

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ  
ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка веб-сайтів з використанням 3D моделей»

на здобуття освітнього ступеня бакалавра

зі 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)

Данило Буджак  
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр.ІСД-41

Данило БУДЖАК

Ім'я, ПРІЗВИЩЕ

Керівник:

науковий ступінь,  
вчене звання

Олександр КІС

Ім'я, ПРІЗВИЩЕ

Рецензент:

науковий ступінь,  
вчене звання

\_\_\_\_\_  
Ім'я, ПРІЗВИЩЕ

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**  
Кафедра Інженерія програмного забезпечення автоматизованих систем  
Ступінь вищої освіти бакалавр  
Спеціальність 126 Інформаційні системи та технології  
Освітньо-професійна програма Інформаційні системи та технології

**ЗАТВЕРДЖУЮ**

Завідувач кафедру ІПЗАС

Каміла СТОРЧАК

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ  
Буджак Данило Віталійович**

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Розробка веб-сайтів з використанням 3D моделей

керівник кваліфікаційної роботи Олександр КІС,

*(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36

2. Строк подання кваліфікаційної роботи «31» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: Готовий веб-сайт з імплементованими інтерактивними 3D моделями .

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

1. Аналіз правила для створення UI/UX дизайну

2. Розробка коду веб-сату

3. Імплементация 3D об'єкти на веб-сторінку

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання: «27» лютого 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обробка теоретичної частини	27.02-05.03.2024	
2	Написання першого розділу	06.03-10.03.2024	
3	Написання другого розділу	11.03-26.03.2024	
4	Написання третього розділу	27.03-10.04.2024	
5	Оформлення випускної роботи, обґрунтування результатів дослідження	10.04-22.04.2024	
6	Підготовка демонстраційних матеріалів	25.04-05.05.2024	
7	Попередній захист роботи	27.05.2024	
8	Пред'явлення роботи в деканат		

Здобувач(ка) вищої освіти

\_\_\_\_\_

(підпис)

Данило БУДЖАК

(Ім'я, ПРІЗВИЩЕ)

Керівник  
кваліфікаційної роботи

\_\_\_\_\_

(підпис)

Олександр КІС

(Ім'я, ПРІЗВИЩЕ)

## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 55 стор., 58 рис., 27 джерел.

*Мета роботи* – Метою цієї дипломної роботи є дослідження та аналіз використання 3D моделей на веб-сайтах з подальшою розробкою веб-сторінки

*Об'єкт дослідження* – Розробка веб-сайтів з використанням 3D моделей

*Предмет дослідження* – Веб-сайт та тривимірні об'єкти

*Короткий зміст роботи:*

У цій дипломній роботі буде проведено комплексне дослідження методів та інструментів використання 3D-моделей у веб-дизайні. Мета дослідження – визначити, як 3D-моделі можуть впливати на інтерактивність та візуальну привабливість веб-сайтів, а також розробити практичні рекомендації щодо їх ефективного використання.

В рамках дослідження буде проведено аналіз існуючих технологій 3D-моделювання та їх застосування у веб-дизайні. Буде вивчено вплив 3D-моделей на поведінку користувачів, їх сприйняття інформації та емоційний стан.

Для практичного дослідження буде розроблено власний веб-сайт з використанням 3D-моделей. На цьому сайті буде проведено тестування та аналіз впливу 3D-моделей на користувачів.

КЛЮЧОВІ СЛОВА:

3D-МОДЕЛІ, ВЕБ-ДИЗАЙН, ІНТЕРАКТИВНІСТЬ, ВІЗУАЛЬНА ПРИВАБЛИВІСТЬ, КОРИСТУВАЦЬКИЙ ДОСВІД, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, REACT, ХУКИ, ІМПЛЕМЕНТАЦІЯ, АНІМАЦІЯ, CSS, JSX.

## ABSTRACT

Text part of the master`s qualification work: 55 pages, 58 pictures, 27 sources.

*The purpose of the work* - The purpose of this thesis is to research and analyze the use of 3D models on websites with the subsequent development of a web page

*Object of research* - Website development using 3D models

*Subject of research* - Website and three-dimensional objects

*Summary of work:*

This thesis will conduct a comprehensive study of methods and tools for using 3D models in web design. The purpose of the research is to determine how 3D models can influence the interactivity and visual appeal of websites, as well as to develop practical recommendations for their effective use.

The study will analyze existing 3D modeling technologies and their application in web design. The impact of 3D models on user behavior, information perception, and emotional state will be studied.

For practical research, i will develop our own website using 3D models. This website will be used to test and analyze the impact of 3D models on users.

KEYWORDS:

3D MODELS, WEB DESIGN, INTERACTIVITY, VISUAL APPEAL, USER EXPERIENCE, USER INTERFACE, REACT, HOOKS, IMPLEMENTATION, ANIMATION, CSS, JSX.

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
Навчально-науковий інститут \_\_\_\_\_

**ПОДАННЯ**  
**ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ**  
**ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Направляється здобувач(ка) Буджак Д.В. до захисту кваліфікаційної роботи  
(прізвище та ініціали)

за спеціальністю 126 Інформаційні системи та технології  
(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології  
(назва)

на тему: «Розробка веб-сайтів з використанням 3D моделей».

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ \_\_\_\_\_  
(підпис) (Ім'я, ПРІЗВИЩЕ)

**Висновок керівника кваліфікаційної роботи**

Здобувач(ка)

---

---

---

---

---

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача(ки) \_\_\_\_\_  
на  
оцінку « \_\_\_\_\_ » та присвоїти йому (їй) кваліфікацію \_\_\_\_\_ .

Керівник кваліфікаційної роботи \_\_\_\_\_  
(підпис) (Ім'я, ПРІЗВИЩЕ)  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_

року

**Висновок кафедри про кваліфікаційну роботу**

Кваліфікаційна робота розглянута. Здобувач(ка) Буджак Д.В. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою \_\_\_\_\_  
(назва) (підпис) (Ім'я, ПРІЗВИЩЕ)

**ВІДГУК РЕЦЕНЗЕНТА**  
**на кваліфікаційну роботу**  
**на здобуття освітнього ступеня бакалавра**

здобувача(ки) вищої освіти Буджак Данила Віталійовича  
(прізвище, ім'я, по батькові)  
на тему: «Розробка веб-сайтів з використанням 3D моделей»

**Актуальність.**

Кваліфікаційна робота Буджак Данила Віталійовича на тему «Розробка веб-сайтів з використанням 3D моделей» є надзвичайно актуальною, адже використання 3D моделей на веб-сайтах стає все більш поширеним. Це пов'язано з тим, що 3D моделі дозволяють зробити веб-сайти більш привабливими, інформативними та інтерактивними.

**Позитивні сторони.**

1. Глибоке теоретичне обґрунтування теми.
2. Практична реалізація теоретичних знань на прикладі розробки веб-сайту.
3. Високий рівень володіння мовами програмування та інструментами для роботи з 3D моделями.

**Недоліки.**

1. Необхідно більш розгорнуто описати реалізацію UI інтерфейсів в коді React.
  2. Необхідно більш детально описати деякі аспекти реалізації проекту.
- Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи бакалаврської.

**Висновок:** *кваліфікаційна робота на здобуття ступеня бакалавра заслуговує оцінку «Відмінно», а здобувач Буджак Данило Віталійович заслуговує присвоєння кваліфікації: Бакалавр з інформаційних систем та технологій*

Рецензент:

науковий ступінь, вчене звання

\_\_\_\_\_

підпис

\_\_\_\_\_

Ім'я, ПРІЗВИЩЕ

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	9
ВСТУП.....	10
СТВОРЕННЯ UI/UX ДИЗАЙНУ.....	12
Що таке UI/UX принципи і концепції.....	12
Аналіз цільової аудиторії.....	15
1.3 Розробка Wireframe , Uikit та прототипізація.....	17
МЕТОДИ СТВОРЕННЯ 3D МОДЕЛЕЙ ТА ФУНКЦІЇ НА ВЕБ-САЙТАХ.....	30
2.1 Сучасні медоти створення тривімірних об'єктів.....	30
2.2 Функції тривімірних об'єктів на веб-сайтах.....	33
.3 Способи використання 3D елементів на сайті.....	35
Р	
Вибір технологій для написання програмного коду.....	37
3.2 Розробка програмного коду Fronend частини.....	41
Р	
3.3 Імплементация 3D моделі та її анімація.....	52
О	
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
Додаток А КОПІЇ ДЕМООНСТРАЦІЙНИХ МАТЕРІАЛІВ.....	68
Ф	
Р	
О	
Н	
Т	
Е	
Н	
Д	
Ч	
А	



## ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

D – three dimensional (тривимірний)

D – two dimensional (двовимірний)

UI – user interface (інтерфейс користувача)

UX – user experience (досвід користувача)

ЦА – цільова аудиторія

JS – Java Script

CSS – Cascad Style Shit (Каскадна Таблиця Стилів)

HTML – HyperText Markup Language (Мова гіпертекстової розмітки)

DOM – Document Object Model (Об'єктна модель документа)

## ВСТУП

**Актуальність теми.** Через стрімкий розвиток веб-технологій та зростання популярності 3D моделювання , завдяки спрощенню створення 3D моделей , їх використання на веб-сторінках стає все більш та більш актуальним . Це дозволяє зробити дизайн сайту більш динамічним , та більше зацікавити користувача для взаємодії з вашим сайтом або продуктом що продає компанія , що значно покращує досвід використання веб-сторінки.

Д

моделі. **Переваги** вебсайтах мають ряд переваг та недоліків :

1. Підвищення візуальної привабливості : 3D моделі роблять сайти більш привабливими та зацікавлюють користувачів більше ніж стандартні 2D картинки.
2. Покращення інтерактивності : 3D моделі можна анімувати та додавати функції масштабування , обертання , що дозволяє користувачам більше дослідити товари , або послуги що представлені на сайті , що по статистиці збільшує довіру до продукту та інтерес потенційних клієнтів до покупки .
3. Покращення сприйняття інформації : Трьох вимірні об'єкти дають змогу відвідувачам краще дослідити складні об'єкти та процеси на відміну від 2D зображень.

**Наукова новизна:** Досліджено вплив 3D графіки на веб-сайтах на користувачів , створено веб-сайт з унікальними 3D функціями

**Мета і завдання дослідження.** Метою цієї дипломної роботи є дослідження та аналіз використання 3D моделей на веб-сайтах з подальшою розробкою веб-сторінки для цього потрібно :

1. Провести аналіз технологій які потрібні для імплементації тривимірних об'єктів на сайт
2. Розробити UI/UX дизайн для вебсайт

3. Визначення технологій для написання Frontend частини вебсайту , та технології що будуть сумісні з інструментами для відображення 3D елементів.

4. Написання програмного коду.

**Об'єкт дослідження.** Розробка веб-сайтів з використанням 3D моделей

**Предмет дослідження.** Веб-сайт та тривимірні об'єкти

**Методи дослідження.** Під час проведення досліджень використовувалися наступні наукові методи:

- Аналіз літератури – Вивчення наукових статей , публікацій та веб-сайтів які присвячені теми «Розробка веб-сторінок з використанням 3D моделей»
- Порівняльний аналіз - порівняння різних форматів збереження 3D моделі для максимальної оптимізації сторінок сайту.
- Метод експериментального дослідження – розробка та тестування власного вебсайту присвяченого даній темі.

**Практичне значення одержаних результатів.** Результати цього дослідження можуть бути використані для :

1. Розробки значно ефективніших та привабливіших веб-сайтів .
2. Збільшення рівня зацікавленості користувачів , та зосередження їхньої уваги на тому що вибажаєте їм показати.
3. Розширення можливостей маркетингу для багатьох компаній та бізнесів.

**Апробація :**

І Всеукраїнська науково-технічна конференція "Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу" , 28 листопада 2023 року , ДУІКТ – «Імплементация 3D моделей на веб-сайт»

V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ» ,18 квітня 2024 року , ДУІКТ –

## СТВОРЕННЯ UI/UX ДИЗАЙНУ

### 1.1.Що таке UI/UX принципи і концепції

UI/UX дизайн інтерфейсу користувача та його досвіду – це процес побудови інтерфейсів , з якими можна взаємодіяти , що використовується в цифрових продуктах до прикладу – веб-сайти , мобільні додатки , програмне забезпечення та подібні.[1]

UI (User Interface ) – Інтерфейс користувача це візуальна частина певного продукту з якою взаємодіють користувачі.[1] До елементів UI відносять :

1. Позиціонування елементів
2. Шрифти
3. Кольори
4. Різні кнопки та елементи навігації
5. Підказки

UX(User Experience) – досвід користувача це те що користувач відчуває після використання цифрового продукту , він включає в себе не лише візуал , а й[1] :

- Простота використання
- Задоволеність
- Ефективність
- Наскільки легко досягти мету

Функції UI/UX дизайну :

1. Зрозумілість: Дизайн інтерфейсу повинен бути простим у використанні, щоб користувачі могли легко знайти та завершити завдання.
2. Ефективність: Користувачі повинні мати можливість виконувати свої завдання швидко та легко.
3. Привабливість: Інтерфейс має бути приємним для очей і відповідати бренду продукту.
4. Доступність: Інтерфейс повинен бути легко використовувати людям з обмеженими можливостями.

5. Довіра: Інтерфейс повинен створювати відчуття безпеки для користувачів.
6. Емоції: Користувачі повинні бути задоволені і задоволені інтерфейсом.

Концепції UI/UX дизайну[2] :

1. Централізація користувача: Дизайн повинен бути орієнтований на користувача, а не на замовника.
2. Інформаційна архітектура: Контент і навігація повинні бути логічними та простими для розуміння.
3. Візуальна ієрархія: Ви повинні візуально виділити найважливіші компоненти інтерфейсу.
4. Відгуки: Користувачі повинні отримувати швидкі та чіткі коментарі щодо своїх дій.
5. Простота: Інтерфейс повинен бути простим для користування та не перевантажувати.
6. Гнучкість: Інтерфейс повинен бути адаптований до різних типів пристроїв і користувачів.

UI/UX дизайн може бути використаний щоб:

- Щоб підвищити конверсію: Збільшення зручності інтерфейсу користувача може призвести до більш регулярних покупок товарів або послуг.
- Збільшити залучення користувачів: Якщо клієнти відчують задоволення від використання продукту, вони більш схильні повторно його використовувати.
- Зменшити витрати на підтримку: Інтуїтивно зрозумілий інтерфейс може допомогти зменшити кількість запитів, які отримує клієнтська служба.
- Зміцнити бренд: створити позитивний імідж бренду може допомогти привабливий і зручний інтерфейс.

Процес дизайну UI/UX[3]:

Дослідження: Дослідження потреб користувачів і цілей продукту є першим кроком у дизайні UI/UX. Це можна досягти за допомогою таких інструментів, як опитування, інтерв'ю, аналіз даних і тестування користувачів.

**Проектування:** Створюються прототипи та макети інтерфейсу на основі досліджень. Моделі інтерфейсу прототипів можна використовувати для тестування та збору відгуків від користувачів. Макети представляють більш детальні зображення інтерфейсу, які демонструють, як він буде виглядати візуально.

**Розробка:** Розробники починають розробляти інтерфейс після того, як прототипи та макети затверджені. Написання коду, створення графіки та тестування інтерфейсу є частиною цього процесу.

**Тестування:** Інтерфейс повинен бути ретельно протестований, щоб переконатися, що він працює правильно та відповідає вимогам користувачів. Можна проводити тестування за допомогою користувачів або автоматизованих інструментів.

**Запуск:** Після завершення тестування інтерфейс буде доступний для публіки.

**Підтримка:** Після запуску інтерфейсу важливо продовжувати збирати коментарі користувачів, щоб зробити відповідні зміни.

Хороша практика в UI/UX дизайні [4]

1. Адаптивність під мобільні пристрої – створення дизайну під різні розміри екрану , дає змогу залучити більшу кількість користувачі , адже вони уже поствлені в рамки того , яким пристроєм вони користуються
2. Оптимізація часу завантажень – тут все просто чим менший час завантаження додатку або вебсайту тим кращий буде користувацький досвід
3. Інтуєтивна навігація – потрібно зробити навігацію , що буде зрозумілою для користувача та простою у використанні .
4. Зворотній зв'язок – використання зворотнього зв'язку з ціллю збору від користувачів інформації

Також потрібно завжди тримати баланс між креативністю та функціональністю , адже це дуже тонка межа.

- Емоційний дизайн - потрібно , щоб дизайни викликав багато позитивних емоцій у користувача.
- Відповідність бренду : потрібно щоб дизайн був узгоджений з індивідуальністю бренду на приклад , кольори , мінімалістичний дизайн і тд.

- Тенденції в дизайні : потрібно знати , що зараз в тренді , а також географічні особливості , адже в різних країнах різна мода на дизайн.

## 1.2 Аналіз цільової аудиторії

Перш за все потрібно знати в якій тематиці буде розроблятися веб-сайт . Я обрав ітернет магазин кросівок , адже ця сфера дуже близька мені , та й на ділі , використовувати 3D моделі на вебсайтах з продажу кросівок , є досить , хорошою ідеєю , бо користувач цим компенсує недостаток інформації які є на картинка та це робить картинку більш реально , що дає змогу краще зрозуміти чи потрібен вам цей товар чи ні .

