробки.



5

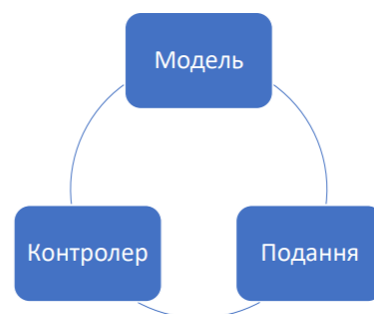
ПРОЕКТУВАННЯ ІНТЕРНЕТ ПЛАТФОРМИ

У нашій системі застосовується принцип MVC (Model-View-Controller) у спрощеному вигляді.

Модель (Model) — структура даних, наприклад, інформація про товар, ціни, характеристики.

Подання (View) — візуальні елементи, що відображаються користувачу (слайдери, тексти, кнопки).

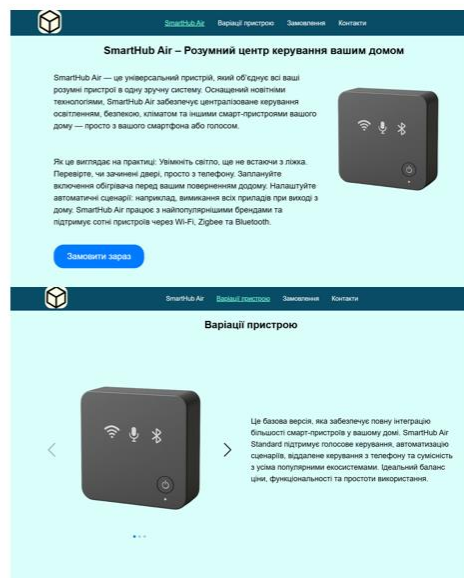
Контролер (Controller) — JavaScript-логіка, яка реагує на дії користувача, наприклад, свайпи, натискання кнопок, прокрутка.



6

РЕАЛІЗАЦІЯ ІНТЕРНЕТ ПЛАТФОРМИ

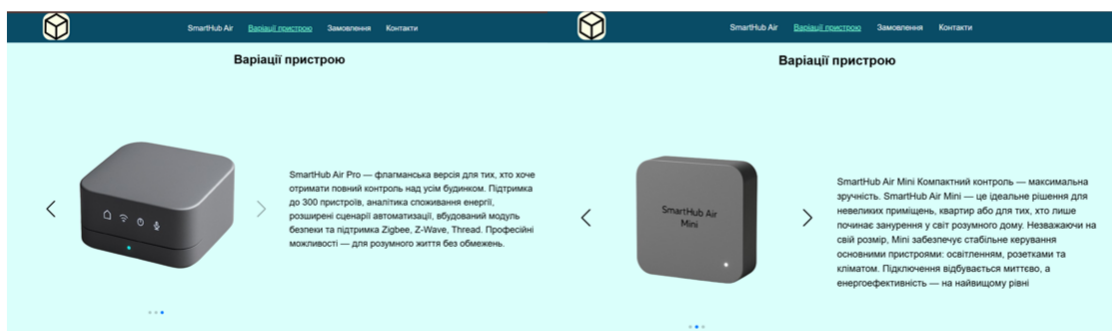
Для початку було створено скелет сайту за допомогою HTML, розділено на декілька рівних секцій. Кожна секція буде нести окремі розділи інформації. Після створення розмітки, всі секції було стилізовано за допомогою CSS



7

РЕАЛІЗАЦІЯ ІНТЕРНЕТ ПЛАТФОРМИ

Для більшої зручності та економії місці на сайті, було додано карусель з вибором варіації пристрою



8

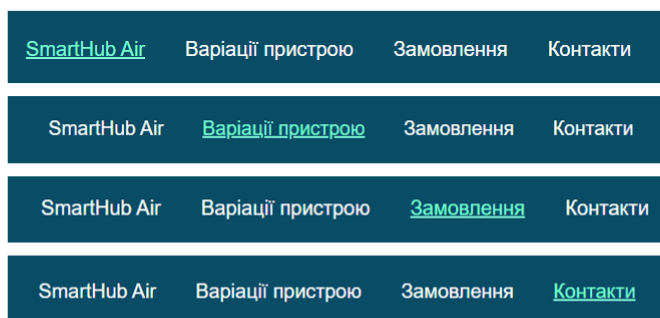
РЕАЛІЗАЦІЯ ІНТЕРНЕТ ПЛАТФОРМИ

- Було вирішено додати саме форму для замовлення товару. Що спростило б та автоматизувало процес оформлення замовлення.
- Було додано контактну інформацію окремою секцією

The image displays two screenshots of the SmartHub website interface. The top screenshot shows the 'Замовити SmartHub' (Order SmartHub) form, which includes fields for 'Ім'я' (Name), 'Телефон' (Phone), 'Email', 'Місто' (City), 'Область/інша' (Region/other), 'Код пошти' (Postal code), 'Адреса доставки' (Delivery address), and 'Коментар' (Comment). A blue button labeled 'Замовити SmartHub' is at the bottom of the form. The bottom screenshot shows the 'Наші контакти' (Our contacts) section, which includes contact information such as 'Адреса' (Address), 'Телефон' (Phone), 'Email', and 'График роботи' (Working hours), along with a map titled 'Ми в соціальних мережах' (We are in social networks).

РЕАЛІЗАЦІЯ ІНТЕРНЕТ ПЛАТФОРМИ

Функціональними рішеннями стало повноекранні скроли між цими секціями, та реалізація навігаційного меню, яким можна користуватись щоб одразу переміститись на потрібну секцію, а підкреслює поточну секцію



10

ВИСНОВКИ

У ході виконання бакалаврської роботи було розроблено сучасну інтернет платформу для продажу розумного пристрою з використанням технологій односторінкового додатку. Основна мета роботи - створення зручного, адаптивного та візуально привабливого інтерфейсу - була успішно досягнута. Головним інструментом реалізації стала бібліотека Swiper.js, яка дозволила створити ефективну систему повноекранного вертикального скролу з плавними переходами між розділами. Особливу увагу було приділено оптимізації користувацького досвіду (UX)

11

АПРОБАЦІЯ

- VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» 15 квітня 2025 року, ДУІКТ – «АНАЛІЗ РІШЕНЬ ДЛЯ СТВОРЕННЯ ІНТЕРАКТИВНИХ ЕЛЕМЕНТІВ КАРУСЕЛЬ АБО СЛАЙДЕР НА ВЕБ-СТОРІНКАХ»