

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

**Розробка веб-сайту для продажу дизайну інтер'єру та інтерактивного
огляду продукту у 3D**

Текстова частина до курсової роботи

за спеціальністю „Інженерія програмного забезпечення” 121

Керівник курсової роботи

Керівник курсової роботи

к.т.н., доц. Жежерун О.П.

(підпис)

“ ____ ” _____ 2021 р.

Виконав студент БП ІПЗ-4

Ревков Є.Г.

“ ____ ” _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики
ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем,
доцент, к.т.н. _____ Жежерун О.П.

(підпис)

«_____» _____ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

Студенту Ревкову Євгенію Геннадійовичу

Факультету інформатики 4 р.н. бакалаврської програми

ТЕМА: Розробка веб-сайту для продажу дизайну інтер'єру та інтерактивного
огляду продукту у 3D

Зміст текстової частини до курсової роботи:

Календарний план

Вступ

Розділ 1. Аналіз предметної області. Постановка завдання курсової
роботи

Розділ 2. Теоретичні відомості

Розділ 3. Опис реалізації програмного продукту

Висновки

Додатки(за необхідністю)

Список використаної літератури

Дата видачі «_____» _____ 2021 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Розробка веб-сайту для продажу дизайну інтер'єру та інтерактивного огляду продукту у 3D

Календарний план виконання роботи:

№	Назва етапу виконання	Термін	Примітка
1.	Отримання теми курсової роботи	30.10.2020	
2.	Пошук тематичної літератури	15.11. 2020	
3.	Ознайомлення з літературою	01.12. 2020	
4.	Вивчення аналогів	20.12. 2021	
5.	Планування веб-сайту	05.01. 2021	
6.	Ознайомлення з платформою Node.js	12.01. 2021	
7.	Ознайомлення з модулем Express	23.01. 2021	
11.	Реалізація клієнт-серверної частини	08.02. 2021	
12.	Оформлення сторінок	14.02. 2021	
13.	Написання текстової частини	21.03. 2021	
14.	Перегляд змісту роботи керівником	07.04. 2021	
15.	Внесення змін до курсової роботи відповідно до зауважень наукового керівника	09.04. 2021	
16	Створення презентації	10.04. 2021	

Студент Ревков Є.Г.

Керівник Жежерун О.П.

« ____ » _____

Зміст

Анотація	5
Перелік умовних позначень.....	6
Вступ.....	7
Актуальність, наукове та практичне значення обраної теми.....	7
Мета та завдання курсової роботи.....	7
Предмет дослідження.....	8
Використане програмне забезпечення	8
РОЗДІЛ 1 Аналіз предметної області. Постановка завдання курсової роботи.....	9
1.1 Аналіз сучасного стану питання та обґрунтування теми	9
1.2 Огляд існуючих аналогів розробки	9
1.3 Постановка задачі.....	10
РОЗДІЛ 2 Теоретичні відомості	10
2.1 React.js	10
2.2 PayPal Express Checkout	11
2.3 Three.js	12
РОЗДІЛ 3 Опис реалізації програмного продукту.....	16
3.1 Аналіз технічного завдання	16
3.1.1 Цільова аудиторія	16
3.1.2 Опис функціоналу сторінок та елементів інтерфейсу	17
3.2 Обґрунтування алгоритму й структури програми.....	28
3.3 Обґрунтування вибору засобів розробки.....	28
3.4 Створення об'єктів і розробка головної програми	34
3.4.1 Початкові налаштування	34
3.4.2 Налаштування маршрутизації	36
3.4.3 Реєстрація та авторизація	38
3.4.4 Middleware перевіряючий авторизацію	39
3.4.5 Middleware перевіряючий роль	40
3.4.6 Створення проекту на React	40
3.4.7 Маршрутизація на клієнтській стороні.....	40
3.4.8 Навігаційне меню	41
3.4.9 Відображення 3D моделі на сторінках.....	42
Висновки.....	46
Список використаної літератури та електронних ресурсів	47
Додатки	48

Анотація

Головною метою даної курсової роботи є створення веб-застосунку, який покращить процес вибору товару для покупця, завдяки можливості переглядання продукту у 3D.

Для виконання поставлених задач був використаний стек MERN. Вибір цього стеку був зумовлений зручністю у використанні. Також використовувалися такі бібліотеки, як `express.js`, `three.js`, `react-three-fiber`, `draco`, завдяки яким здійснювалось налаштування серверної частини та реалізації відображення 3d-моделей на веб-сайті.

Результатом роботи стало створення робочої версії веб-застосунку для продажу елементів дизайну інтер'єру, у якому при огляді продукту з'являється можливість його перегляду у тривимірному просторі.

Перелік умовних позначень

HTML – HyperText Markup Language

CSS – Cascading Style Sheets (Каскадні таблиці стилів)

DOM – (Document Object Model – «об’єктна модель документа»)

JS – JavaScript

СКБД – система керування базами даних

БД – база даних

MERN – M – MongoDB, E – Express.js, R – React.js, N – Node.js

JSX – JS + XML

XML – eXtensible Markup Language

SQL - Structured Query Language

SEO – Search Engine Optimization

ES6 – EcmaScript6

SaSS - Syntactically Awesome Stylesheets

SCSS - Sassy CSS

VS Code – VisualStudio Code

IDE – Integrated Development Environment

Вступ

Актуальність, наукове та практичне значення обраної теми

Тема даної курсової роботи – розробка веб-сайту для продажу дизайну інтер'єру та інтерактивного огляду продукту у 3D. Актуальність теми цієї курсової роботи в тому, що в нинішній час досить не малий відсоток людей вважають за краще робити покупки в інтернет-магазинах. В основному при виборі людиною будь-якого продукту в інтернет-магазині, не завжди достатньо однієї або пари фотографій, з огляду на те, що присутні фотографії не з усіх ракурсів. Особливо це стосується магазинів меблів, на сайті яких відображається в основному тільки передня частина продукту.

Саме тому був створений цей веб-застосунок, завдяки якому проблем з оглядом товару не буде зовсім.

Мета та завдання курсової роботи

Метою даної курсової роботи є створення веб-сайту, який поліпшить процес вибору товару для покупця, завдяки можливості переглядання продукту у 3D.

Завданням даної курсової роботи є розробка веб-застосунку розробка веб-сайту для продажу дизайну інтер'єру та інтерактивного огляду продукту у 3D використовуючи стек MERN та бібліотеку three.js.

Предмет дослідження

Вивчення та засвоєння нових навичок у роботі зі стеком MERN(M – MongoDB, E – Express.js, R – React.js, N – Node.js) та допоміжними інструментами Three.js, react-three-fiber.