Що ж перейдемо до цільової аудиторії (ЦА). Вивчення та визначення ЦА досить сильно допомагає зрозуміти потреби і досвід користувачів, дозволяє зробити інтерфейс зручним і легким у використанні, що вплине на задоволення продуктом.[5]

Є декілька способів збору і впорядкування інформації для визначення цільової аудиторії:

1. 5 “W”. Цей спосіб найпоширеніший і складається з 5 основних питань які можуть змінюватися в залежності від галузі– What (Що продаємо?), Who (Кому продаємо?), Why (Чому купує саме тут?), Where (Де знаходиться місце продажу?), When (Коли продаємо?).

2. Google Analytics/SimilarWeb. Сервіси які дозволяють аналізувати інтернет-сайти. Можна дізнатись трафік, інтереси аудиторії, ознаки поведінки користувача та інші данні.

3. Анкетування або складання персон. Проведення інтерв'ю у людей, родичів, колег або запуск масового опитування. Цей спосіб складніший бо як мінімум потрібно опитати людей з різних вікових груп, статусів, інтересів.

4. Моніторинг соціальних мереж. Аналізувати підписників магазину який є обов'язково прямим конкурентом. Також, перегляд спільнот аналізуючи обговорення.

Для цього проекту я обрав метод 5W тож розглянемо його трохи детальніше.[6]

Who? (хто?) – повинно надавати основну інформацію про користувача . Чим більше ознак ЦА ми можемо назвати тим краще , адже це значно звужує коло тих людей на які орієнтовано певний продукт та тих людей яким це буде цікаво.

What? (що?) – Тут важливо враховувати не тільки те що ви продаєте або пропонуєте , але й те що користувачі планують від вас отримати.

Where? (Де?) – тут повинно бути вказане місце де ви контактуєте з клієнтами , здебільшого це певна локація

When? (Коли?) – це питання полягає в тому , коли користувачі готові користуватись вашими послугами , доприкладу деякі товари мають сезонність , а інші актуальні цілий рік.

Why? (Навіщо?) – це питання повинно визначити причину , що потенційні покупці повинні купити саме ваш товар.

Тож після того , як ми визначили ,концепцію цих запитань пропонуємо перейти до відповіді на них.

Who? – кому ми продаємо , це будуть люди віком від 16 до 35 років , адже це найпоширеніший вік серед любителів кросівок , здебільшого це будуть чоловіки приблизно (60%), дохід у них буде середній , бо ми не плануємо продавати дуже дорогі кросівки , адже для нас самих це не вигідно , також це здебільшого будуть люди що вже працевлаштувалися , це повинні бути люди , які звикли купувати онлайн та їх інтереси можуть бути спорт , мода та активний відпочинок.

What? – Що продаємо – це будуть кросівки світових брендів , адже це найлегше для початку , бо не потрібно планувати виробництво . Плануємо продавати бренди такі , як Nike , Adidas, Puma, Reebok, Asics.

Why? – чому саме ми ? Перше це низькі ціни , зручний вебсайт , та можливість обрати кросівки по 3D моделям які не відрізняються від товару , що продаємо.

Where? – де ми знаходимось ? Це онлайн магазин (веб-сайт).



Виходячи з цих питань можна підбити висновки того, на що потрібно звернути уваги при розробці UI - з першого питання ми дізнались що продає магазин і яка його тематика. Тобто, інтерфейс має бути лаконічний, мінімалістичний, щоб не забирало всю увагу від кросівок, та увага була зосереджена на 3Д-моделях.

### 1.3 Розробка Wireframe, Uikit та прототипізація

Wireframe – це початковий макет сторінки, який може бути в декількох варіантах, зазвичай має низький рівень деталізації. Показує де приблизно будуть розташовані основні блоки проекту, для подальшого уявлення досвіду користувача. [7]

З аналізу цільової аудиторії ми розуміємо що розробляється магазин кросівок, з цього виходить що сайт повинен бути більш зосереджений на кросівках та 3д моделі. Але 3д модель не повинна бути всюди щоб сайт не був перевантажений, бо це може відлякувати користувачі та дуже сильно впливає на оптимізацію сайту (він буде довше завантажуватися), тож ми розташуємо її по центру на головній сторінці для привертання уваги користувачів до покупки саме цієї нової моделі, яка додатково буде виділятися іншим кольором фону. (Рис. 1.1)

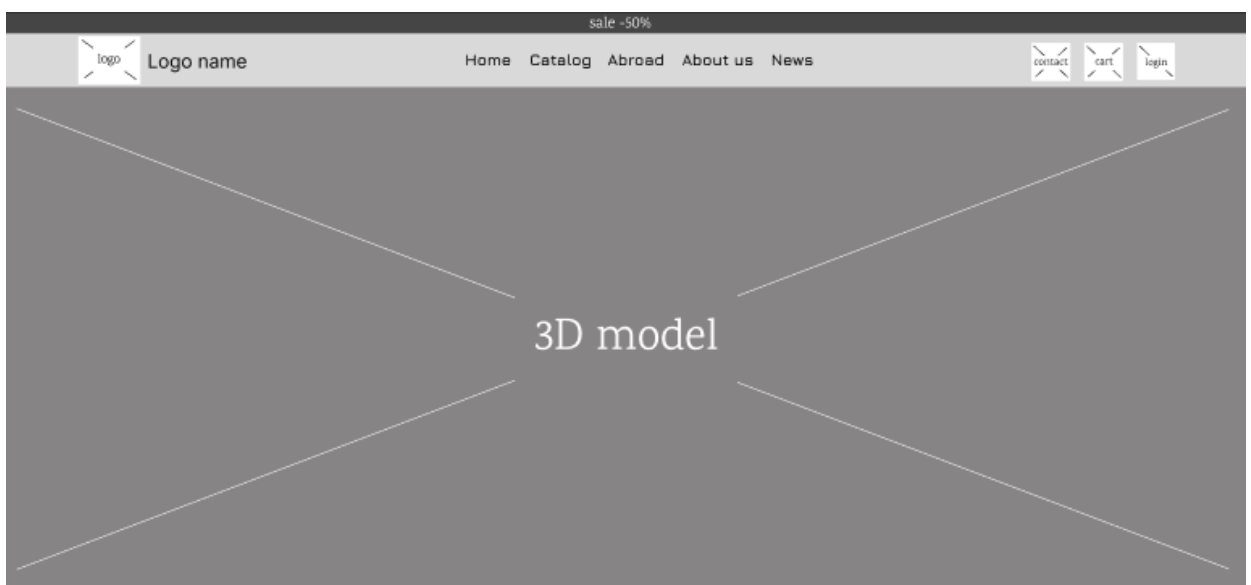


Рис. 1.1 Розташування 3D моделі

Далі по сітці зберігаючи основні інтервали між блоками додаємо основні блоки як кнопки бренду(розбиття каталогу під кожен бренд), Нові надходження(слайдер) та новини(слайдер). Також кожний блок трохи деталізуємо щоб розуміти як вони будуть виглядати в майбутньому(карточки товари додаємо іконки та зону на якій буде фото та інформація про кросівок, додаємо як буде виглядати слайдер та картка новин).(Рис. 1.2)

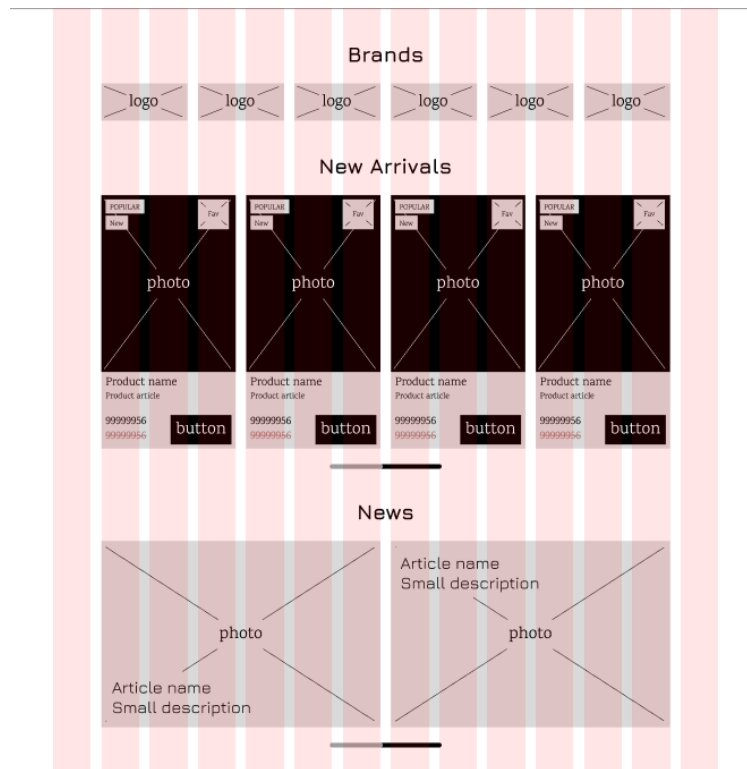


Рис. 1.2 Розсташування основних елементів

Деталізуємо footer та шапку сайту, пишемо можливий текст в футтері та в пункти меню додаємо іконки для контактної інформації пошуку і тд, що робить інтерфейс менш нагруденим текстом але водночас зрозумілим. Також до шапки додаємо блок зі знижками який постійно при скролі буде видний користувачу.(Рис.1.3)

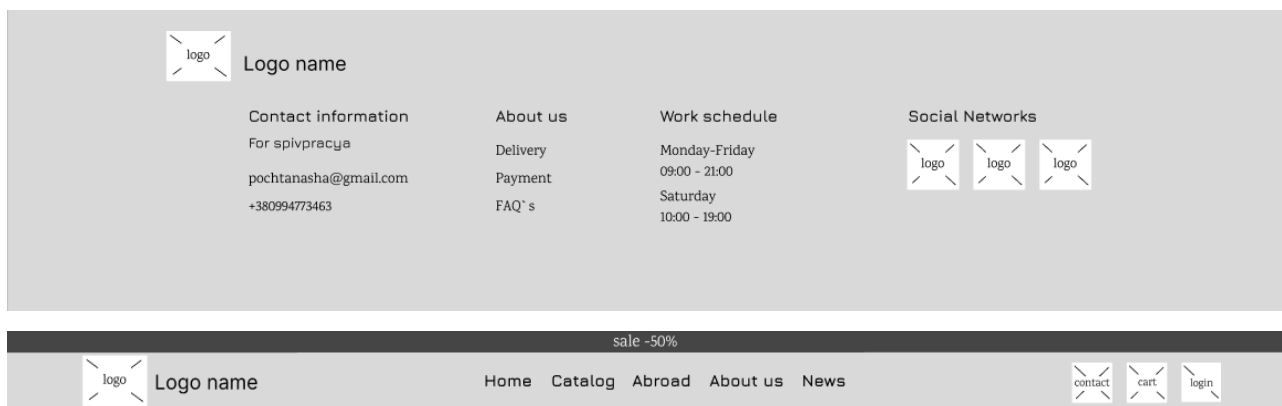


Рис. 1.3 Wireframe Menu та Footer

Використовуючи сітку робимо сторінку каталогу товару додаючи навігацію по сторінкам, розположення фільтру та вже розроблені картки товару.(Рис. 1.4)

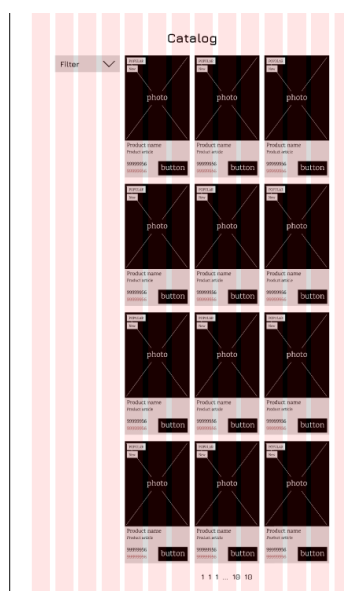


Рис. 1.4 Wireframe каталогу

Для сторінки з детальною інформацією про кросівки додаємо 3д модель та слайдер для можливості огляду всіх фотографій кросівок у разі необхідності. Також для більш корисного і комфортного користування додамо картки кросівок з іншими кольорами.

Також додаємо кнопку з розмірною сіткою всіх брендів для точного підбору розміру.

Також під фотографіями та блоком з 3D моделлю додаємо важливу коротку інформацію про кросівки.(Рис. 1.5)

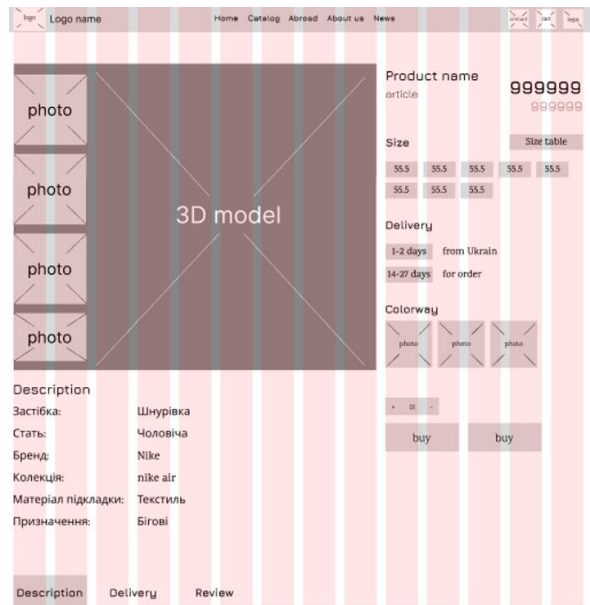


Рис. 1.5 Wireframe сторінки з детальної інформацією

Після успішного створення Wireframe для подальшої прототипізації проекту необхідно створити UIkit.

UIkit – це файл зазвичай розділений на декілька частин, таких як типографія, кольора, компоненти та організми. Всі елементи робляться в одному стилі для одного проекту.[8]

Компоненти – інструмент в фігмі за допомогою якого ти створюєш один елемент і можеш використовувати його всюди, коли змінюєш батьківський компонент – змінюються всі дочірні. Також у кожного такого компоненту є свої властивості.[9]

Організм – це той самий компонент але який складається з двох та більше компонентів та має властивості всіх компонентів які в ньому містяться, також може мати і свої властивості.[10]

До типографії входять всі стилі тексту які використовуються які діляться на стилі тексту для заголовків різних рівнів – H1, H2, H3, H4, H5 та стилі для основного тексту які позначаються як «body/12 medium» body – говорить про те що це

стиль для основного тексту та відповідно 12 medium – говорить про розмір та інтенсивність жирності.(Рис. 1.6)

## Typography styles

<b>H1</b> Juts Bold 55px / auto	The quick brown fox jumps over the lazy dog.
<b>H2</b> Juts Bold 40px / 120%	The quick brown fox jumps over the lazy dog.
<b>H3</b> Juts SemiBold 30px / 120%	The quick brown fox jumps over the lazy dog.
<b>H4</b> Juts Light 25px / 150%	The quick brown fox jumps over the lazy dog.
<b>H4 bold</b> Juts Bold 25px / 150%	The quick brown fox jumps over the lazy dog.
<b>H5</b> Juts Bold 18px / auto	The quick brown fox jumps over the lazy dog.

Рис. 1.6 Стилi типографiї

В створенні приємного для ока інтерфейсу в типографії використовується максимум 2 типи шрифтів, один для заголовків, а інший для основного тексту. Для заголовків використовується максимум 2 гарнітури, в основному жирний(Bold) та напівжирний(SemiBold), в той час як для основного тексту може використовуватись більше але не набагато, наприклад нахилений(Italic) для виділення цитат чи посилань. Для зручного створення UIkit з типографікою використовується плагін Typography style guide – допомагає перенести всі стилі з прототипу в UIkit.

В кольоровій частині UIkit вказуються всі використані кольори з вказанням інтенсивності від 100 до 0 після назви кольору його інтенсивності в палеті градієнту, також вказаний код HEX або RGB. Наприклад purple/100 – буде найтемніший й найнасиченіший ближче до чорного, а от purple/00 – буде найсвітлішим ближче до білого.(Рис. 1.7)

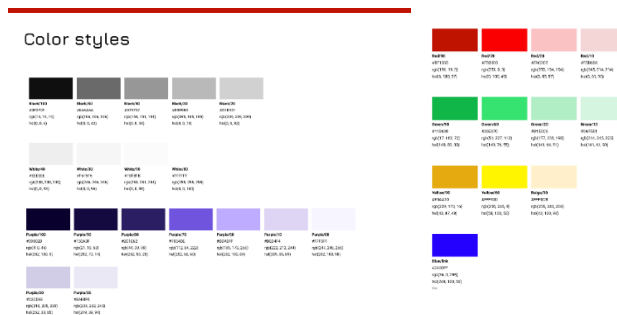


Рис.1.7 Інтенсивність кольорів UIkit

Для зручного створення UIkit з типографікою використовується плагін Color style guide. Також для зручного підбору використовується плагін Stark, який допомагає у визначенні правильної контрастності для кнопок та інших елементів. Наприклад, ми створюємо кнопку з темно фіолетовою заливкою і білим текстом, вводимо всі кольори і плагін покаже наскільки добра контрастність, з цього ми можемо зробити висновок наскільки буде читабельним текст.