Використане програмне забезпечення

- **Мови програмування**
 - HTML, CSS, JS
- **Фреймворки**
 - React
 - Bootstrap
- **Платформи**
 - Node.js
- **Основні бібліотеки**
 - Express.js
 - Three.js
 - React-three-fiber
 - Fort-awesome
- **База даних**
 - MongoDB
- **Редактори коду**
 - Visual Studio Code
 - WebStorm
- **Препроцесор**
 - ScSS

РОЗДІЛ 1 Аналіз предметної області. Постановка завдання курсової роботи

1.1 Аналіз сучасного стану питання та обґрунтування теми

Основна відмінність інтернет-магазину від традиційного – тип торгового майданчику. Інтернет-магазин – це сукупність програм, працюючих на веб-сайті, котрі дозволяють покупцю здійснювати дистанційно вибір товару з каталогу, переглянути його інформацію та зображення, оформити замовлення. На відміну від традиційного, функцію вітрини виконують сторінки, функцію товару – картки з його ілюстрацією, консультантів – інструкції та опис.

1.2 Огляд існуючих аналогів розробки

Найпоширенішим і популярним інтернет магазином є Розетка. Це досить зручний веб-сайт, завдяки якому можна легко і швидко знайти потрібний товар.

Розетка - це сайт на якому можна легко і швидко придбати товар, навіть не реєструючись, за наявності форми зворотного зв'язку на сторінці кошика, в якій можна залишити всього лише номер телефону.

Основні можливості сайту:

- Присутність великого каталогу товарів;
- Присутність великої кількості категорій;
- Можливість порівняти два і більше товарів за характеристиками;
- Можливість додавання товарів в категорію обрані;
- Наявність історії переглянутих товарів;
- Можливість залишити відгук про товар (тільки для авторизованих користувачів);
- Фільтрація товарів за ціною і за внутрішніми характеристиками

- Доступний перегляд галереї товарів

1.3 Постановка задачі

Розробити зручний, адаптивний під мобільні пристрої веб-сайт з продажу продукту дизайну інтер'єру, на якому клієнту будуть доступні такі основні можливості:

- Переглянути основну інформацію про веб-сайт;
- Мати можливість ознайомитися з товаром;
- Оформити покупку.

РОЗДІЛ 2 Теоретичні відомості

2.1 React.js

React – це JavaScript-бібліотека для створення користувацьких інтерфейсів.

Однією з особливостей React – використання JSX. Це комбінація коду JavaScript та XML, котрий надає простий та зрозумілий спосіб визначення коду візуального інтерфейсу.

Хуки - це функції, за допомогою яких ви можете під'єднатися до стану функціональних компонентів. Хуки не працюють у класі, так як вони дають можливість використовувати React без класів. Простий приклад використання хука, завдяки якому рендериться лічильник. При натисненні на кнопку значення лічильника буде інкрементовано (збільшено на один)

```
import React, { useState } from 'react';
```

```
function Example() {
```

```

const [count, setCount] = useState(0);

return (

  <div>

    <p>Ви натиснули {count} раз</p>

    <button onClick={() => setCount(count + 1)}>

      Натисни на мене

    </button>

  </div>

);
}

```

useState – це хук. Він викликається для того, щоб надати функціональному компоненту внутрішній стан, який буде зберігатися між рендерами. Виклик useState повертає масив з двома елементами, котрий включає в себе: нинішній стан та функцію для його оновлення.

Тому і було вирішено при розробці використовувати функціональні компоненти, які побудовані на основі хуків.

2.2 PayPal Express Checkout

PayPal Express Checkout допомагає збільшити обсяги продажів, надаючи своїм клієнтам можливість здійснювати оплату кредитною картою або з безпеки своїх особистих рахунків PayPal. Під час оформлення замовлення клієнт перенаправляється на безпечний сайт PayPal для заповнення інформації про платіж. Потім клієнта повертають до вашого магазину для завершення решти процесу оформлення замовлення. Вибір

Express Checkout додає у ваш магазин знайому кнопку PayPal, яка, як повідомляється, збільшує продажі.

Клієнти, які мають поточні рахунки PayPal, можуть здійснити покупку за один крок, натиснувши кнопку Оформити з PayPal. Express Checkout можна використовувати як самостійно, так і в поєднанні з одним із рішень PayPal «все в одному». Якщо ви вже приймаєте кредитні картки через Інтернет, ви можете запропонувати Express Checkout як додаткову можливість залучити нових клієнтів, які вважають за краще платити за допомогою PayPal.

2.3 Three.js

Як відомо Three.js полегшує роботу с WebGL. При використанні цієї бібліотеки відпадає необхідність в написанні шейдерів, та з'являється можливість оперувати звичними поняттями.

Моделювання графіки з використанням three.js можна порівняти зі знімальним майданчиком, так як у нас є можливість оперувати такими поняттями як сценарій, світло, камеру, об'єкти та їх матеріали.

Три основні засоби для використання:

- Scene – платформа, на якій розміщуються всі об'єкти, котрі створюються
- Camera – це «око», котре направлене на сцену. Камера знімає та відображає об'єкти, котрі знаходяться на сцені
- Renderer – візуалізатор, котрий дозволяє відображати сцену, зняту на камеру

В Three.js існує декілька типів камери:

- Cube Camera
- Stereo Camera
- Perspective Camera.

- Orthographic Camera.

Найрозповсюдженіші – Orthographic Camera та Perspective Camera(Рис. 2.1)

Perspective Camera

Перспективну камеру використовують для імітації того, що бачить людське око. Камера приймає всі об'єкти в перспективній проекції, тобто: чим далі знаходиться об'єкт від нас, тим він здається меншим.

Orthographic Camera

В цьому режимі проектування об'єкт у відображеному зображенні залишається статичним, незалежно від відстані камери. Це може бути корисним для відображення елементів UI та 2D сцен.