Основною складовою UIkit який містить в собі компоненти та організми. З компонентів які використовуються самостійно ми маємо:

Кнопки –мають 4 стани(Default - статичний,Pressed – при натисканні, Active - активна, Hover – при наведенні, Disabled - неактивна) та три види(General – основна кнопка яка має привертати увагу користувача наприклад «Купити», Secondary – другорядна кнопка яка не має сильно привертати увагу наприклад «Обране» у нас в пріоритеті щоб користувач купив товар , Delete – зазвичай червоного кольору і сильно виділяється, Description/Brands – інші види кнопок для використання) (Рис. 1.8)

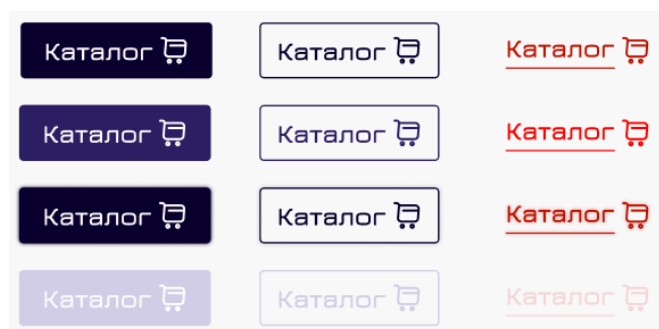


Рис. 1.8 Візуалізація кнопок

Перейдемо до організмів , що буду використовувати на веб-сайті :

Шапка сторінки – має статичний стан, також можна додати стани для виведення пошуку каталог та корзину. Містить в собі такі компоненти як Логотип(завжди в статичному стані), кнопки для навігації по сторінці та організм в якому додані кнопки пошуку, корзини, контактів і тд.(Рис. 1.9)

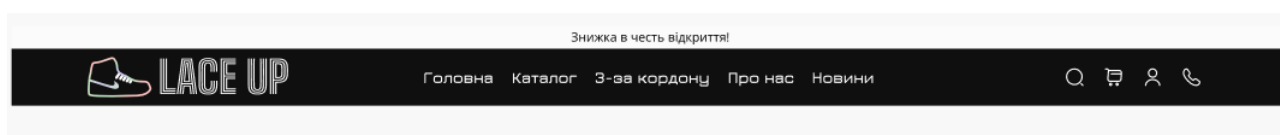


Рис. 1.9 Шапка сторінки

Footer – який складається з окремих елементів тексту для позначення контактної інформації, про магазин, графіку роботи та посилань на соціальні мережі. Має два розміри великий та маленький для подальшого використання в оформленні замовлення та реєстрації, який має менше тексту щоб не відволікати увагу і мати більш зосереджений інтерфейс на інших діях. (Рис.1.10)

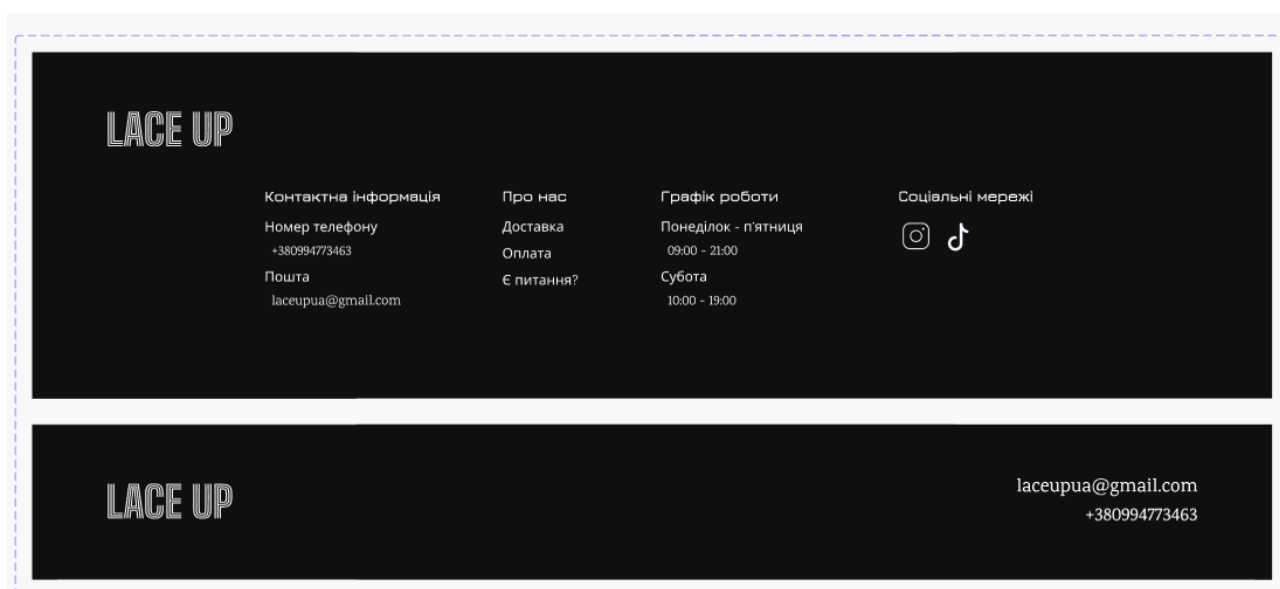


Рис. 1.10 Footer(нижня частина) сторінки

Картки товару– містять в собі іконки з статусом товару та кнопки додавання в обране та купівлі які з’являються тільки при наведенні знову ж таки щоб не відвертати увагу користувача від кросівок. (Рис. 1.11)

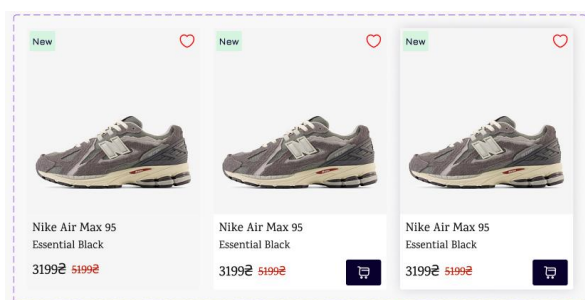


Рис. 1.11 Картки товару

Карточки новин – які містять в собі властивості зміни тексту та іконок. Також мають чотири різні розміри для більш цікавого і нетипового інтерфейсу. (Рис. 1.12)



Рис. 1.12 Картки новин

Фільтр – має декілька станів класичні при натисканні та розгорнутий. В розгорнутому стані маємо елементи тексту та компоненти CheakBox(там де можна обрати одні і не більше підкатегорій) та Radiobutton(можна обрати хоч всі підкатегорії). Також присутня кнопка «очистити» для очищення всіх виборів та кнопки «Більше» щоб розгорнути всі варіанти обраної категорії.

Всі дизайнери використовують різні види сіток для кожного проєкту, для створення сайтів/додатків на ПК/планшети(пристрої з великим екраном) використовується сітка виду Columns(вертикальні колоночки) або



Rows(горизонтальні колоночки) в яких можна регулювати їх колір, ширину, відстань між ними а розташування по лівому/правому краю або центральне, а для створення ігор та сайтів/додатків на смартфони зазвичай використовують сітку виду Grid(виглядає як піксельна сітка) в якій також можна змінювати колір та розмір «пікселів». (Рис. 1.13)

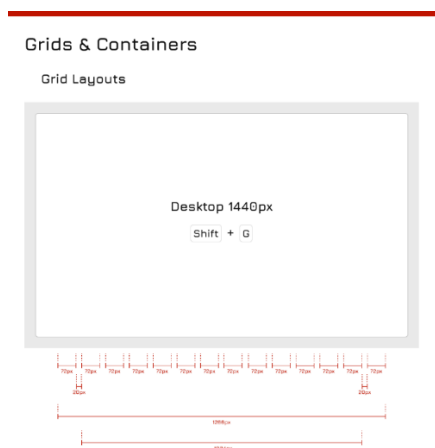


Рис. 1.13 Сітка Grid

Прототипізація – це створення моделі того , як буде виглядати готовий веб-сайту , мобільний додаток або якась програма.[11]

Прототипізація створюється вже на основі готового Wireframe , та Uikit це забезпечує розробнику , уже готовий шаблон по якому він буде створювати інтерфейс з додаванням кольорових палітр .

Тож перейдемо до розробки першої – головної сторінки веб-сайту . Ключову роль тут буде відігравати основний вміст сторінки , де й буде знаходитися 3D модель.

Почнемо з початку , на веб-саті на всіх сторіках буде знаходитися шапка (меню) та footer(нижня частина вебсайту ) про них я розказував вище .Далі на веб-сторінці знаходиться банер з 3D моделлю кросівка , у якого повинна бути анімація , та в залежності від повороту цього об'єкта на екрані повинен з'явитися текст. (Рис. 1.14)

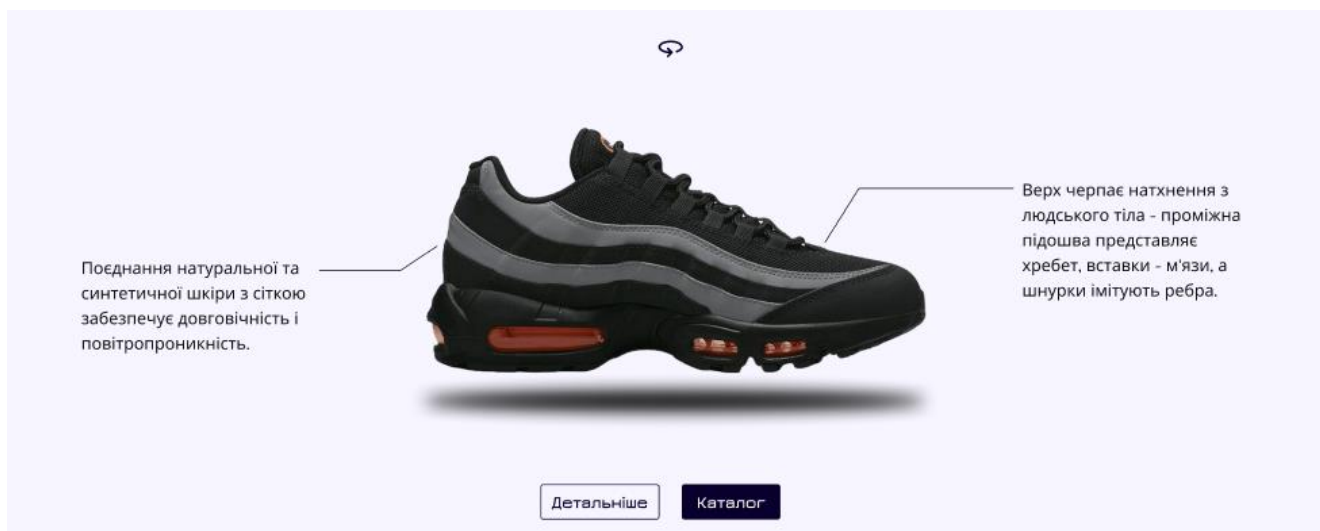


Рис. 1.14 Банер з 3D об'єктом

Цей банер є основним інтерактивним елементом веб-сайту, він повинен заохотити користувачів взаємодіяти з ним, за рахунок чого вони будуть дізнаватися певну інформацію про сам кросівок, що повинно їх більше стимулювати до покупки.

Наступний основний блок – це 2 слайдери, з кросівками, що щойно надійшли, та з новинами.

Слайдер – це такий інтерактивний елемент на веб-сайті, де в одному блоці розташовано безліч фотографій або певних блоків, що має функцію прокрутки з метою економії простору на сторінці, та збільшенню її динамічності та інтерактивності.[12]

Перний слайдер з новими надходження має мету показати користувачам, свіжі кросівки, що з'явилися на сайті цим самим їх зацікавити.

Другий – це слайдер для користувачів, що бажають дізнатися більше, там будуть знаходитися різні новини зі світу кросівок.

Також на обох слайдерах, є елементи навігації - це кнопки, для того щоб користувач міг переключати фотографії що є дуже важливим елементом UX. (Рис.1.15)

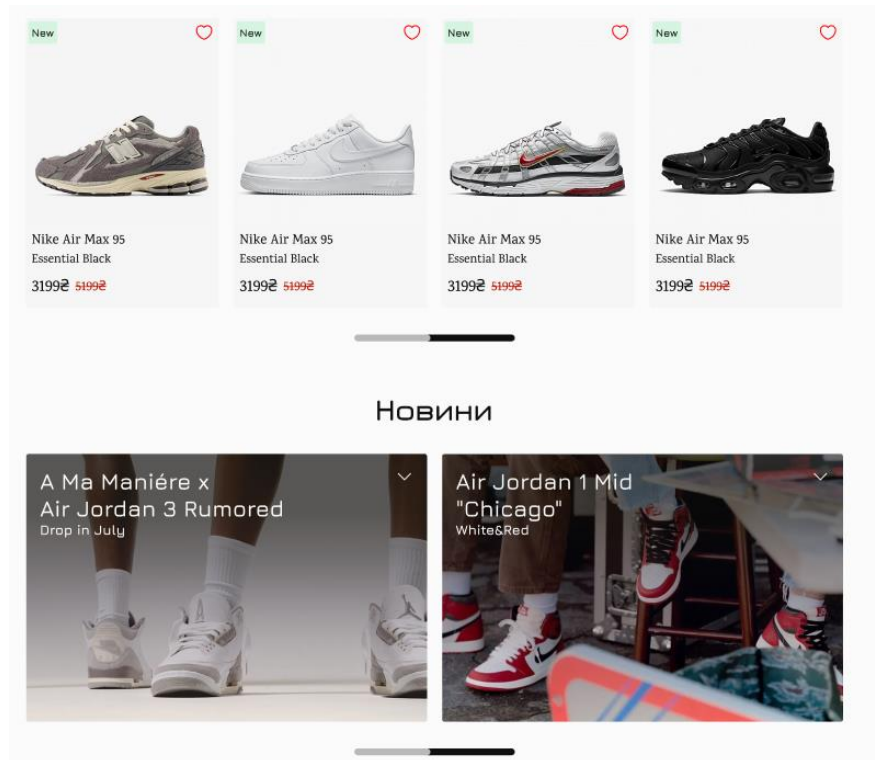


Рис. 1.15 Слайдери головної сторінки

Друга сторінка це Каталог товарів , що несе за мету показати користувачам розширений список того , що вони можуть придмати , з можливістю фільтрувати тільки те що їм потрібно , це є дуже важливим елементом UX.(Рис. 1.16)

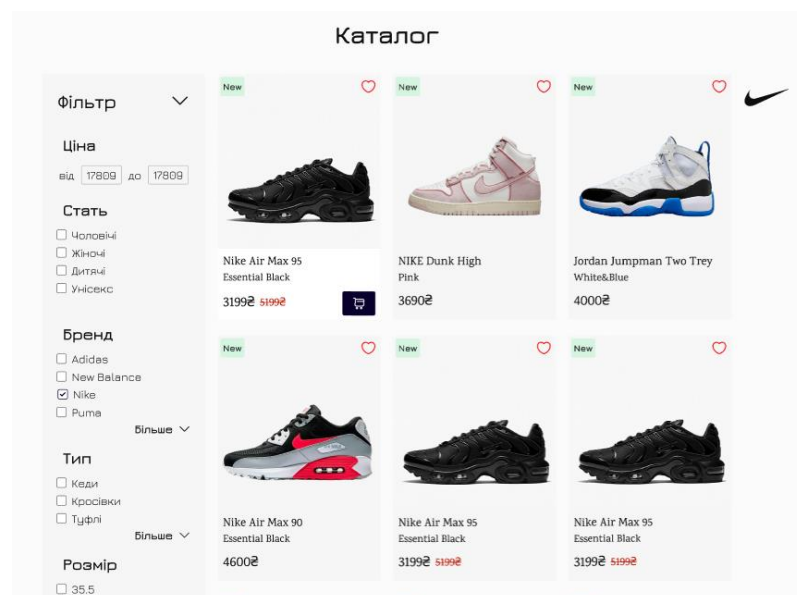


Рис. 1.16 Сторінка з Каталогом

Остання сторінка це сторінка з для окремого кросівка , тут знаходиться друга 3D модель , також вертикальний слайдер з фотографіями кросівка , інформація , що може бути цікава користувачеві , кнопки з розмірами кросівок та кнопки для покупки . 3D модель тут відіграє ключову роль, адже користувач , може детально роздивитись пару яку він обрав , та краще уявити обрану модель , це й є ключаовою ціллю розробки веб-сайтів з використанням 3D моделей , що також є чудовим елементом UX , адже приближує дизайн до реального світу. (Рис. 1.17)

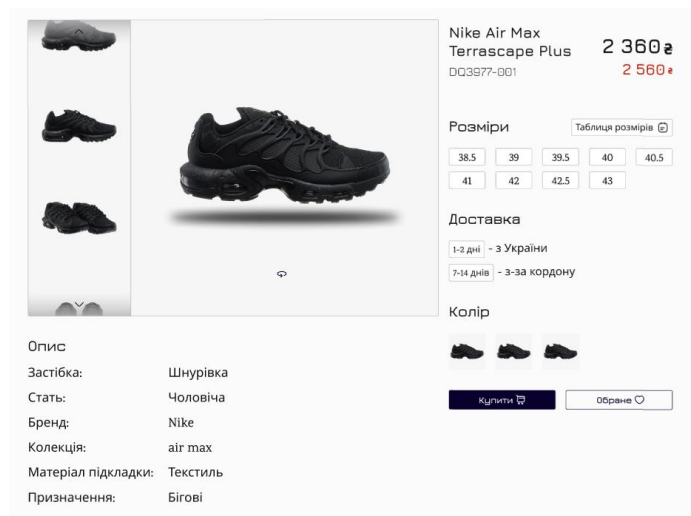


Рис. 1.17 Сторінка з детальної інформацією про кросівки

## 2 МЕТОДИ СТВОРЕННЯ 3D МОДЕЛЕЙ ТА ФУНКЦІ НА ВЕБ-САЙТАХ

### 2.1 Сучасні методи створення тривимірних об'єктів

3D модель – сукупність маленьких полігонів, що поєднані між собою . Для візуалізації тривимірних об'єктів використовують текстури , колір , матеріал поверхні та світло , 3D модель можна уявити як будь-який глиняний виріб , або об'єкт на вашому столі. [13] . (Рис. 2.1)

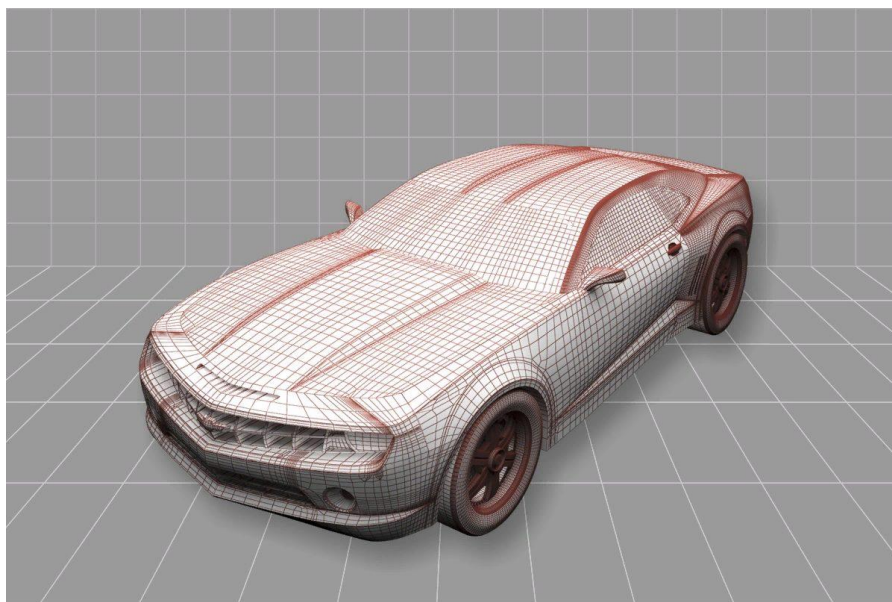


Рис. 2.1 Приклад 3D моделі[24]

На сьогодні існує велика кількість методів які допомагаю створювати 3D моделі такі, які : лазерне сканування , фотограмметрія , художнє моделювання або Sculpting , генерація за допомогою штучного інтелекту та інші...