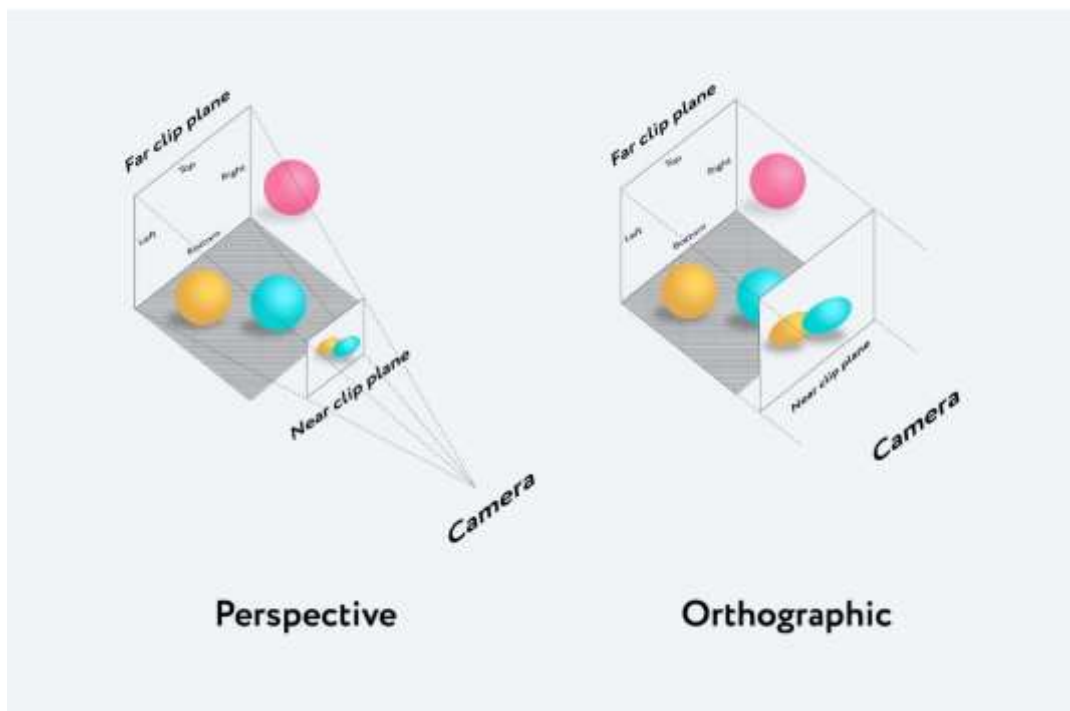


Рис. 2.1 Orthographic Camera та Perspective Camera

Освітлення

Завдяки освітленню сцени, відпадає враження знаходження в темному просторі. З допомогою освітлення сцени можна надати велику реалістичність. Окрім цього, освітленню можна задати колір.

Приклади типів освітлення(Рис. 2.2):

- Ambient Light - фонове освітлення, яке використовується для освітлення всіх об'єктів сцени однаково. Відсутня можливість використання для створення тіні, оскільки цей тип освітлення не має напрямлення;
- Point Light – світло, що випромінюється з однієї точки в усіх напрямках. Звичайний випадок використання такого освітлення це повторення освітлення від простої лампочки;
- Spot Light - дане світло випромінюється з однієї точки в одному напрямку, вздовж конуса, розширеного в міру віддалення від джерела світла;
- Directional Light - світло, що випромінюється в певному напрямку. Це світло буде вести себе так, як якщо б воно було нескінченно далеко, а промені, що випромінюються з нього, були паралельні.

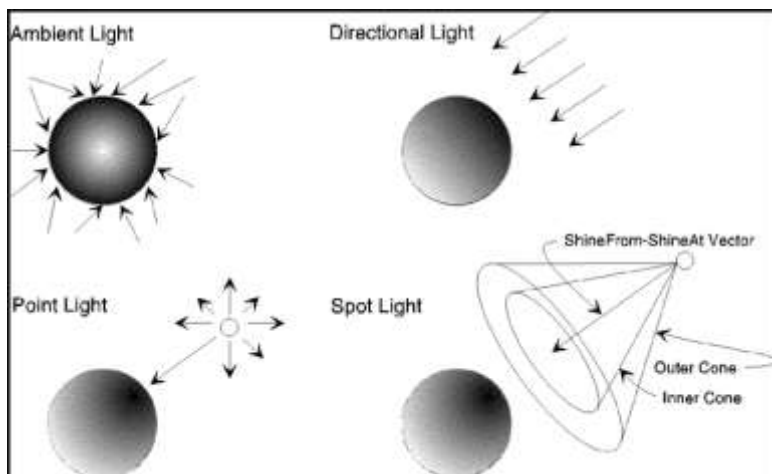


Рис 2.2 Типи освітлення

Створення об'єкту на сцені

Об'єкт, що створюється на сцені називається Mesh. Mesh – це клас, котрий являє собою об'єкти полігональної сітки.

Цей клас приймає два аргументи:

- Material – описує зовнішній вид об'єктів (текстура, колір, тощо);
- Geometry – описує форму (положення вершин, радіус, грані, тощо).

GLTF

Gltf – це специфікація відкритого формату для ефективної доставки та завантаження 3D-вмісту. Моделі можуть надаватися у форматі JSON(.gltf), або у двійковому форматі (.glb). Зовнішні файли зберігають текстури у форматі .jpg, .jpeg та .png, та додаткові двійкові дані у форматі .bin (Рис. 2.3).

Обрані моделі було взято з сайту <https://sketchfab.com>

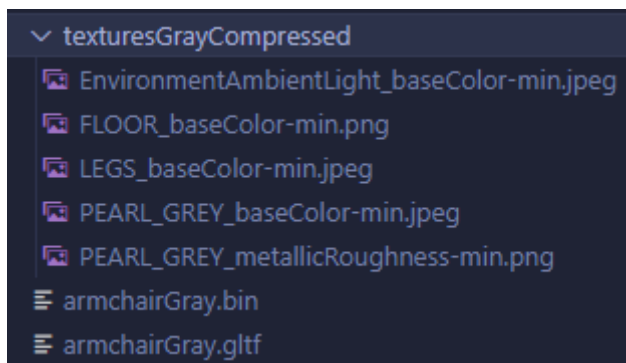


Рис. 2.3 Вміст необхідних даних для завантаження моделі

OrbitControls

Орбітна камера - це та камера, яка дозволяє користувачу обертатись навколо центральної точки, але при цьому фіксована вісь залишається заблокованою. Це надзвичайно зручно, оскільки запобігає «нахилу» сцени поза віссю. Ця версія блокує вісь Y (вертикальна) і дозволяє користувачам

здійснювати орбіту, масштабування та панорамування лівою, середньою та правою кнопками миші (або певними подіями дотику).

Також сценарій OrbitControls має кілька налаштувань, які можна додати або змінити:

```
controls = new THREE.OrbitControls( camera, renderer.domElement );
```

```
controls.enableDamping = true;
```

```
controls.dampingFactor = 0.25;
```

```
controls.enableZoom = true;
```

```
controls.autoRotate = true;
```

React-three-fiber

Завдяки цій бібліотеці з'являється можливість змусити моделі безперервно обертатись, а також рухатись вгору-вниз з певною частотою. Для цього нам потрібно імпортувати useFrame з react-three-fiber. Це важливий хук, який дозволяє застосовувати обчислення, які потрібні при кожному обчисленні кадру. Для реалізації цього хука, потрібно додати наступні стрічки коду в компонент, в якому зберігається модель:

```
const ref = useRef();
```

```
useFrame(() => (ref.current.rotation.y += 0.01));
```

РОЗДІЛ 3 Опис реалізації програмного продукту

3.1 Аналіз технічного завдання

3.1.1 Цільова аудиторія

Розроблений сайт призначений для людей, які надають перевагу здійснювати покупки дистанційно.

3.1.2 Опис функціоналу сторінок та елементів інтерфейсу

Навігаційне меню:

- Має бути присутнім на всіх сторінках;
- Для неавторизованого користувача містить посилання на головну сторінку, сторінку перегляду товарів, сторінку реєстрації/авторизації (Рис. 3.1);
- Для адміністратора містить посилання на головну сторінку, сторінку перегляду товарів, сторінку створення категорій товару, сторінку додавання товару, кнопку “Logout” (Рис. 3.2);
- Для авторизованого користувача містить посилання на головну сторінку, сторінку перегляду товарів, сторінку історії покупок, сторінку кошика, кнопку “Logout” (Рис. 3.3).



Рис. 3.1 Навігаційне меню неавторизованого користувача



Рис. 3.2 Навігаційне меню адміністратора



Рис. 3.3 Навігаційне меню авторизованого користувача

Футер: (Рис. 3.4)

- Містить посилання на сторінки;
- Містить посилання на соціальні мережі;
- Містить посилання на сторінку підтримку.

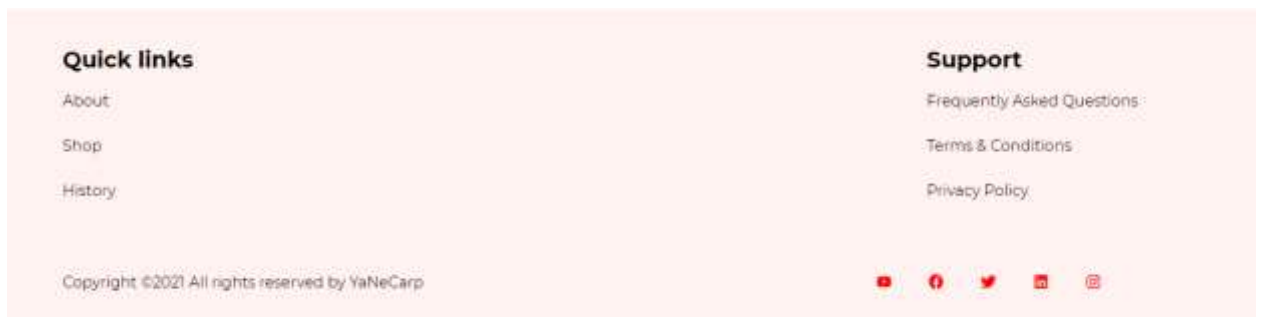


Рис. 3.4 Секція-футер

Головна сторінка:

- Містить на початку баннер, в якому є посилання на сторінку товарів;
- Секція «Особливості магазину»;
- Секція-галерея «Роботи»;
- Секція «Про нас»;
- Секція-галерея з продуктами, які можна оглянути в 3D перейшовши за посиланням при наведенні на зображення.

Сторінка «Товари»

- Для авторизованих та неавторизованих користувачів: (Рис. 3.5)
 - Останні 9 товарів, які були додані адміністратором;
 - Система пошуку товарів на сайті за назвою;
 - Система пошуку товарів на сайті за категорією;

- Сортуння товарів в алфавітному порядку, за ціною і популярністю;
- Картки з товарами;
- Зображення товару на картці;
- Назва товару на картці;
- Опис товару на картці;
- Ціна товару на картці;
- Кнопка «Buy», що додає товар у корзину у випадку, коли користувач авторизований;
- Кнопка «View», яка перенаправляє на сторінку обраного товару;
- Кнопка «Load More», яка дає можливість передивитися наступні 9 товарів.

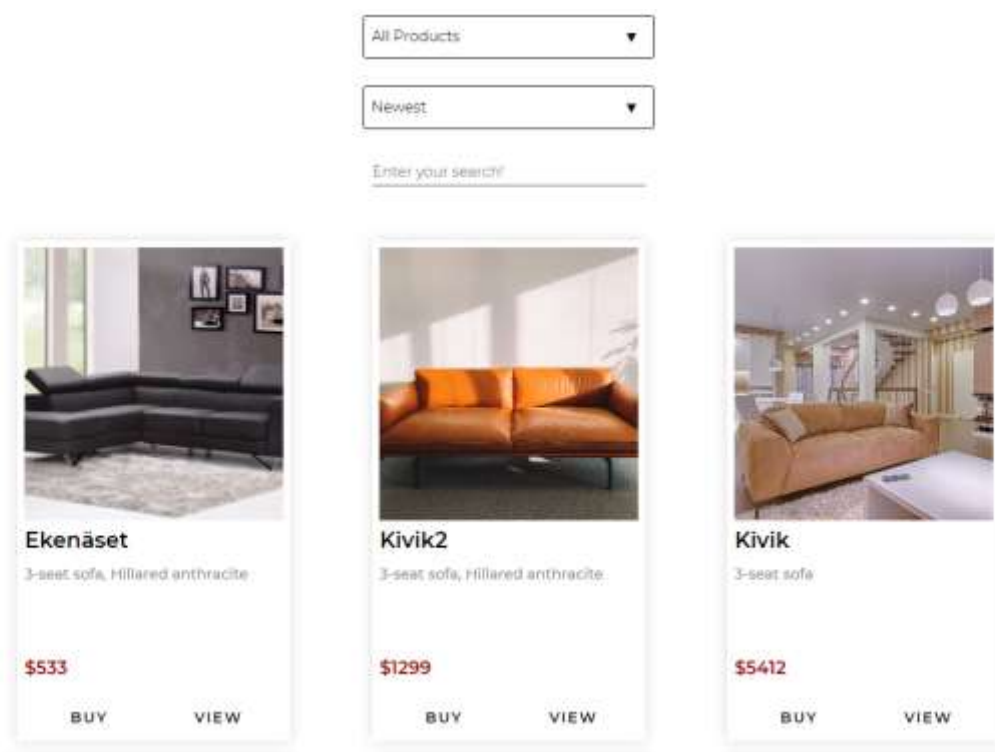


Рис 3.5 Сторінка «Товари» для авторизованих користувачів

- Для адміністратора: (Рис. 3.6)
 - Останні 9 товарів, які були додані адміністратором;
 - Система пошуку товарів на сайті за назвою;
 - Система пошуку товарів на сайті за категорією;

- Сортування товарів в алфавітному порядку, за ціною і популярністю;
- Картки з товарами;
- Зображення товару на картці;
- Назва товару на картці;
- Опис товару на картці;
- Ціна товару на картці;
- Кнопка «Delete», яка видаляє товар з бази даних;
- Кнопка «Edit», яка перенаправляє на редагування товару;
- Кнопка «Load More», яка дає можливість передивитися наступні 9 товарів;
- Кнопка «Select All», завдяки якій обираються всі товари;
- Кнопка «Delete Selected», яка дозволяє видалити обрані товари.