- Лазерне сканування – це процес в якому вимірюється горизонтальне та вертикальне розповсюдження лазерних променів , в результаті якого отримується «Хмара точок» це так звана 3D модель отримана від сканування лазером [14] (Рис. 2.2)



Рис. 2.2 Приклад лазерного сканування[25]

- Фотограмметрія – це аналіз фотографій певного об’єкта зроблених з різних кутів . [15] (Рис. 2.3)



Рис. 2.3 Приклад Фотограмметрії



- Художнє моделювання : це створення 3D об'єктів в спеціальних програмах , на приклад Blender , Сінема 4D та інші , на основі приладів з реального життя або з нуля залежно від вимог поставлених перед розробником.[16] (Рис. 2.4)

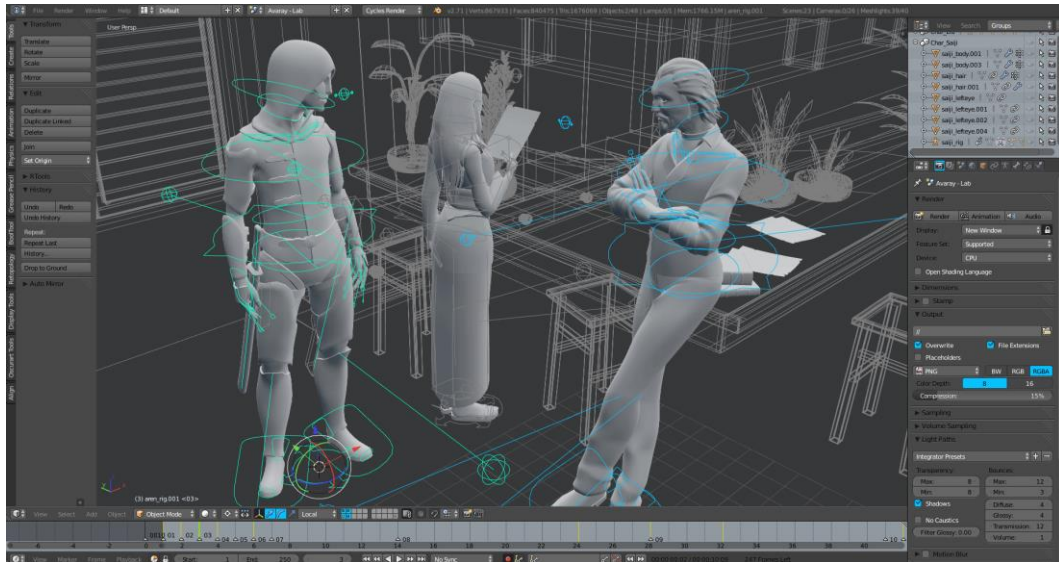


Рис. 2.4 Приклад художнього моделювання (Sculpting)[26]

- Генерація за допомогою штучного інтелекту – це спосіб створення 3D моделей які за вас створює ШІ на основі текстового запиту , або графічних зображень. До прикладу, неймережа Tripo AI в якій с за допомогою простого запиту “Create futuristic concept of nike air max shoes ” створив власну 3D модель за допомогою штучного інтелекту. (Рис. 2.5)

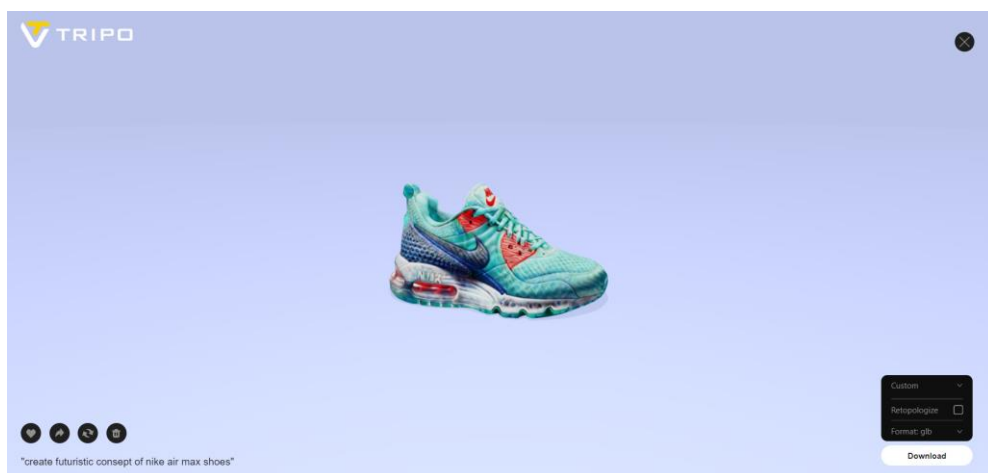


Рис. 2.5 Створення 3D моделей за допомогою ШІ

Також в сучасних телефонах є можливість сканувати предмети за допомогою власної камери , що також може бути допоміжним способом для створення тривимірних моделей , оскільки він є не точним та потребує подальшого допрацювання .До прикладу на рисунку нижче буде зображено 3D скан кросівка , що я зробив на свій телефон за допомогою камери та додатку Polycam , який можна завантажити в потрібному форматі , відредагувати програмі Blender або іншій та використати на власному веб-сайті наприклад для продажу кросівок . (Рис. 2.6)

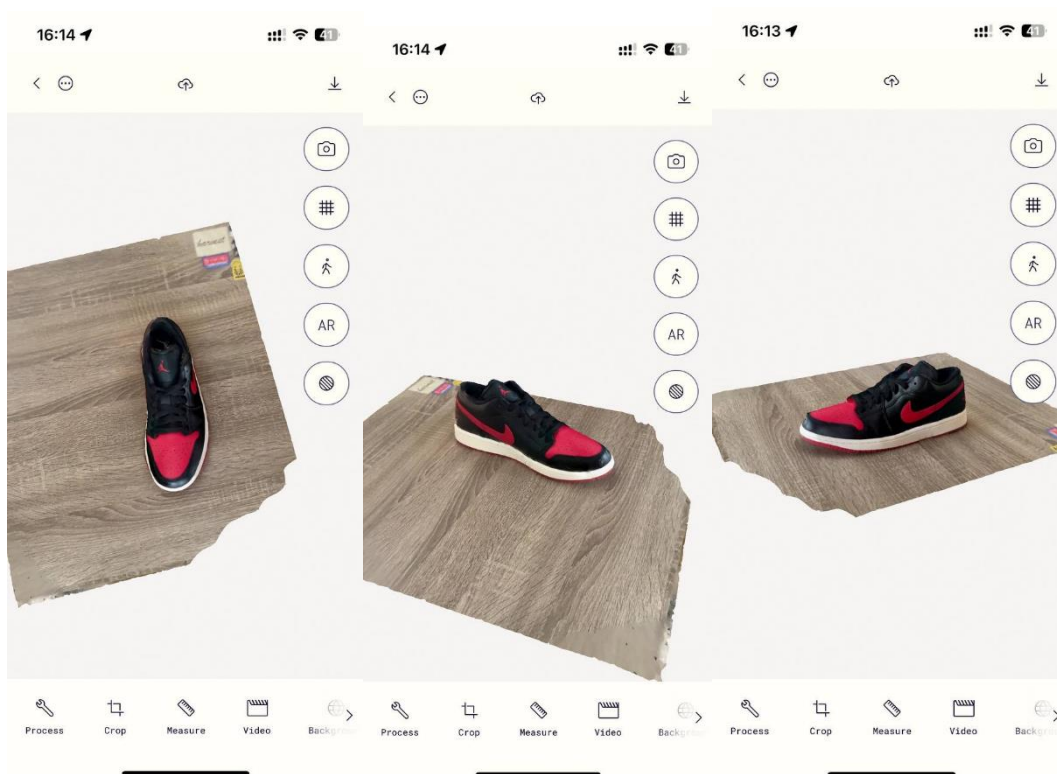


Рис. 2.6 3D Сканування об'єктів в мобільному додатку Polycam

## 2.2 Функції тривимірних об'єктів на веб-сайтах

Сьогодні , веб-сайти це більше ніж просто інструмент для надання інформації. Вони перетворилися на середовище , що заохочує користувачів взаємодіяти та проявляти свою увагу до певних процесів . Один із способів



покращити цей досвід для відвідувачів є використання 3D елементів дизайну . В цьому розділі буде описано численні переваги використання елементів 3D на вебсайтах , їх переваги , недоліки та те як вони підвищують зацікавленість користувачів веб-сайту за рахунок покращення візуальні привабливості та UX (досвіду користувача).[17]

### 1. Покращення привабливості.

Використовуючи 3D моделі , ми можемо створити глибину та реалістичність веб-сайту , що збільшує привабливість сайту та дає змогу користувачеві, робити порівняння з реальним світом.

Можна експериментувати зі світлом , тим самим додавати та міняти тіні на тривимірних об'єктах , що наближає візуал до реального , також поєднання різних текстур збільшує реалістичність сайту. 3D можна використовувати для того , що привернути увагу до ключових елементів веб-сторіки , наприклад товару що продає компанія або послуги які вона надає.

### 2. Інтерактивність

Інтерактивні 3D елементи , до прикладу продукт компанії тощо, захоплюють користувачів та збільшують їх залучення. Використовуючи різноманітні анімації та даючи користувачам можливість контролювати 3D об'є своїми руками ми підвищуємо їх бажання користувась сайтом , зацікавлюємо , та завдяки цьому вони будуть проводити більше часу на подібних веб-сайтах.

### 3. Підвищення впізнаваність бренду

Подібні сайти , які використовують 3D моделі , на довго увійдуть в пам'ять до користувачів , адже це новий досвід , який вони наврядчи раніше відчували.

Використання 3D в дизайні веб-сторінок стає все більш поширеним в останні роки. Зараз я хочу розглянути найпоширеніші елементи , що можуть бути використані на сайті , та покращити користувацький досвід .[18]

- 3D анімація – оживляє елементи використовуючи рух та взаємодії

- 3D моделі - додавання інтерактивних моделей , з якими можна взаємодіяти в тривимірному просторі
- 3Dграфіка – створює відчуття глибини та реальності.

### **2.3 Способи використання 3D елементів на сайті**

Використання 3D елементів на сайті заключається не тільки в імплементації 3D моделі на сайт , окрім цього є багато інших способів які я зараз хочу розглянути.

#### **1. Parallax фонове зображення.**

Parallax це ефект коли фонове зображення рухається повільніше ніж основний контент сторінки це й задає глибину сайту.

#### **2. Інтерактивний елемент.**

Можна задавати тривимірні моделі як інтерактивні елементами , з якими користувач може взаємодіяти або дати анімацію при скролі , що також дуже оживляє сторінку та більше привертає увагу.

#### **3. Вітрина товару**

Додати 3D модель яку користувач зможе обертати та масштабувати , цим самим підкреслити та показати ваш продукт з усіх сторін.

Недоліки використання 3D елементів на веб-сайтах:

- Основною проблемою є сумісність. Не всі браузері можуть підтримувати 3D об'єкти на сайтах , тому в даному випадку потрібно розробнику придумати альтернативне рішення для таких браузерів.
- Також апаратне забезпечення користувачів може бути не достатньо потужним для тихих веб-сторінок , що зробить їх досвід користування не дуже приємним.

- Мобільна адаптивність також грає ключову роль , деяку анімацію майже не реально адаптувати до мобільних пристроїв просто через те що вона призначена для великих екранів.
- Складні 3D моделі та текстури з високою роздільною здатністю будуть завантажуватися довше , адже потребують більших ресурсів , тому це впливає на швидкість завантаження цілої веб-сторінки.

### 3 РОЗРОБКА FRONTEND ЧАСТИНИ ВЕБ-САЙТУ ТА ІМПЛЕМЕНТАЦІЯ 3D МОДЕЛІ

#### 3.1 Вибір технологій для написання програмного коду

Для виконання цієї курсової роботи я обрав Visual studio code як середовище для розробки сайту, бо VS code має велику кількість допоміжних розширень, що полегшують розробку, React js для написання програмного коду, та CSS для стилізації вебсайту. Почнемо з першого, а саме React js – це бібліотека для мови програмування JavaScript, що використовується для розробки UI (інтерфейс користувача). Ця бібліотека розроблена компанією Facebook. React дозволяє розробляти високопродуктивні та масштабні проекти. Основною коцепцією React js є компоненти, на основі яких й пишеться програмний код. Перевага компонентів полягає в тому, що розробник може розбити код на блоки, що спрощує візуальне сприйняття коду та його написання, також компоненти можна використовувати повторно, що також дуже допомагає розробникам, адже економить час. Окрім цього React js є одною з найпопулярніших бібліотек для JavaScript, тому для нього доступно багато додаткових бібліотек, що спрощують вирішення багатьох задач.[19]

React js має ряд наступних переваг :

- Використовує віртуальну DOM (Document Object Model), та алгоритм постійного оновлення, що дозволяє оптимізувати внесення в зміни DOM та зміни стану сторінки.
- React js використовує компонентну структуру – де кожен компонент це відповідний блок інтерфейсу користувача, це спрощує тестування, оптимізацію та підтримку коду, оскільки компоненти можуть бути перевикористані та видозмінені за потреби.

Для ефективного написання коду на React потрібно також володіти знаннями HTML для створення компонентів та структури сторінки , CSS для стилізації JavaScript , адже React це бібліотека для цієї мови програмування , тож весь код програми буде написаний на JS , React Routing він потрібен для навігації між сторінками , мати розуміння що таке Component Lifecycle (життєвий цикл компонента) – це події що відбуваються при створенні , оновленні , та знищенні компонента .

React використовує синтаксис JSX це є синтаксичне розширення для JavaScript , воно використовує поєднання HTML та JS . React вважає , що логіка рендерингу тісно пов'язана з логікою інтерфейсу користувача , а саме те як обробляються події що відображаються на сторінці та як саме змінюється її стан , та те як дані будуть відображатись на сторінці.(Рис. 3.1)

```
const name = 'Josh Perez';
const element = <h1>Hello, {name}</h1>;
```

Рис. 3.1 Приклад використання синтаксису JSX

Компоненти в React js мають 3 фази життєвого циклу : монтаж , оновлення та демонтаж. Кожен з них має свої функції , що відповідають за певний етап циклу[20]. (Рис. 3.2)

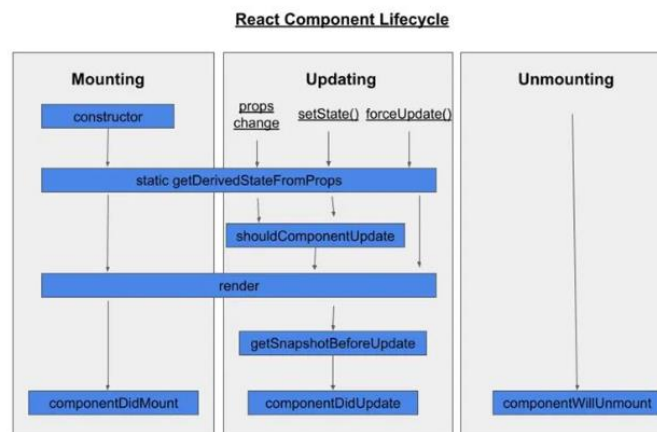


Рис. 3.2 Життєвий цикл компонентів React[27]

Фаза монтування – процес створення нового компонента , він додається до DOM , простими словами це початок життя нового компонента або по іншому «Початковий рендеринг». Є чотири методи цієї фази життєвого циклу - constructor, static getDerivedStateFromProps, render і componentDidMount[20]. (Рис.3.3)

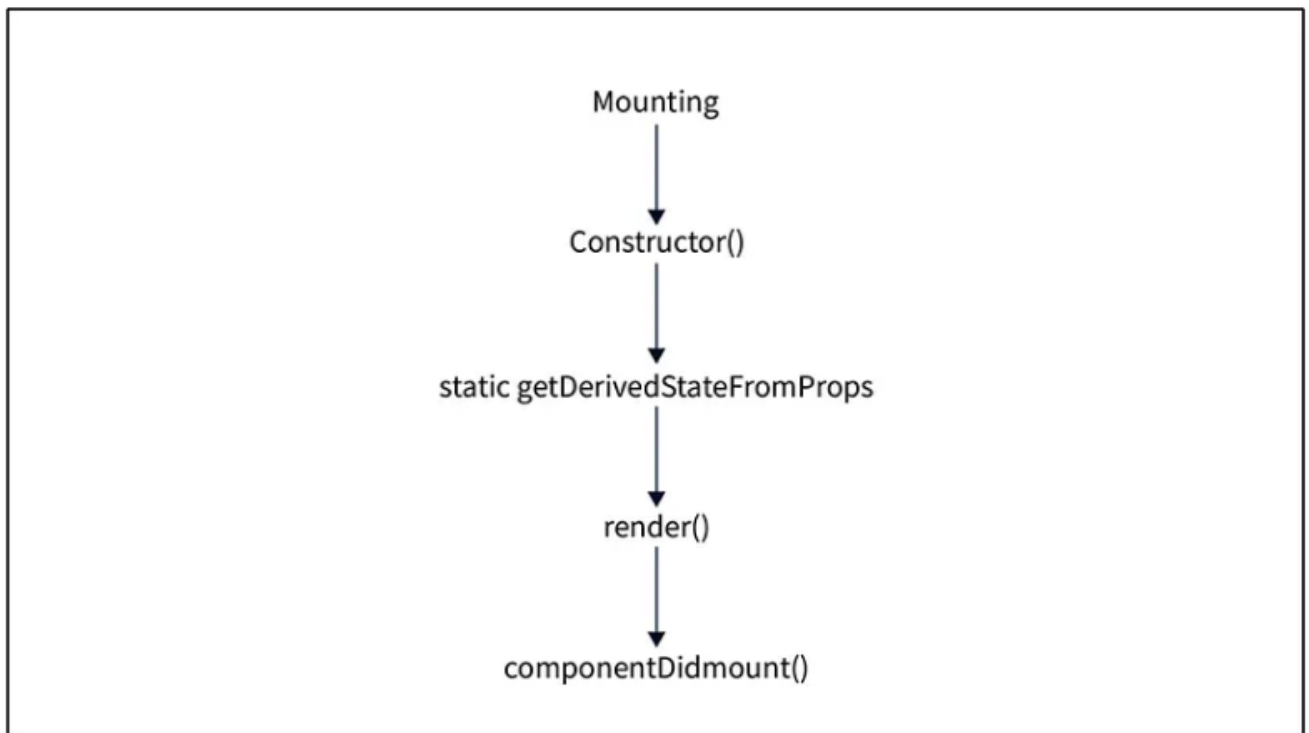


Рис. 3.3 Методи фази монтажу компоненту[27]

- Constructor – перший метод фази монтажу , він ініціалізує компонент та зв’язує його з методами , що обробляють події.
- getDerivedStateFromProps – після ініціалізації викликається цей метод , він дозволяє компоненту оновлювати свій стан на основі props (значення яке передається в компонент).
- Render – рендерить компонент простими словами вставляє HTML в DOM та відображає його на сторінці.
- componentDidMount – це місце для виконання функції які ми не можемо виконати в без DOM.

Фаза оновлення – запускається коли компонент повинен знову рендеритись через зміну стану. В ній використовуються такі методи[20] (Рис. 3.4) :

- `static getDerivedStateFromProps` – отримує значення `props`.
- `shouldComponentUpdate()` – використовується для оптимізації.
- `render()` – рендерить компонент.
- `getSnapshotBeforeUpdate()` - викликається перед тим як зміни будуть задіяні в DOM.
- `componentDidUpdate()` – викликається останній та отримує поперезді значення `props` та стану компоненту як аргументи.

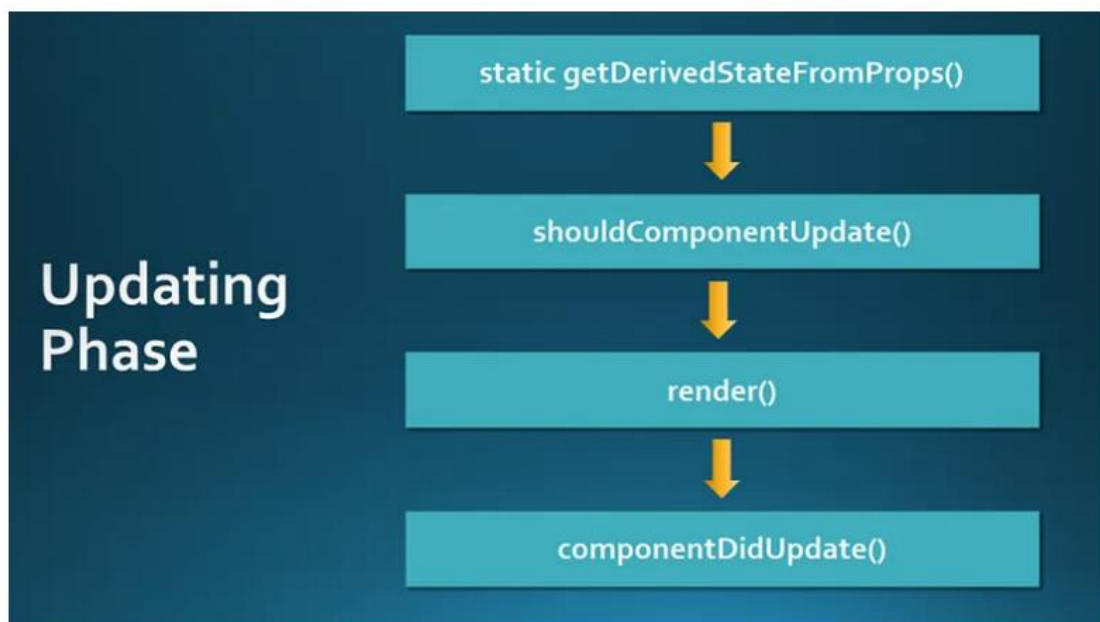


Рис. 3.4 Методи фази оновлення[27]

Фаза демонтажу – остання фаза життєвого циклу , і означає що компонент React видаляється з DOM. Під час цієї фази React очищує DOM.[20]

Хуки дозволяють використовувати багато функцій React та змінювати стан компонентів в реальному часі. [21]

- `State hook` – дозволяє React запам'ятовувати потрібну інформацію , до прикладу данні введені користувачем або поточний стан компонента для

його подальшої зміни. Хук `useState` задає зміну стану компоненту яку корзробник може контролювати як йому потрібно .

- `Ref hook` – `useRef` створює посилання на певний об'єкт `DOM` , це також потрібно для подальшого контролю стану цього об'єкту
- `Effects` – дозволя підключатися для зовнішнього середовища , наприклад до бібліотеки що не є частиною `React` , створювати та керувати анімаціями . `useEffect` – хук , що підключає компонент до зовнішньої системи.

`CSS` (Каскадні таблиці стилів) – це мова таблиць стилів призначена для стилізації `HTML` , а саме для розподілення вмісту сторінки , колір , шрифт та розмір тексту , адаптацію під різні розміри екрану , значення курсору , зміни стану `HTML` елементів , зміни заднього фону , стилізації зображень тощо.[22]

### 3.2 Розробка програмного коду `Frontend` частини

На картинці нижче сторінки , що будуть присутні на вебсайті.(Рис. 3.5)

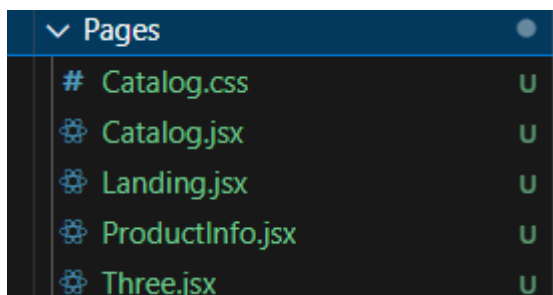


Рис. 3.5 Сторінки веб-сайту

1. `Landing` – початкова сторінка сайту.
2. `Catalog` – каталог товарів.
3. `ProductInfo` – сторінка для кожного кросівку окремо.
4. `Three` – полотно для 3D моделі.



На сторінці Landing буде знаходитись , меню , банер з 3D моделлю , панель з назвами брендів , два слайдери та нижня частина сайту (Footer).

Сторінка Catalog містить в собі бічну панель з фільтрами , а також список з наявних моделей кросівок.

Остання сторінка ProductInfo – це сторінка для окремої пари кросівок , де знаходиться вертикальний слайдер ще один 3D об'єкт , а також певна інформація про кросівки , окрім цього , там знаходиться , меню для навігації , слайдер та footer.

Кожна сторінка складається з певної кількості різних компонентів (окремих блоків коду , на які я умовно розділив сторінку)

Далі я хотів би коротко розібрати кожну з цих сторінок . почнемо з першої сторінки , а саме Landing – вона складається з таких компонентів :

- Navbar
- Banner
- Main
- Footer

Navbar – це компонент для меню веб-сайту, який буде виконувати функції переходу між сторінками.(Рис. 3.6)

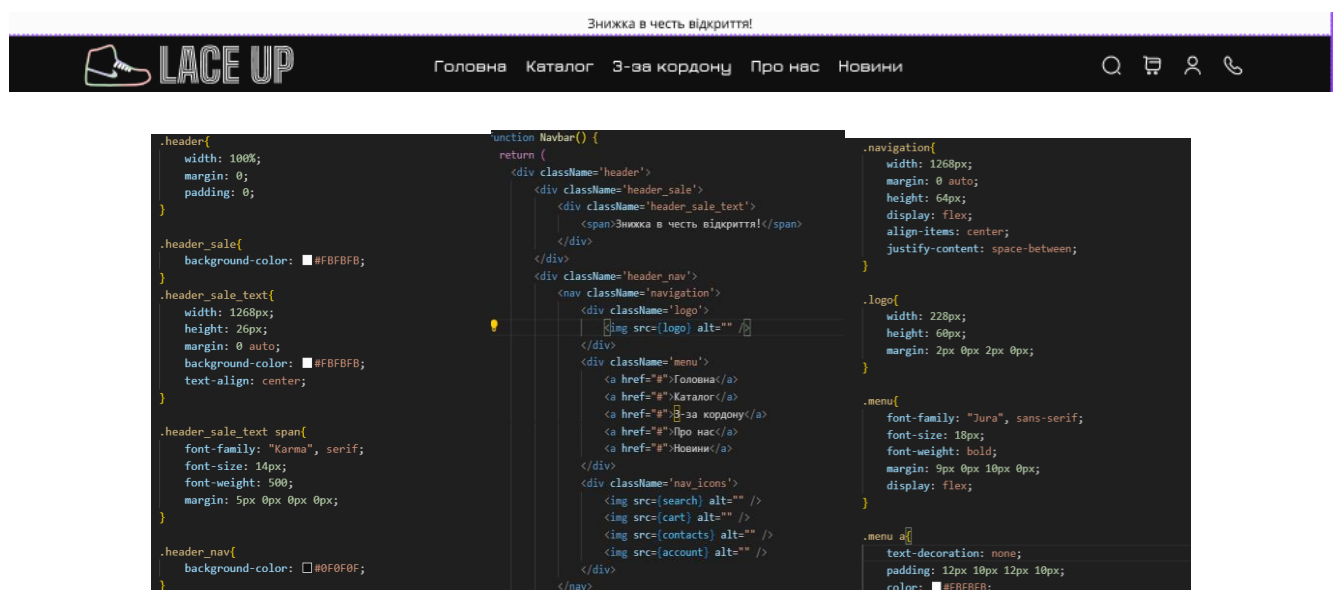


Рис. 3.6 Фрагменти коду React та CSS для компоненту Navbar

Компонент Navbar присутній на всіх сторінках , тож він буде перевикористовуватись постійно .

Наступний компонент це Banner – це компонент який містить саму 3D модель , про яку мова буде йти в наступному розділі .(Рис. 3.7)



```
function Banner() {
  return (
    <div>
      <Three/>
    </div>
  )
}
```

Рис. 3.7 Вигляд компоненту Banner

Компонент Main містить в собі , основну частину сторінки Landing , а саме слайдери , та панель з брендами .(Рис. 3.8)

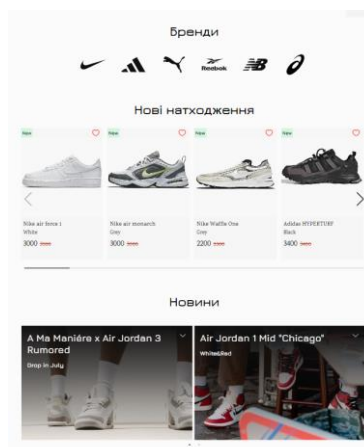


Рис. 3.8 – Вигляд компоненту Main

Цей компоненті розбитий на три блоки : блок з брендами , слайдер з новими натходженнями , та слайдер з новинами зі світу кросівок.(Рис. 3.9)

```

<div className='main'>
  <div className='container'>
    <div className='main_brands'>
      <p>Бренди</p>
      <div className='main_brands_images'>
        <img src={nike} alt="" />
        <img src={adidas} alt="" />
        <img src={puma} alt="" />
        <img src={reebok} alt="" />
        <img src={newbalance} alt="" />
        <img src={asics} alt="" />
      </div>
    </div>
    <div className='main_catalog'>
      <h2 className='main_catalog_title'>Нові натходження</h2>
      <ProductSlider card={card}/>
    </div>
    <div className='main_news'>
      <h2 className='main_news_title'>Новини</h2>
      <News/>
    </div>
  </div>
</div>

```

Рис. 3.9 Фрагмент коду компоненту Main

На рисунку вище , видно блок з брендами , та два блоки в яких знаходяться компоненти з слайдерми(Нові натходження та Новини).

Компонент ProductSlider , написаний за допомогою бібліотеки Swiper js для React js вона спрощує роботу в написанні слайдера/каруселі , адже в ній уже влаштовані CSS стилі та код js для функціонування слайдеру.(Рис. 3.10)

```

function ProductSlider ({card}) {
  return(
    <div className='card_container'>
      <Swiper
        modules={[Navigation, Pagination, Scrollbar, A11y]}
        spaceBetween={0}
        slidesPerView={4}
        navigation
        scrollbar={{ draggable: true }}
        onSwiper={ (swiper) => console.log(swiper)}
        onSlideChange={() => console.log('slide change')}
        className='product_slider'
        style={{width: '1084px'}}
      >
        {card.map((cards) =>(
          <SwiperSlide key={cards.id}>
            <Card cards={cards}/>
          </SwiperSlide>
        ))}
      </Swiper>
    </div>
  )
}

```

Рис. 3.10 Фрагмент коду компонента ProductSlider

Тут за допомогою props передано інформацію з файлу , що зберігає данні про кросівки, обробляє їх та додає до слайдеру за допомогою функції *map*. (Рис. 3.11)

```
src > Components > Data > JS product_cardjs > card
1  import af from '../Assets/productcard_af.png'
2  import nb from '../Assets/productcard_nb.png'
3  import nf from '../Assets/productcard_nf.png'
4  import psix from '../Assets/productcard_psix.png'
5  import new_icon from '../Assets/new icon.svg'
6  import like_icon from '../Assets/like icon.svg'
7
8  const card = [
9    {
10     id : 1,
11     model : "New Balance",
12     color : "Brown",
13     new : new_icon,
14     like : like_icon,
15     old_price : "5199",
16     new_price : "3899",
17     img : nb
18   },
19   {
20     id : 2,
21     model : "Nike Air Force 1",
22     color : "White",
23     new : new_icon,
24     like : like_icon,
25     old_price : "5199",
26     new_price : "3899",
27     img : af
28   },
29 ]
```

Рис. 3.11 Javascript файл з даними для слайдеру

На відміну від слайдеру з Новими натходженнями , слайдер з Новинами написаний за допопмогою бібліотеки react-slick . Вона виконує тіж самі функції що й Swiper , проте в цьому випадку , він мені більше підходить по функціоналу.(Рис.3.12)

```
function News() {
  var settings = {
    infinite: true,
    slidesToShow: 2,
    slidesToScroll: 2,
    autoplay: true,
    autoplaySpeed: 4000,
    pauseOnHover: true,
  };
};
return (
  <div className='news_container'>
    <Slider {...settings} className='news_slider'>
      {news.map((item) =>
        <div className='news_component' key={item.id}>
          <div className='news_content' style={{
            backgroundImage: 'linear-gradient(180deg, rgba(15,15,15,1) 0%, rgba(255,255,255,0) 100%)',
            backgroundSize: 'cover',
            backgroundRepeat: 'no-repeat',
            backgroundPosition: '50% 50%',
            width: '532px',
            height: '355px'
          }}>
            <div className='news_text'>
              <h1>{item.title}</h1>
              <p>{item.context}</p>
              <img src={nr} alt="" />
            </div>
          </div>
        )}
    </Slider>
  </div>
)
```