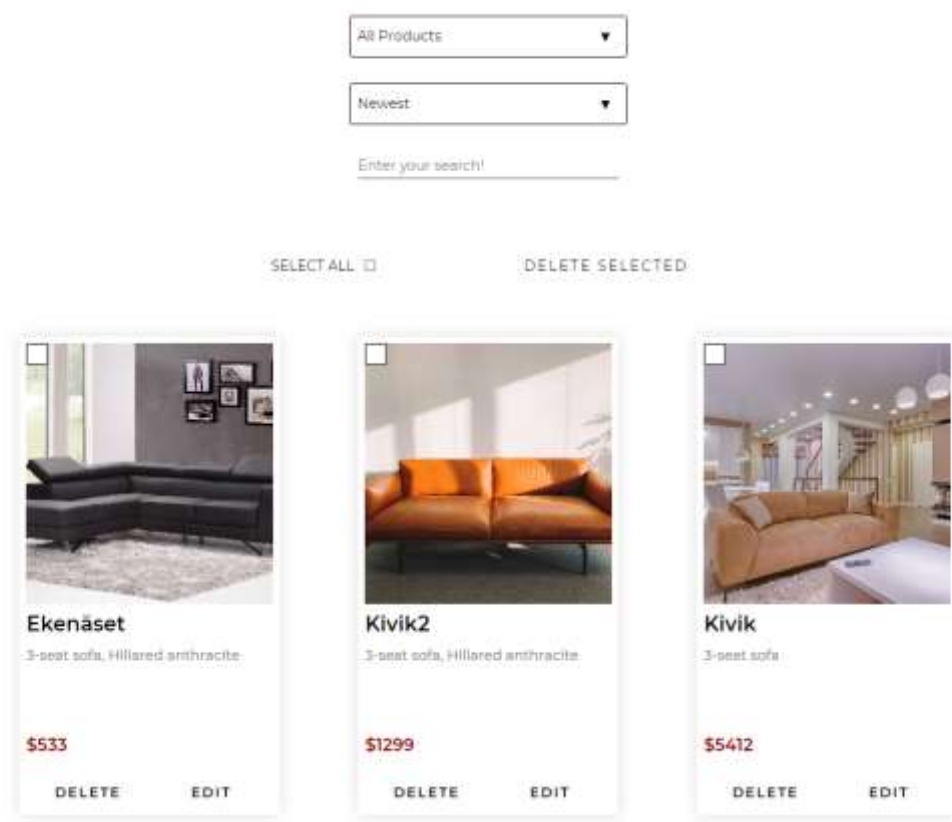


Рис 3.6 Сторінка «Товари» для адміністратора

Сторінка «Створити товар»

- Тільки для адміністратора: (Рис. 3.7)
 - Назва сторінки;
 - Додати фотографію товару;
 - Вказати ID товару;
 - Вказати назву товару;
 - Вказати ціну товару;
 - Вказати опис товару;
 - Вказати текст, який прикріплений до товару;
 - Обрати категорію товару;
 - Кнопка «Create», яка відповідає за створення проекту.



Рис 3.7 Секція «Створити товар»

Сторінка «Створити категорію» (Рис. 3.8)

- Тільки для адміністратора:
 - Назва сторінки;
 - Поле для назви категорії;
 - Список існуючих категорій;
 - Можливість видалити категорії, натиснувши кнопку «Delete»;

- Можливість змінити назву категорії, натиснувши кнопку «Edit», після якої у полі для назви категорії з'являється її назва, яку можна зберегти, натиснувши кнопку «Update».

Create Category

CATEGORY

CREATE

Chairs	EDIT DELETE
Sofas	EDIT DELETE
Shelfs	EDIT DELETE
Tables	EDIT DELETE

Рис 3.8 Секція «Створити категорію»

Сторінка «Історія» (Рис. 3.9)

- Для адміністратора:
 - Назва сторінки;
 - Кількість всіх замовлень;
 - Табличка, в якій присутні: ID замовлення, дата оформлення замовлення, кнопка перегляду замовлення «View» .
- Для авторизованого користувача:
 - Назва сторінки;
 - Кількість замовлень користувача;
 - Табличка, в якій присутні: ID замовлення, дата оформлення замовлення, кнопка перегляду замовлення «View» .

HISTORY			
YOU HAVE 5 ORDERED			
Payment ID		Date Of Purchased	
PAID-HBPSN7V6T2733W7C2EL		27.03.2021	View
PAID-HBPSN7V6T2733W7C2EL		27.03.2021	View
PAID-HBPSN7V6T2733W7C2EL		27.03.2021	View
PAID-HBPSN7V6T2733W7C2EL		27.03.2021	View
PAID-HBPSN7V6T2733W7C2EL		27.03.2021	View

Рис 3.9 Секція «Історія замовлень»

Сторінка «Історія/id_замовлення» (Рис. 3.10)

- Для адміністратора та авторизованого користувача:
 - Ім'я замовника;
 - Адреса замовника;
 - Поштовий індекс;
 - Код країни;
 - Табличка, в якій присутні: зображення товару, назва товару, кількість замовленого товару, загальна ціна товару відносно його кількості;

Name	Address	Postal Code	Country Code
John Doe	Free Trade Zone - Santiago	950215	CL
	Products	Quantity	Price
	Product 03	1	\$ 123
	Product 05	1	\$ 112
	Product 04	1	\$ 15

Рис 3.10 Секція «Замовлення»

Сторінка «Товар» (Рис. 3.11)

- Для авторизованого та неавторизованого користувача
 - Зображення товару;
 - Назва товару;
 - Ціна товару;
 - Опис товару;
 - Текст, який прикріплений до товару;

- Кількість продаж товару;
- Кнопка «Купити», яка додає товар у кошик, та редіректить на сторінку «Кошик» та яка змінює кількість товару біля іконки кошика, яка знаходиться у навігаційному меню ;
- «Related Product» - секція, яка знаходиться після основного поля товару, в якій знаходяться товари тієї ж категорії, що і обраний товар.



Рис 3.11 Секція «Товар»

Сторінка «Кошик» (Рис. 3.12)

- Тільки для авторизованого користувача:
 - Зображення обраного товару;
 - Назва товару;
 - Опис товару;
 - Текст, який прикріплений до товару;
 - Кількість товару з можливістю змінювати його кількість;
 - Кнопка, яка дозволяє видалити товар з кошику;
 - Загальна ціна замовлення;
 - Кнопка «Оформити покупку».

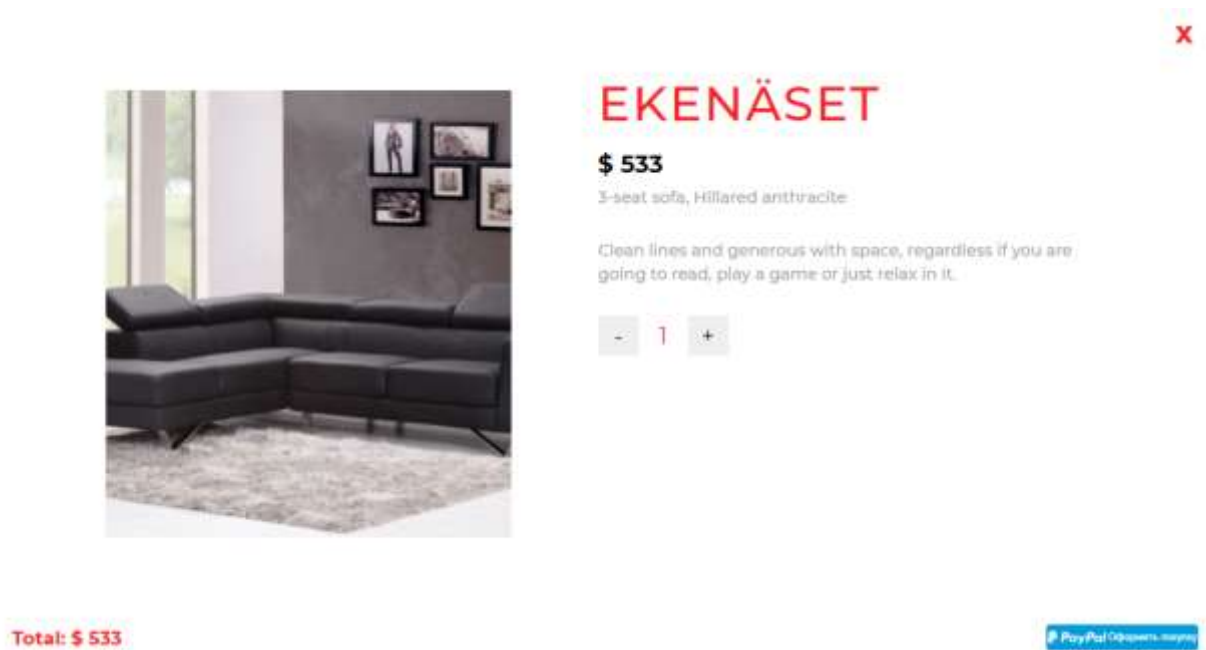


Рис. 3.12 Секція «Кошик»

Сторінка «Унікальний товар» (Рис. 3.13)

- Для всіх користувачів:
 - Назва товару;
 - Опис товару;
 - Деталі товару;
 - Товар, який можна спостерігати у 3D просторі з можливістю обертати та приближувати його. Модель товару обертається відносно осі Y тривимірної площини;
 - Додаткові зображення товару.

EKOLSUND

EKOLSUND recliner has soft rounded shapes which are sized to fit in even small spaces. Adjustable in three positions – so choose whether you want to sit upright, lean back or lie back for a nap.

Product Details

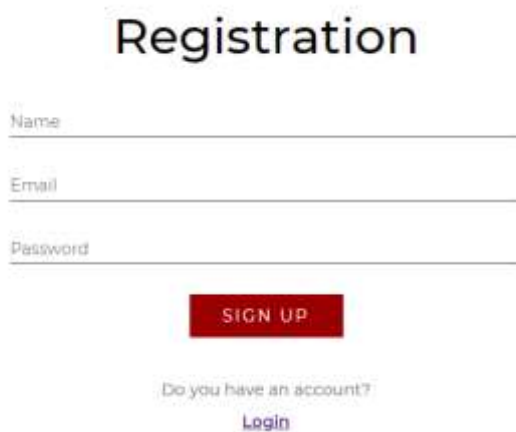
You can set this reclining armchair in 3 positions – upright, lean back and full recline – making it easy to adapt to various activities at home. The cover is easy to keep clean since it is removable and can be machine washed, while the perfect fit makes it look just as neat as a fixed cover.



Рис. 3.13 Секція «Унікальний товар»

Сторінка реєстрації (Рис. 3.14)

- Для неавторизованих користувачів:
 - Назва сторінки;
 - Поле для вводу ім'я;
 - Поле для вводу пошти;
 - Поле для вводу пароля;
 - Кнопка SignUp для закінчення реєстрації;
 - Посилання на авторизацію, якщо у користувача вже присутній акаунт.



Registration

Name

Email

Password

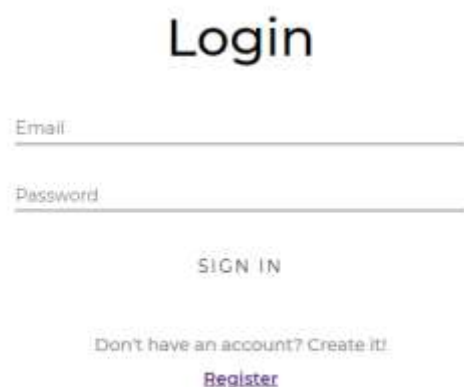
SIGN UP

Do you have an account?
[Login](#)

Рис. 3.14 Секція «Реєстрація»

Сторінка авторизації (Рис. 3.15)

- Для неавторизованих користувачів:
 - Назва сторінки;
 - Поле для вводу електронної скриньки;
 - Поле для вводу пароля;
 - Кнопка SignIn для закінчення авторизації;
 - Посилання на реєстрацію, якщо у користувача відсутній акаунт.



Login

Email

Password

SIGN IN

Don't have an account? Create it!
[Register](#)

Рис. 3.15 Секція «Авторизація»

Безпека

1. Хешування паролів;
2. Фільтрація вхідних значень;
3. Історія авторизацій.

Дизайн та зовнішній вигляд

1. Анімація при наведенні на елементи;
2. Адаптивність під мобільні пристрої;
3. Кросбраузерність;

3.2 Обґрунтування алгоритму й структури програми

Для створення програми був використаний стек MERN. Стек MERN – це JavaScript-стек, котрий розроблений для поліпшення процесу розробки. Цей стек складається з чотирьох компонентів з відкритим вихідним кодом: MongoDB, Express, React, Node.js.