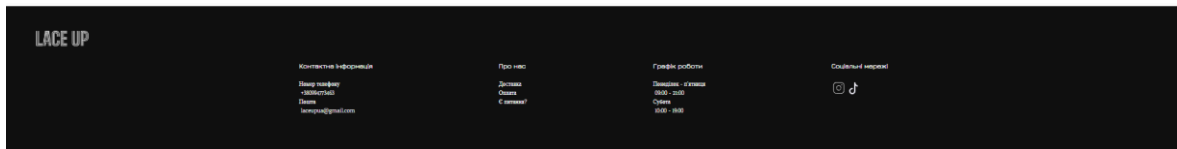
Рис. 3.12 Фрагмент коду слайдеру з новинами

В ньому використовується змінна `settings` , за допомогою якої можна задати всі потрібні налаштування для слайдера , він також за допомогою функції `map` додає елементи з файлу `news.js` в слайдер.(Рис. 3.13)

```
const news = [  
  {  
    id: 1,  
    image : j_three,  
    title : "A Ma Manière x Air Jordan 3 Rumored",  
    context : "Drop in July"  
  },  
  {  
    id: 2,  
    image : j_one,  
    title : "Air Jordan 1 Mid \"Chicago\"",  
    context : "White&Red"  
  },  
  {  
    id: 3,  
    image : lb,  
    title : "\"Light Bone\" Captivates The Nike Cosmic Unity 3",  
    context : "The ecologically minded kicks go totally neutral."  
  },  
  {  
    id: 4,  
    image : cb,  
    title : "Where To Buy The Nike Kobe 4 Protro \"Philly\"",  
    context : "The Nike Kobe 4 Protro \"Philly\" officially releases on April 13th"  
  }  
]
```

Рис. 3.13 Фрагмент коду файлу з даними для слайдеру з новинами

Наступний компонент сторінки Landing це Footer , в подальшій розробці цей компонент буде також перевикористовуватись , що зекономить , багато часу в розробці та , оптимізує сам код та сайт. (Рис. 3.14)



```

.footer_container {
  margin-top: 82px;
  width: 100%;
  background-color: #0F0F0F;
  height: 402px;
}

.footer_logo img {
  margin-top: 60px;
  margin-left: 86px;
}

.footer_content {
  width: 50%;
  margin: auto;
  display: flex;
  align-items: top;
  justify-content: space-between;
}

.footer_content h5 {
  font-family: "Jura", sans-serif;
  font-weight: 700;
  font-size: 18px;
  color: #FFFFFF;
}

function Footer() {
  return (
    <div className='footer_container'>
      <div className='footer_logo'>
        <img src={logo} alt="" />
      </div>
      <div className='footer_content'>
        <div className='footer_contacts'>
          <h5>Контактна інформація</h5>
          <div className='footer_contacts_phone'>
            <p className='footer_text'>Номер телефону</p>
            <p className='footer_subtext'>+380994773463</p>
          </div>
          <div className='footer_contacts_email'>
            <p className='footer_text'>Пошта</p>
            <p className='footer_subtext'>laceupua@gmail.com</p>
          </div>
        </div>
        <div className='footer_about'>
          <h5>Про нас</h5>
          <p className='footer_text'>Доставка</p>
          <p className='footer_text'>Оплата</p>
          <p className='footer_text'>Є питання?</p>
        </div>
        <div className='footer_schedule'>

```

Рис. 3.14 Зовнішній вигляд та фрагмент коду React та CSS компоненту Footer

Натупна сторінка це Catalog , тут зібраний список всіх наявних кросівок та бічна панель для фільтрації .(Рис. 3.15)

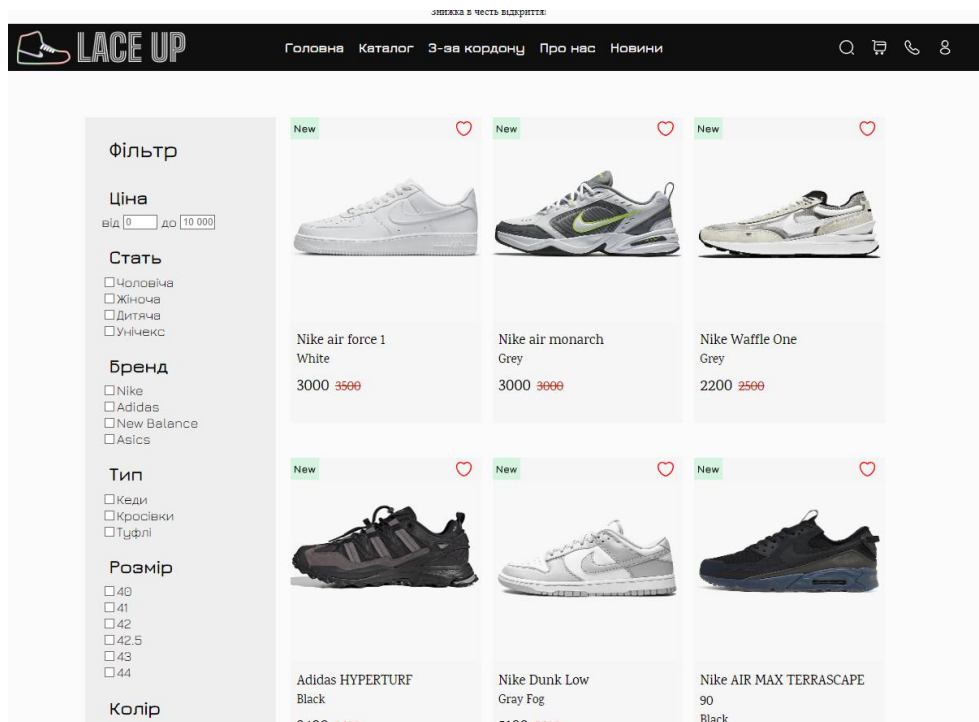


Рис. 3.15 Зовнішній вигляд Сторінки Каталог

Сторінка Catalog складається з компонентів Navbar, Footer з якими ми уже знайомі , та основний блок складається з Sidebar та Card list.

Sidebar – це компонент з фільтрами потрібний для фільтрації елементів.(Рис.3.16)

```
function Sidebar() {
  return (
    <div className='sidebar_container' >
      <div className="sidebar_filter">
        <div className="filter_title">
          Фільтр
        </div>
        <div className="sidebar_price filter_mg">...
        </div>
        <div className="sidebar_sex filter_mg">...
        </div>
        <div className="sidebar_brand filter_mg">...
        </div>
        <div className="sidebar_category filter_mg">...
        </div>
        <div className="sidebar_size filter_mg">...
        </div>
        <div className="sidebar_color filter_mg">...
        </div>
        <div className="sidebar_seson filter_mg">...
        </div>
        <div className="sidebar_delivery filter_mg">...
        </div>
        <div className="filter_clear">
          <p>Очистити</p><img src={tr} alt="" />
        </div>
      </div>
    </div>
  )
}

.sidebar_container{
  width: 256px;
  height: 1200px;
  background-color: #EEEEEE;
  padding: 24px 0 16px 0;
  margin-right: 20px;
}

.filter_mg{
  margin-bottom: 21px;
}

.filter_checkbox{
  margin-right: 3px;
}

.sidebar_filter{
  margin-left: 22px;
}

.sidebar_filter p{
  font-family: "Jura", sans-serif;
  font-weight: 400;
  font-size: 18px;
}

.filter_title{
  font-family: "Jura", sans-serif;
  font-weight: 600;
  font-size: 30px;
}
```

Рис. 3.16 Фрагменти коду React та CSS для компоненту sidebar

Компонент Card – містить в собі карточку товара , дані для якого беруться з файлу з даними , та за допомогою функції map створює елементи для кожної карточки . (Рис. 3.17)

```
<div className='catalog_list' >
  {card.map((cards)=>(
    <Card cards={cards}/>
  ))}
</div>
```

Рис. 3.17 Створення карток для каталогу

Тут через props передані дані з файлу з даними в компонент Card . На рисунку нижче зображено фрагменти коду для картки з товаром.(Рис. 3.18)

```
function Card({cards}) {
  return (
    <div className='card_list_container'>
      <div className='card' key={cards.id}>
        <div className='card_icons'>
          <img src={cards.new} alt="" className='card_icons_new' />
          <img src={cards.like} alt="" className='card_icons_like' />
        </div>
        <div className='card_img'>
          <img src={cards.img} alt="" />
        </div>
        <div className='card_content'>
          <div className='card_content_text'>
            <p className='card_content_model'>{cards.shoe_name}</p>
            <p className='card_content_color'>{cards.shoe_color_name}</p>
          </div>
          <div className='card_content_price'>
            <p className='card_content_price_new'>{cards.new_price}</p>
            <p className='card_content_price_old'>{cards.old_price}</p>
          </div>
        </div>
      </div>
    </div>
  )
}

.card_list_container{
  width: 256px;
}
.card {
  width: 256px;
  height: 410px;
  background-color: #F6F6F6;
  margin-bottom: 32px;
}
.card_icons{
  position: relative;
  top: 0;
  display: flex;
}
.card_icons_new{
  height: 29px;
  width: 38px;
}
.card_icons_like{
  width: 30px;
  height: 30px;
  margin-left: 180px;
}
```

Рис. 3.18 Фрагменти коду React та CSS для компоненту Card

Остання сторінка сайту – це сторінка , окремо для кожної пари кросівок з каталогу , вона складається з таких блоків (Рис. 3.19) :

1. Меню
2. Вертикальний слайдер
3. Полотно з 3D моделлю
4. Опис товару
5. Основна частина (складається з ціни , розмірів , кольорів доступних, строки доставки)
6. Слайдер рекомендуємих кросівок
7. Футер

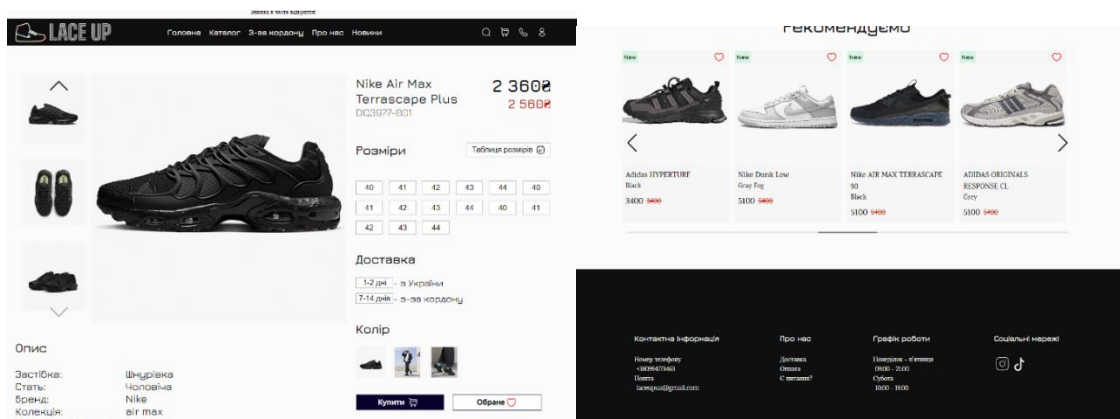


Рис. 3.19 Вигляд сторінки з певною парою



Перший компонент це вертикальний слайдер – він написаний за допомогою бібліотеки Swiper JS , по аналогії з попередніми слайдерами , цей також додає елементи за допомогою функції map , єдина відмінність від попередніх , це те що він розташований горизонтально. (Рис. 3.20)

```
function ProductInfoVertical() {
  return (
    <div className="vertical_container">
      <Swiper
        modules={[Navigation,Mousewheel,]}
        direction='vertical'
        spaceBetween={10}
        slidesPerView={3}
        navigation
        mousewheel={true}
        onSwiper={(swiper) => console.log(swiper)}
        onSlideChange={() => console.log('slide change')}
        className='product_slider'
        style={{height: '584px'}}
      >
        {pi.map((item) => (
          <SwiperSlide key={item.id} style={{textAlign: 'center'}}>
            <img src={item.img} alt="" />
          </SwiperSlide>
        ))}
      </Swiper>
    </div>
  )
}
```

Рис. 3.20 Фрагмент коду для вертикального слайдера

Наступний компонент це полотно з 3D моделлю , який я детально розберу в наступному розділі. Після нього йдуть блоки з описом кросівок , та певною інформацією про них (вартість, розміри, колір , строки доставки), в цих блоках немає нічого нестандартного , це просто поєднання синтаксису JSX та CSS,також тут знаходиться перевикористай компонент слайдера та футера (Рис. 3.21)

```

<div className="description_content">
  <h3 className="content_title">
    Опис
  </h3>
  <div className="content_text">
    <div className="text_start">
      <p>Застібка:</p>
      <p>Стать:</p>
      <p>Бренд:</p>
      <p>Колекція:</p>
      <p>Матеріал підкладки:</p>
      <p>Призначення:</p>
    </div>
    <div className="text_end">
      <p>Шнурівка</p>
      <p>Чоловіча</p>
      <p>Nike</p>
      <p>air max</p>
      <p>Текстиль</p>
      <p>Бігові</p>
    </div>
  </div>
</div>
</div>
<div className="product_choose">
  <div className="choose_name">...
  </div>
  <div className="choose_size">...
  </div>
  <div className="choose_delivery">...
  </div>
  <div className="choose_color">...
  </div>
  <div className="choose_buttons">...
  </div>
</div>
<div className="product_recommend">
  <h1>
    Рекомендуємо
  </h1>
  <ProductSlider card={card}/>
</div>
</div>
)
}

.product_container{
  width: 1268px;
  margin: 0 auto;
  margin-top: 70px;
}

.product_content{
  display: flex;
}

.product_content button{
  background-color: #ffffff;
  border: 1px solid #898989;
  border-radius: 1px;
}

.description_sliders{
  display: flex;
}

.product_description{
  margin-right: 20px;
}

.choose_name{
  display: flex;
  justify-content: space-between;
  align-items: top;
  margin-bottom: 26px;
}

```

Рис. 3.21 Фрагмент коду React та Css для сторінки ProductInfo

Також на веб-сайт була додана навігація , в React вона називається React Routing, навігація буде між головною сторінкою , каталогом та сторінкою з детальної інформацією про кросівки.(Рис. 3.22)

```

function App() {
  return (
    <Router>
      <div>
        <Routes>
          <Route path="/" element={<Landing/>}</Route>
          <Route path="/catalog" element={<Catalog/>}</Route>
          <Route path="/ProductInfo" element={<ProductInfo/>}</Route>
        </Routes>
      </div>
    </Router>
  );
}

```

Рис. 3.22 Навігація між сторінками

На рисунку вище зображено компонент Router , який ініціалізує те що тут буде проводитись навігація , компонент Routes , зберігає в собі всі шляхи навігації

, компонент Route це сам шлях в параметри якого ми задаємо посилання , що буде відобразитись на вебсайти , та в середину цього тегу компонент сторінки.

В компоненті з нашим меню , ми повинні до кнопок прив'язати ці шляхи (Route), це ми робимо за допомогою компоненту Link , на відміну від стандартного переходу в HTML , що виконується за допомогою тегу "a". (Рис. 3.23)

```

<div className='menu'>
  <Link to="/">Головна</Link>
  <Link to="/catalog">Каталог</Link>
  <a href="#">З-за кордону</a>
  <a href="#">Про нас</a>
  <a href="#">Новини</a>

```

Рис. 3.23 Фрагмент коду Компоненту Link

В компоненті Link знаходиться параметр "to" який вказує посилання на сторінку на котруми повинні перейти.

### 3.3Імплементация 3D моделі та її анімація

На вебсайті який я розробив для дипломного проекту , повинно бути дві 3D моделі , одна на головній сторінці та на окремій сторінці з товаром. Обидві вони мають різні функції , розглянемо перший випадок . 3D модель що знаходиться на сторінці на головній сторінці , повинна інтерактивно відобразити інформацію про кросівки , також мати анімацію для того , щоб сильніше зацікавити користувача , також має давати доступ користувачам , взаємодіяти з собою. Тож приступимо до реалізації цих функцій.

Розглянемо теоретичні матеріали , які потрібні для виконання , реалізації цих функцій .

Як я зазначав раніше в React існує багато допоміжних бібліотек , що значно спрощують життя , тож для роботи з 3D моделями та реалізації анімацій я обрав наступні[23] :

- 1) React-three/drei – набір компонентів , який дозволяє та спрощує роботу з Three.js
  - a) Переваги :
    - i) є готові компоненти , для освітлення , камер , 3D об'єктів , середовища тощо.
    - ii) Зручний та зрозумілий синтаксис
    - iii)можлива інтеграція з іншими бібліотеками .
  - b) Приклади пикористання :
    - i) Створення реалістичного освітлення
    - ii) Розробка інтерактивних 3D об'єктів
    - iii)Зручна робота з камерою
- 2) React-three/fiber – потрібен для створення сцен з використанням 3D
  - a) Переваги:
    - i) Висока продуктивність
    - ii) Сумісність з React-three/drei
  - b) Приклади використання :
    - i) Створення 3D сцен , що можна перевикористовувати.
- 3) Gsap – бібліотека для створення різних плавних та динамічних анімацій .
  - a) Переваги :
    - i) Можливість синхронізації анімації з часом
    - ii) Широкий спектр налаштувань , на ефектів анімації
  - b) Приклади використання :
    - i) Анімація камер та 3D об'єктів
    - ii) Розробка анімованих 3D-ефектів для покращення користувацького досвіду.