Основною перевагою використання цього стека є те, що кожна стрічка коду написана на мові програмування JavaScript. Ця мова використовується як для коду на стороні клієнта, так і для коду на стороні сервера.

3.3 Обґрунтування вибору засобів розробки

React.js

React - це JavaScript-бібліотека для розробки користувацького інтерфейсу. Цей фреймворк можна використовувати для розробки односторінкових, багатосторінкових та мобільних додатків. Його ціль - запропонувати високу швидкість, простоту та масштабість. У якості

бібліотеки для розробки користувацьких інтерфейсів React часто використовується з такими бібліотеками, як MobX, Redux, GraphQL.

Переваги використання React:

- В основі прості мови програмування.
- Надзвичайна гнучкість програми.
- Використання DOM.
- Додаток витримує великі навантаження.
- React і SEO добре ладнають. Пошуковим роботам простіше переглядати сайти, поліпшується взаємодія користувачів з Вашим ресурсом.
- Забезпечує незмінність батьківських даних.
- Має відкриту бібліотеку даних.
- Невелика вага бази даних
- Забезпечує просту міграцію між версіями.
- Гібридні мобільні додатки на React зовні майже не відрізняються від нативних.

Bootstrap

Для адаптивної верстки використовувався Bootstrap, тому що якщо говорити про фіксовані макети, то їх легко створювати навіть з нуля. Просто створювати блоки, задавати їх фіксовану ширину.

Після процесу вивчення цього CSS-фреймворку, він сильно спрощує для проекту верстку. Фреймворк бере на себе кросбраузерність та адаптивність, а це є основними речами у процесі створення верстки. З bootstrap це реалізувати дуже просто. Це дозволяє створювати html-шаблон навіть людині, котра раніше дуже мало часу займалася версткою і не дуже знайома з CSS.

Було вирішено обрати цей фреймворк за причини хорошої, зрозумілої документації, великої кількості прикладів та простоти у використанні.

Node.js

Node.js - це програмна платформа, яка робить JavaScript мовою загального призначення. Також називається середовищем виконання JavaScript. Платформа вміє зв'язуватися з зовнішніми бібліотеками, викликати команди з коду і виконувати роль веб-сервера. Простіше, цей інструмент додає до повністю фронтендової мови бекендну частину, дозволяючи створювати з його допомогою не тільки веб-сайти, а й повноцінні програми, без задіяння браузера.

Основні причини використання:

- Присутня можливість застосовувати одну мову на клієнті та сервері.
- Швидкість. Створення робочого прототипу не віднімає багато часу. Перший етап створення проекту відбувається дуже швидко. Якщо архітектура дуже добре продумана, то в майбутньому не виникне складностей з тим, щоб розширити сайт на Node.js також не з'явиться.
- Синтаксис JavaScript. Це забезпечує відносну простоту і доступність.

WebStorm

Програмне забезпечення JetBrains WebStorm являє собою інструмент для розробки web-сайтів і редагування HTML, CSS і JavaScript коду. Рішення забезпечує швидку навігацію по файлах і генерує повідомлення про виникаючі проблеми в коді в режимі реального часу. Також це ПЗ дозволяє додавати розмітку HTML-документів або елементів SQL безпосередньо в JavaScript.

Нове в версії WebStorm 9:

- Підтримка фреймворків Meteor та React.
- Підтримка модуля EcmaScript 6.
- Удосконалення CSS3.
- Підтримка останніх версій Stylus і Sass.
- Функція, що відображає значення змінних безпосередньо в редакторі.

Visual Studio Code

Visual Studio Code - редактор вихідного коду, розроблений Microsoft для Windows, Linux і macOS. VS Code не є IDE. Це зручний, простий і доступний кодовий редактор з розширеними можливостями.

Основні можливості:

- Розширення. Зручна система додавання розширень. Щоб почати користуватися ними, достатньо просто перейти на вкладку з розширеннями та вписати в пошуковій стрічці необхідне доповнення.
- Git. Зручна робота з системою контролю версій. Зручне відображення комітів, гілок, історій тощо.
- Пошук. Розумний пошук з можливістю заміни за різними критеріями.
- Термінал. Можна працювати через неї з Git і налагоджувати додаток.
- Налаштування. Велика кількість налаштувань. Застосовуються моментально, перезапускати редактор не потрібно.
- Сніпети. Сніпети коду - це фрагменти коду, які ми можемо багаторазово використовувати в наших проектах
- IntelliSense. IntelliSense - загальний термін для різних функцій редагування коду, включаючи: заповнення коду, інформацію про параметри, швидку інформацію та списки членів. Функції IntelliSense іноді називають іншими іменами, такими як "автозаповнення коду".

Postman

Postman — зручний HTTP-клієнт для тестування веб-застосунків. За допомогою розширення можна складати і редагувати прості або складні HTTP-запити. Складені запити автоматично зберігаються на майбутнє для повторного застосування. Відповіді від сервера теж зберігаються як файли на жорсткому диску. Таким чином, Postman зекономив велику купу часу при надсиланні запитів на сервер при розробці проекту.

Three.js

Three.js – це бібліотека JavaScript, котра включає в себе набір готових класів для відображення та створення інтерактивної 3D графіки в WebGL.

WebGL – це бібліотека для JavaScript, яка дозволяє створювати 3D графіку, котра функціонує в браузері. Дана бібліотека заснована на архітектурі бібліотеки OpenGL. У WebGL код моделюється безпосередньо в браузері. Для цього WebGL використовує об'єкт Canvas, який був введений в HTML5.

Щоб все візуалізувати (кожну лінію, кожну точку, грань і т.д) необхідно прописати доволі великий об'єм коду, але завдяки бібліотеці three.js, цей процес значно прискорюється.

Express.js

Express являє собою популярний веб-фреймворк, написаний на JavaScript і працює всередині середовища виконання Node.js. Цей модуль висвітлює деякі ключові переваги цього фреймворка, установку середовища розробки і виконання основних завдань веб-розробки і розгортання. Його

можна використовувати різноманітними варіантами шаблонних мов(наприклад, EJS, Pug, Jade). Також можна використовувати базу даних, наприклад MongoDB з mongoose, щоб забезпечувати бекендну частину для додатку Node.js. Express.js в основному допомагає керувати всім: від маршрутів до обробки запитів.

База даних MongoDB

MongoDB — класичний приклад NoSQL СКБД. В основному використовується при роботі з великим об'ємом напівструктурованими і неструктурованими даними. Це документоорієнтоване рішення з відкритим вихідним кодом, яке використовує для зберігання структуру JSON.

MongoDB відрізняється швидкою роботою з даними, кешуванням даних, високою доступністю і можливостями аналітики в реальному часі.