Почнемо з того , які функції повинна виконувати ця модель , та поступово будемо визначати дані які нам потрібно знати , для реалізації цих функцій . Я хочу

щоб на головному екрані була 3D модель кросівка , яку користувач , може покрутити , та в залежності від кута нахилу , висвічувався текст , що буде розказувати про те що саме показується в даний момент .Також , додатково , я хочу додати анімацію при завантаженні сторінки , де 3D модель кросівка з'являється на екрані ніби імітуючи крок людини .

Тож почнемо з того , що ми створимо нашу сцену. (Рис. 3.24)

```
const Three = () => {
  return (
    <div className='threed_container' style={{width: '100%', height:'100hv'}}>
      <Canvas id='landing_shoe_threed' shadows>
        <LandingShoe/>
      </Canvas>
    </div>
  )
}

export default Three
```

Рис. 3.24 Фрагмент коду з створення сцени

На рисунку вище , зображено як створюється Сцена на яку можна додати 3D моделі . Сама сцена це компонент Canvas в середині якого компонент Landing shoe, в якому створено всі потрібні нам елементи такі як : світло , камера , навколишнє середовище , та сам 3D об'єкт.

Перейдемо до компоненту Landing shoe , перше на що я хочу звернути увагу , це камера , бо тільки через неї рендериться вся сцена , і якщо її не буде , то сцена відповідно нічого не буде відображати . Компонент PerspectiveCamera приймає такі параметри як MakeDefault робить її головною камерою та Position виставляє камеру в просторі по координатам x,y,z. Наступний компонент Airmax , це сама 3D модель просто реалізована в окремому компоненті з метою оптимізації коду , до нього я повернусь пізніше. AmbientLight , це один із видів світлового променя , який поширює світло по всій сцені , приймає такі параметри як : колір світла та інтенсивності поширення . Це один із найважливіших компонентів , адже без світла

н сцені не буде нічого видно , буде просто чорне полотно , тому важливо його не забувати додавати. Наступний компонент це Environment , ним можна задати те як буде виглядати навколишнє середовище , для цього за допомогою тега mesh який приймає параметр scale (масштабування) та масштабує себе в 100 разів , створюю 1 сферу ( компонент sphereGeometry) з параметром аргумент , який визначає розмір цієї сфери, та за допомогою компоненту MeshBasicMaterial , додаємо колір до сфери . Цей компонент приймає такі параметри як side (це потрібно для того , щоб колір відображався з внутрішньої сторони 3D об'єкта), та сам колір заливки.(Рис. 3.25)

```
const LandingShoe = () => {
  return (
    <>
      <PerspectiveCamera makeDefault position={[0, 1, 5]}/>
      <Airmax scale={2}/>
      <ambientLight args={["white", 1]}/>
      <Environment background>
        <mesh scale={100}>
          <sphereGeometry args={[1, 64, 64]}/>
          <meshBasicMaterial side={THREE.BackSide} color='■ #edd2ff' />
        </mesh>
      </Environment>
    </>
  )
}
export default LandingShoe
```

Рис. 3.25 фрагмент коду елементів сцени

Наступний компонент який я хочу розглянути це Airmax в ньому знаходяться функції та компоненти для додавання самої 3D моделі , а також анімація зроблена за допомогою бібліотеки Gsap , а також реалізація контролю 3D моделі з виведення на екран опису кросівок в залежності від куту повороту об'єкта.(Рис. 3.26)

```

import function Airmax(props) {
  const { nodes, materials } = useGLTF('/Models/Airmax/scene-transformed.glb')
  const [displayText, setDisplayText] = useState(null);
  const OrbitRef= useRef(null)
  const shoeRef = useRef(null)
  let polarangle;
  let azimuthalangle;
  useFrame(()=>{
    if(!OrbitRef.current){
      polarangle = OrbitRef.current.getPolarAngle()
      azimuthalangle = OrbitRef.current.getAzimuthalAngle()
      console.log('polar',polarangle)
      //console.log('azimuthal',azimuthalangle)

      if(azimuthalangle>-0.5 && azimuthalangle<0.6 && polarangle > 0.9 && polarangle < 1.8){
        setDisplayText('Остаточний дизайн моделі черпає натхнення з людського тіла.');
```

```

      }else if(azimuthalangle<-1.2 && azimuthalangle > -2.2 && polarangle > 0.8 && polarangle < 1.7){
        setDisplayText('backview');
      }else if(azimuthalangle>1.1 && azimuthalangle < 2 && polarangle > 0.97 && polarangle < 1.75){
        setDisplayText('nose view');
      }else if(azimuthalangle>2.3 && azimuthalangle >-2.2 && polarangle > 0.8 && polarangle < 1.85){
        setDisplayText('back view');
      }else if(polarangle < 0.4 ){
        setDisplayText('Верхня частина кросівок, виділена шаруватими панелями, схожими на м'язові волокна');
      }
      else if(polarangle > 2){
        setDisplayText('підшова - нагадує хребет');
```

```

    useEffect(()=>{
      if(!shoeRef.current){
        const timeline = gsap.timeline({paused:true})
        timeline.to(shoeRef.current.position, {
          x: 0,
          y:0,
          duration: 0.7,
          ease: "power1.out"
        })
        timeline.to(shoeRef.current.rotation, {
          y: angleToRadians(0),
          duration: 0.3,
        }, '<0.5')
        timeline.play()
      }
    },[shoeRef.current,])

    return (
      <>
      <OrbitControls ref={OrbitRef}/>
      <group {...props} dispose={null}>
        <mesh geometry={nodes.defaultMaterial.geometry}
          material={materials['1001']} position=[-2,3,0]
          rotations=[-angleToRadians(90), -angleToRadians(30), 0]
          ref={shoeRef}
        />
        <displayText && <html><div className='model_container'><div className='model_text' dangerouslySetInnerHTML={{__html: displayText }} /></div></html>
      </group>
    </>
  )
}

useGLTF.preload('/Models/Airmax/scene-transformed.glb')
```

Рис. 3.26 Фрагменти коду компонента Airmax

За допомогою Об'єкта GLTFLoader , завантажуюємо о'б'єкт який ми зберегли в форматі GLTF , та перетворили у формат GLB цей формат, зменшує розмір 3D об'єкту чим дозволяє робити сайт більш оптимізованим.

В наступному блоці коду виводимо 3D об'єкт на сцену , за допомогою уже знайомого нам тегу `mesh` , задаємо йому матеріал , розташування , та кут оберту (він знадобиться , коли буду розглядати анімацію) (Рис. 3.27)

```
<OrbitControls ref={OrbitRef}/>
<group {...props} dispose={null}>
  <mesh geometry={nodes.defaultMaterial.geometry}
    material={materials['1001']} position={[-2,3,0]}
    rotation=[-angleToRadians(90), -angleToRadians(30), 0]}
    ref={shoeRef}/>
</group>
```

Рис. 3.27 Виведення на сцену 3D об'єкта

Наступний блок коду , це створення анімації імітації кроку людини після завантаження сайту за допомогою бібліотеки `gsap` (Рис. 3.28)

```
const OrbitRef= useRef(null)
```

```
useEffect(()=>{
  if(!shoeRef.current){
    const timeline = gsap.timeline({paused:true})
    timeline.to(shoeRef.current.position, {
      x: 0,
      y:0,
      duration: 0.7,
      ease: "power1.out"
    })
    timeline.to(shoeRef.current.rotation, {
      y: angleToRadians(0),
      duration: 0.3,
    }, '<0.5')
    timeline.play()
  }
}, [shoeRef.current,])
```

Рис. 3.28 Анімація кроку людини



Як ми бачимо на рисунку вище я створив хук `UseRef` та присвоїв референс до моделі з кросівками , потім за допомогою хуку `useEffect` – який спрацьовує на оновленні , задаю анімацію , спочатку йде умова (якщо поточний стан кросівка існує , то ми виконуємо наступні дії ). Створив змінну `timeline` , вона потрібна для контролю анімацій в часі за допомогою бібліотеки `gsap` , `timeline.to` означає що ми беремо початкові данні моделі , а саме розсташування та кут нахилу , та перетворюємо їх , присвоюємо значення позиції на нульові координати по іксу та ігрику , задаємо тривалість в 0.7 секунд , та тип анімації `ease` , в даному випадку я обрав `power1.out` це один із готових варіантів , що пропонує бібліотека.Ця анімація дозволяє плавно винести о'б'єкт з-за кадру на нульові координати на які якраз диться камера. Наступна частина анімації це анімація куту нахилу по вілі `y` , довжина цієї анімації 0.3 секунди та вона настає після того як перша анімація програтася 0.5 секунд . Друга анімація дозволяє нам зробити імітацію кроку людини про який я говорив раніше . Та функцією `timeline.play` ми дозволяємо анімації програтись та відобразитись на сайті.

Наступний блок коду присвячений , виводу певного тексту на екран в залежності від куту повороту кросівок . Але перед цим потрібно знати деякі теоретичні аспекти , по перше потрібно дати доступ користувачу до камери , щоб він міг з нею взаємодіяти та крутити на всі 360 градусів , це ми можемо зробити за допомогою вбудованого в бібліотеку `React-three/drei` компонента `OrbitControls`. Також якимось чином потрібно визначити як можна зчитати кути нахилу ,для цього в компоненті `OrbitControls` є такі властивості як `azimuthal angle` , `polar angle` кут азимуту це кут нахилу камери по вісі `X` а полярний кут це кут нахилу по вісі `Y`. Тож для того , щоб затати якийсь певний функціонал для тексту , потрібно витягнути значення цих кутів та за допомогою умовних операторів задати обмеження коли текст буде з'являтися а коли ні . (Рис. 3.29)

```

const [displayText, setDisplayText] = useState(null);
const OrbitRef= useRef(null)
const shoeRef = useRef(null)
let polarangle;
let azimuthalangle;
useFrame(()=>{
  if(!OrbitRef.current){
    polarangle = OrbitRef.current.getPolarAngle()
    azimuthalangle = OrbitRef.current.getAzimuthalAngle()
    console.log('polar',polarangle)
    //console.log('azimuthal',azimuthalangle)

    if(azimuthalangle> -0.5 && azimuthalangle<0.6 && polarangle > 0.9 && polarangle < 1.8){
      setDisplayText('Остаточний дизайн моделі черпає натхнення з людського тіла.');
```

```

    }else if(azimuthalangle<-1.2 && azimuthalangle > -2.2 && polarangle > 0.8 && polarangle < 1.7){
      setDisplayText('backview');
```

```

    }else if(azimuthalangle>1.1 && azimuthalangle < 2 && polarangle > 0.97 && polarangle < 1.75){
      setDisplayText('nose view');
```

```

    }else if(azimuthalangle>2.3 && azimuthalangle >-2.2 && polarangle > 0.8 && polarangle < 1.85){
      setDisplayText('back view');
```

```

    }else if(polarangle < 0.4 ){
      setDisplayText(`Верхня частина кросівок, виділена шаруватими панелями, схожими на м'язові волокна`);
    }
  }
  else if(polarangle > 2){
    setDisplayText('підшва - нагадує хребет');
```

```

  }
}
}, [OrbitRef.current])
```

```

    else if(polarangle > 2){
      setDisplayText('підшва - нагадує хребет');
```

```

    }
    else{
      setDisplayText(null);
    }
  }
}, [OrbitRef.current])
```

Рис. 3.29 Фрагменти коду задання тексту в залежності від куту оберту 3D об'єкта

Спочатку створено хук `useState` – він потрібен для того, щоб в залежності від певного кута, зберігав в собі текст який потрібен, та виводив його на екран. Також створено хук `useRef` – який дає референс на компонент `OrbitControls`, з якого в хуці `useFrame`, я буду зчитувати потрібні дані. Далі йде умова, яка за аргументи бере поточний стан компонента `OrbitControl`, потім ми записуємо данні азимутального та полярного кута в дві змінні, після чого підбираємо потрібні значення кутів для виведення тексту, та змінюмо стан хуку `useState`, на потрібний нам текст. Після чого використовуючи компонент `HTML` він потрібен для того,

щоб в середині компонента Canvas можна було використовувати стандартні html теги , відображаємо текстове значення взяте з хуку useState (Рис. 3.30)

```
{displayText} && <Html><div className='model_container'><div className='model_text' dangerouslySetInnerHTML={{ _html: displayText }} /></div></Html>
```

Рис. 3.30 вивід тексту на екран

Тож в результаті рендерингу цього коду , ми отримали 3D модель на екрані , яка в залежності від того як поверне її користувач буде видавати різний текст , також має анімацію при завантаженні сторінки та може бути масштабована. (Рис. 3.31)



Рис. 3.31 Функціонал 3D моделі

Натупна сторінка в якій використовуються елементи 3D графіки це сторінка, окремим товаром. (Рис. 3.32)

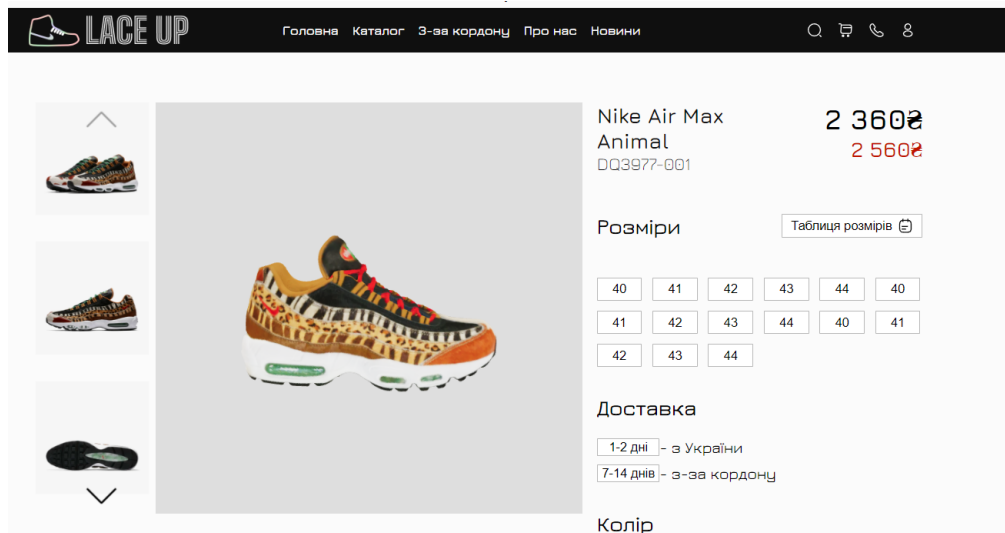


Рис. 3.32 Сторінка з використання 3D графіки

На картинці вище, тривимірний модель, розташована, по центру екрану як головна картинка з якою користувач може взаємодіяти: крутити на 360 та масштабувати. Код для цієї сторінки написаний по аналогії з попередніми, за виключенням того, що тут немає ніякої анімації, тільки компонент OrbitControls.

Пропоную розглянути код трішки детальніше спочатку я створив компонент канвас, в який і помістив 3D модель, освітлення, камеру та навколишнє середовище та за допомогою CSS стилів задав розміри полотну. (Рис. 3.33)

```
<div className="horizontal_container" style={{width: '606px',height:'584px'}} >
  <Canvas id='solo_shoe_threed' shadows>
    | <SoloShoe/>
  </Canvas>
</div>
```

Рис. 3.33 фрагмент коду створення компоненту Canvas

Наступний фрагмент коду показує налаштування камери, освітлення, колір навколишнього середовища та рендерить саму 3D модель. Із змін тут колір навколишнього середовища, я зробив його світлим, інтенсивність світла, світло я

щобив більш насиченим та розмір моделі зробив меншим , так як саме полотно має менші розміри (Рис. 3.34)

```
return (
  <>
  <PerspectiveCamera makeDefault position={[0, 1, 5]}/>
  <SoloShoeScene scale={1.5}/>
  <ambientLight args={["white" , 3]}/>
  <Environment background>
    <mesh scale={100}>
      <sphereGeometry args={[1, 64,64]}/>
      <meshBasicMaterial side={THREE.BackSide} color='#f2f2f2' />
    </mesh>
  </Environment>
</>
)
```

Рис. 3.34 Фрагмент коду компоненту Сцени

Перейдемо до налаштувань 3D моделі. За допомогою Хуку useGLTF завантажуюємо 3D об'єкт на нашу сцену , додаємо компонент OrbitControls , для того щоб користувач міг керувати камерою , та обертати її , за допомогою тегу mesh додаємо саму 3D модель взятую з Хуку useGLTF , додаємо до неї матеріали , та задаємо початкові координати , в даному випадку мені потрібно повернути об'єкт, щоб він відображався коректно.(Рис. 3.35)

```
export function SoloShoescene(props) {
  const { nodes, materials } = useGLTF('/Models/Airmax/scene-transformed.glb')
  return (
    <>
    <OrbitControls/>
    <group {...props} dispose={null}>
      <mesh geometry={nodes.defaultMaterial.geometry}
        material={materials['1001']}
        rotation={[-angleToRadians(90), 0, 0]}/>
    </group>
  </>
)
}

useGLTF.preload('/Models/Airmax/scene-transformed.glb')
```

Рис. 3.35 Фрагмент коду завантаження 3D об'єкта на сцену

## ВИСНОВОКИ

У ході цієї бакалаврської роботи було проведено глибоке дослідження та аналіз методів та інструментів для використання 3D-моделей у веб-дизайні. Метою дослідження було покращити інтерактивність та візуальну привабливість веб-сайтів.

Було вивчено різні формати 3D-моделей, методи та інструменти для їх інтеграції на веб-сайти. Здійснено порівняльний аналіз впливу 3D-моделей на користувацький досвід та візуальну привабливість веб-сайтів.

Для практичного дослідження було розроблено власний веб-сайт з використанням 3D-моделей. Цей сайт продемонстрував, як 3D-моделі роблять веб-сайти більш інтерактивними та цікавими, що може призвести до кращого залучення користувачів та конверсії.

Результати дослідження показали, що використання 3D-моделей на веб-сайтах може значно покращити їх якість та зробити їх більш ефективними.

Важливо зазначити, що для успішного використання 3D-моделей на веб-сайтах необхідно правильно обирати формати моделей, методи та інструменти для їх інтеграції, щоб не перевантажувати сайт та не погіршувати його продуктивність.

Дане дослідження може бути розширене в наступних напрямках:

- Дослідження нових методів та інструментів для інтеграції 3D-моделей на веб-сайти.
- Вивчення впливу 3D-моделей на емоційний стан користувачів та їх схильність до певних дій.
- Розробка рекомендацій щодо оптимального використання 3D-моделей на веб-сайтах з урахуванням специфіки галузі та цільової аудиторії.

Загалом, використання 3D-моделей у веб-дизайні є перспективним напрямком, який може значно покращити якість веб-сайтів та зробити їх більш ефективними.

## ПЕРЕЛІК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. Understanding the Basics of UI/UX Design: A Beginner's Guide *Medium*  
URL: <https://bootcamp.uxdesign.cc/understanding-the-basics-of-ui-ux-design-a-beginners-guide-11752310317a> .
2. Understanding UI/UX design concepts *Medium*  
URL : <https://medium.com/@bonadoze/understanding-ui-ux-design-concepts-315b62d06b27> .
3. 5 steps of the UI/UX design process every expert follows!. *Aufait UX*.  
URL: <https://www.aufaitux.com/blog/ui-ux-design-process/>
4. The Best Practices and Key Principles of UX Design – Baymard Institute. *Baymard Institute*. URL: <https://baymard.com/learn/ux-design-principles>
5. Як визначити цільову аудиторію – Блог Idea Digital. *IdeaDigital Agency*.  
URL: [https://ideadigital.agency/blog/yak-viznachiti-tsilovu-auditoriyu/#list\\_title3](https://ideadigital.agency/blog/yak-viznachiti-tsilovu-auditoriyu/#list_title3) .
6. Роман І. Методика 5W Шеррінгтона для сегментації аудиторії - *Elit-Web. elit-web.ua*. URL: <https://elit-web.ua/ua/blog/segmentatsija-auditorii-s-pomoschju-metodiki-5w-sherringtona>
7. What Is A Wireframe? Your Best Guide [With Examples]. *CareerFoundry*.  
URL: <https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/>
8. What is a UI Kit? Examples and Why You Should Use One | *Untitled UI. Untitled UI â Figma UI Kit and Design System*.  
URL: <https://www.untitledui.com/blog/what-is-a-ui-kit>
9. UI Components- What Are They? - *CometChat. In-app Chat SDK & API For Messaging And Calling - CometChat*. URL: <https://www.cometchat.com/tutorials/what-are-ui-components>
10. Atomic design components in UI design - *LogRocket Blog. LogRocket Blog*.  
URL: <https://blog.logrocket.com/ux-design/atomic-design-components-ui-design/>

11. UI Prototyping: Why Do It, Examples, and Common Mistakes. *Hotjar: Website Heatmaps & Behavior Analytics Tools*. URL: <https://www.hotjar.com/ui-design/glossary/prototype/>
12. What Is An Image Slider? | Elementor. *Elementor*. URL: <https://elementor.com/resources/glossary/what-is-an-image-slider/>
13. What is a 3D model? | Glossary | RenderThat. *Fotorealistische 3D-Produktvisualisierung | RenderThat*. URL: <https://www.renderthat.com/en/glossary/3d-model>
14. Contributors to Wikimedia projects. 3D scanning - Wikipedia. *Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/3D\\_scanning](https://en.wikipedia.org/wiki/3D_scanning)
15. Що таке фотограметрія? - Aero3D. *Aero3D*. URL: <https://aero3d.ua/shcho-take-fotogrametriia/>
16. What is 3D Digital Sculpting?. *Concept Art Empire*. URL: <https://conceptartempire.com/what-is-3d-sculpting/>
17. Nextech3D.ai. Best Practices to Embed 3D Models on Websites. *Nextech3D.ai: Your 3D Company*. URL: <https://www.nextechar.com/blog/best-practices-to-embed-3d-models-on-websites>
18. 10 Striking 3D Website Examples (and How Theyâre Made). *Vev Blog*. URL: <https://www.vev.design/blog/3d-website-examples/>
19. React – A JavaScript library for building user interfaces. *React – A JavaScript library for building user interfaces*. URL: <https://legacy.reactjs.org>
20. React Lifecycle. *W3Schools Online Web Tutorials*. URL: [https://www.w3schools.com/react/react\\_lifecycle.asp](https://www.w3schools.com/react/react_lifecycle.asp)
21. Built-in React Hooks – React. *React*. URL: <https://react.dev/reference/react/hooks>
22. What is CSS? - Learn web development | MDN. *MDN Web Docs*. URL: [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/What\\_is\\_CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS)



23. Nadeem A. Importing and Interacting with a 3D Model in React with React-Three-Fiber. *DEV Community*. URL: <https://dev.to/abdnadeem382/importing-and-interacting-with-a-3d-model-in-react-with-react-three-fiber-2fni>

24. Топ 3 бесплатных приложений для 3D-моделирования для начинающих - NEOR. *NEOR*. URL: <https://neor.ua/ru/novosti/top-3-besplatnyh-prilozhenij-dlya-3d-modelirovaniya-dlya-nachinayushhih/>

25. Лазерне 3D сканування: революція у цифровому моделюванні. *NNews*. URL: <https://nnews.com.ua/lazerne-3d-skanuvannya-revolyuciya-u-cifrovomu-modelyuvanni.html>

26. Аніматор чи 3d-моделлер? – i-nure. *Головна* – *I-NURE*. URL: <http://i.nure.ua/student/1842-animator-chi-3d-modeller>

27. Understanding the React Component Lifecycle. *Medium* URL: <https://medium.com/@arpitparekh54/understanding-the-react-component-lifecycle-a-deep-dive-into-the-life-of-a-react-component-74813cb8dfb5>

## Додаток А

# КОПІ ДЕМОНСТРАЦІЙНИХ МАТЕРІАЛІВ

Державний університет інформаційно-комунікаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

## «Розробка веб-сайтів з використанням 3D моделей»

на здобуття освітнього ступеня бакалавра  
зі спеціальності 126 Інформаційні системи та технології  
освітньо-професійної програми Інформаційні системи та технології

Виконав : Буджак Данило Віталійович  
Науковий керівник роботи:  
Кіс О.Я.

Кіїв - 2024

- **Актуальність теми:** 3D моделювання стрімко розвивається, і завдяки цьому їх використання на веб-сторінках стає все більш актуальним. Це робить дизайн сайтів динамічнішим та значно покращує досвід користувача.
- **Наукова новизна:** Досліджено вплив 3D графіки на веб-сайтах на користувачів, створено веб-сайт з унікальними 3D функціями
- **Об'єкт дослідження:** Розробка веб-сайтів з використанням 3D моделей
- **Предмет дослідження:** Веб-сайт та тривимірні об'єкти
- **Мета дослідження:** Метою цієї дипломної роботи є дослідження та аналіз використання 3D моделей на веб-сайтах з подальшою розробкою веб-сторінки
- **Завдання дослідження:**
  1. Дослідити та проаналізувати можливості використання 3D моделей на веб-сайтах.
  2. Розробити веб-сайт з використанням 3D моделей, який буде мати унікальні функції.
  3. Оцінити вплив використання 3D моделей на поведінку користувачів

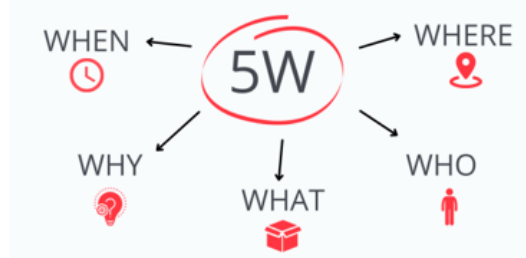
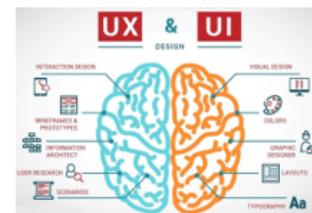
# 1. СТВОРЕННЯ UI/UX ДИЗАЙНУ

UI/UX – це процес створення інтерфейсів для сайтів , додатків , ігор тощо. Де UI – це візуальна частина , з якою взаємодіють користувачі.

UX – це те що відчуває користувач після користування інтерфейсом.

Аналіз ЦА – це процес визначення групи людей , що потенційно будуть користуватись цим сайтом.

Для аналізу ЦА я обрав метод “5W”

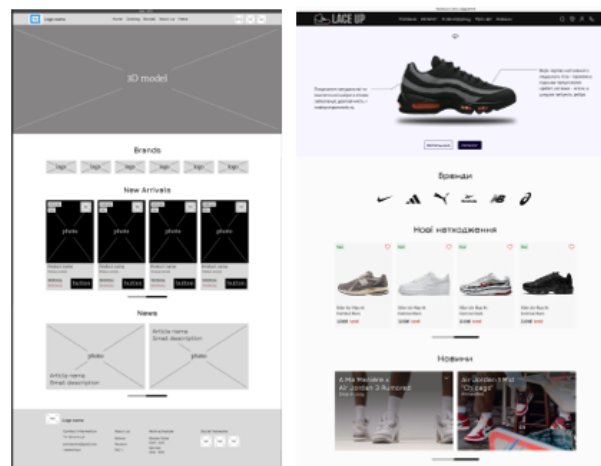


3

# 1. СТВОРЕННЯ UI/UX ДИЗАЙНУ

Wireframe – це чорновий макет дизайну який візуалізує структуру розміщення елементів на сторінці

Прототипізація - це процес створення функціональної моделі інтерфейсу користувача (UI) та користувацького досвіду (UX)



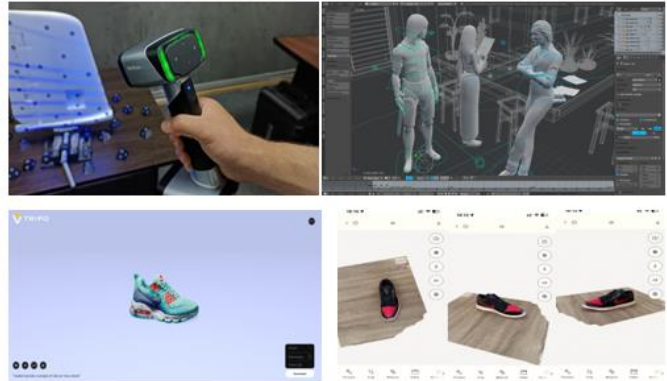
4

## 2. МЕТОДИ СТВОРЕННЯ 3D МОДЕЛЕЙ ТА ФУНКЦІЇ НА ВЕБ-САЙТАХ

3D модель – це сукупність полігонів , що з'єднані між собою , для того модель набула реального вигляду до неї додають текстури , кольори , матеріали , тощо.

Методи створення 3D моделей :

- Лазерне сканування
- Фотограметрія
- Sculpting
- За допомогою ШІ



5

## 2. МЕТОДИ СТВОРЕННЯ 3D МОДЕЛЕЙ ТА ФУНКЦІЇ НА ВЕБ-САЙТАХ

3D на веб-сайті допомагає:

1. Покращити привабливість
2. Додає інтерактивності
3. Збільшує впізнаваність бренду

Види 3D елементів на веб-сайті :

- Parallax фонове зображення
- Інтерактивний елемент.
- Вітрина товару



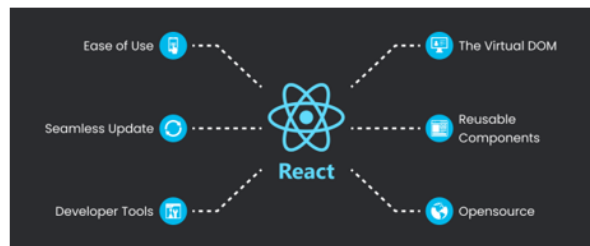
6

### 3. РОЗРОБКА FRONTEND ЧАСТИНИ ВЕБ-САЙТУ ТА ІМПЛЕМЕНТАЦІЯ 3D МОДЕЛІ

React js – це JavaScript-бібліотека для створення інтерфейсів користувача (UI). Вона розроблена компанією Meta (раніше Facebook) і використовується для створення односторінкових веб-додатків (SPA)

Основна концепція React це компоненти - будівельні блоки інтерфейсів користувача (UI) у React JS.

Вони дозволяють розбивати UI на менші, повторювані частини, що робить код більш організованим, модульним та простим у підтримці.



7

### 3. РОЗРОБКА FRONTEND ЧАСТИНИ ВЕБ-САЙТУ ТА ІМПЛЕМЕНТАЦІЯ 3D МОДЕЛІ

Код сайту реалізований на трьох основних сторінках сайту :

- Головна сторінка
- Каталог
- Сторінка з детальною інформацією про кросівки

3D модель знаходиться на Головній сторінці, та сторінці з детальною інформацією.

Також на кожній сторінці присутні перевикористані компоненти Меню, нижньої частини веб-сайту , та на деяких сторінках перевикористані слайдери.

```
Pages
# Catalog.css U
Catalog.jsx U
Landing.jsx U
ProductInfo.jsx U
Three.jsx U

<div className='catalog_list'>
  {card.map((cards)=>{
    <Card cards={cards}/>
  })}
</div>
```

```
function ProductSlider ({card}) {
  return(
    <div className='card_container'>
      <Swiper
        modules={[Navigation, Pagination, Scrollbar, A11y]}
        spaceBetween=8
        slidesPerView=4
        navigation
        scrollbar={{ draggable: true }}
        onSwiper=(swiper) => console.log(swiper)
        onSlideChange={() => console.log('slide change')}
        className='product_slider'
        style={{width: '1000px'}}
      >
        {card.map((cards) =>{
          <SwiperSlide key={cards.id}>
            <Card cards={cards}/>
          </SwiperSlide>
        })}
      </Swiper>
    </div>
  )
}
```

8

### 3. РОЗРОБКА FRONTEND ЧАСТИНИ ВЕБ-САЙТУ ТА ІМПЛЕМЕНТАЦІЯ 3D МОДЕЛІ

Для роботи з 3D моделями було використано 2 бібліотеки для React (`react-three/fiber` та `react-three/drei`)

Для того щоб додати модель на сторінку потрібно створити для неї сцену додати саму модель за допомогою влаштованих функцій у бібліотеку та обов'язково додати світло

```
const LandingShoe = () => {
  return (
    <>
      <PerspectiveCamera makeDefault position={[0, 1, 5]}/>
      <Airmax scale={2}/>
      <ambientLight args={['white', 1]}/>
      <Environment background>
        <mesh scale={100}>
          <sphereGeometry args={[1, 64, 64]}/>
          <meshBasicMaterial side={THREE.BackSide} color={'#add2ff'} />
        </mesh>
      </Environment>
    </>
  )
}
export default LandingShoe
```

```
const Three = () => {
  return (
    <div className='threeed_container' style={{width: '100%', height: '100vh'}}>
      <canvas id='landing_shoe_threeed' shadows />
      <LandingShoe />
    </div>
  )
}
export default Three
```

### 3. РОЗРОБКА FRONTEND ЧАСТИНИ ВЕБ-САЙТУ ТА ІМПЛЕМЕНТАЦІЯ 3D МОДЕЛІ

Для анімації була використана бібліотека GSAP , яка дає змогу управляти часовою смугою.

Анімацію я зроби для позиціонування моделі при завантаженні веб-сайту , вона має вигляд схожий на крок людини , це зроблено за допомогою зміни X та Y координат.

```
useEffect(()=>{
  if(!shoeRef.current){
    const timeline = gsap.timeline({paused:true})
    timeline.to(shoeRef.current.position, {
      x: 0,
      y:0,
      duration: 0.7,
      ease: "power1.out"
    })
    timeline.to(shoeRef.current.rotation, {
      y: angleToRadians(0),
      duration: 0.3,
    }, '<0.5')
    timeline.play()
  }
}, [shoeRef.current,])
```

## Висновки

- Досліджено методи імплементації 3D об'єктів на веб-сайт , інструменти які для цього потрібні та їх вплив на дизайн.
- Показано , що тривимірні об'єкти роблять сайт більш інтерактивним та цікавим .
- Встановлено , що 3D значно покращують якість веб-сторінок та роблять їх більш ефективними.
- Апробація

І Всеукраїнська науково-технічна конференція "Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу" , 28 листопада 2023 року , ДУІКТ – «Імплементація 3D моделей на веб-сайт»

V МІЖНАРОДНА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ» ,18 квітня 2024 року , ДУІКТ – «Сучасні методи розробки 3D моделей