Препроцесор SaSS

SaSS є одним з найбільш широко використовуваних CSS-препроцесорів. Він має різні функції, що допомагають розробникам писати CSS-код краще і чистіше.

SaSS – це один з найбільш широко використовуваних CSS-препроцесорів.

Він має велику кількість різноманітних функцій, які допомагають прописувати код на CSS краще та чистіше.

Оскільки проект стає більше, CSS викликає деякі проблеми, такі як:

- Великі зусилля для внесення невеликих змін;
- Труднощі при структуруванні коду;
- Надмірність коду;

- Нескінченні рядки CSS-класів і правил.

Відмінність ScSS від SaSS полягає в тому, що ScSS більше схожий на звичайний CSS код.

Приклад SaSS-коду:

```
$blue: #3b23ee

.content-navigation

border-color: $red

color: lighten($red, 11%)
```

Приклад ScSS-коду:

```
$blue: #3b23ee;

.content-navigation{

border-color: $red;

color: lighten($red, 11%);

}
```

Особливість ScSS

- Змінні;
- Вкладеність;
- Mixins;
- Імпорти.

3.4 Створення об'єктів і розробка головної програми

3.4.1 Початкові налаштування

Після створення файлу `server.js` виконувався процес встановлення ряду необхідних пакетів таких як:

- Express.js – фреймворк для створення веб-додатків. У нього вбудований функціонал для вирішення багатьох проблем, зокрема, для налагодження маршрутизації.
- Mongoose. Це адміністратор об'єктних даних (ODM), що забезпечує взаємодію між додатком express.js і базою даних MongoDB.
- dotenv. Дозволяє використовувати файли з розширенням .env для роботи з конфіденційними даними
- bcrypt. Дає можливість шифрування конфіденційних даних, наприклад, паролів
- Nodemon. Завдяки Nodemon, можна не зупиняти і не перезапускати сервер після внесення будь-яких змін.

```
require("dotenv").config();
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const fileUpload = require("express-fileupload");
const cookieParser = require("cookie-parser");
const path = require("path");

const app = express();
app.use(express.json());
app.use(cookieParser());
app.use(cors());
app.use(
  fileUpload({
    useTempFiles: true,
  })
);

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {
  console.log(`Server started on ${PORT} port`);
});
```

Наступним кроком було підключення бази даних до серверу. В якості бази даних була використана MongoDB. Команда для встановлення необхідного пакету : `npm install mongoose --save`.

В папку `models` потрібно додати файл `userModel.js`, у якому зберігається визначення використовуваної моделі.

```
const userSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    trim: true,
  },
  email: {
    type: String,
    required: true,
    unique: true,
  },
  password: {
    type: String,
    required: true,
  },
  role: {
    type: Number,
    default: 0,
  },
  cart: {
    type: Array,
    default: [],
  },
}, {
  timestamps: true,
});
module.exports = mongoose.model("Users", userSchema);
```

Тепер модель користувача являє собою модель `mongoose`, котра визначається за допомогою схеми `userSchema`. І таким чином були створені моделі для продукту, оплати та категорій відповідно.

3.4.2 Налаштування маршрутизації

Маршрутизація визначає, як додаток відповідає на клієнтський запит до конкретної адреси (URI).

Код, наведений нижче, є прикладом маршрутів

```
// Routes
app.use("/user", require("./routes/userRouter"));
app.use("/api", require("./routes/categoryRouter"));
app.use("/api", require("./routes/upload"));
app.use("/api", require("./routes/productRouter"));
app.use("/api", require("./routes/paymentRouter"));
```

Express підтримує велику кількість методів маршрутизації відповідні методам HTTP: get, post, put, delete, options тощо. Завдяки методу route були здійснені маршрутизації, визначені для методів запитів GET, POST, PATCH до заданого каталогу програми.

Для спрощення логічної організації маршрути, які обробляються контролером userController, організовані в окремий роутер - userRouter.

userRouter.js:

```
router.post("/register", userCtrl.register);
router.post("/login", userCtrl.login);
router.get("/logout", userCtrl.logout);
router.get("/refresh_token", userCtrl.refreshToken);
router.get("/infor", auth, userCtrl.getUser);
router.patch("/addcart", auth, userCtrl.addCart);
router.get("/history", auth, userCtrl.history);
```

Наступний крок – створення контролерів, в які буде внесена логіка обробки. Для цього в каталозі створюється папка з назвою controllers, в яку додаємо нові файли userController.js, productController.js, paymentController.js та categoryController.js. Фактично кожен файл представляє окремий контролер, який містить набір функцій-обробників маршрутів.

Контролери підключаються як стандартні модулі, і їх функції використовуються для обробки маршрутів.

```

const Users = require("../models/userModel");
const Payments = require("../models/paymentModel");
const bcrypt = require("bcrypt");
const jwt = require("jsonwebtoken");

const userCtrl = {
  register: async (req, res) => { ...
  },
  login: async (req, res) => { ...
  },
  logout: async (req, res) => { ...
  },
  refreshToken: (req, res) => { ...
  },
  getUser: async (req, res) => { ...
  },
  addCart: async (req, res) => { ...
  },
  history: async (req, res) => { ...
  },
};

const createAccessToken = (user) => {
  return jwt.sign(user, process.env.ACCESS_TOKEN_SECRET, { expiresIn: "22m" });
};

const createRefreshToken = (user) => {
  return jwt.sign(user, process.env.REFRESH_TOKEN_SECRET, { expiresIn: "14d" });
};

module.exports = userCtrl;

```

3.4.3 Реєстрація та авторизація

Головною вимогою до проекту є наявність бази даних, в якій буде зберігатися адрес електронної пошти та пароль користувача. При створенні нового користувача, його пароль необхідно хешувати і зберігати в базі даних.

```

const newUser = new Users({
  name,
  email,
  password: passwordHash,
});

```

При авторизації користувача виконується така послідовність дій:

- Клієнт відправляє на сервер дані, які містять ідентифікатор та приватний ключ користувача. В даному проекті – це пошта та пароль;
- Сервер шукає користувача в базі даних по адресу електронної скриньки;
- Якщо користувач з обраною електронною скринькою присутній в базі даних, то сервер хешує відправлений йому пароль та порівнює з паролем, збереженим в базі даних;
- При успішній перевірці сервер генерує токен(JWT).

Генерація JWT

```
const createAccessToken = (user) => {  
  return jwt.sign(user, process.env.ACCESS_TOKEN_SECRET, { expiresIn: "22m" });  
};
```

ACCESS_TOKEN_SECRET – секретне слово для формування токена

3.4.4 Middleware перевіряючий авторизацію

```
const jwt = require("jsonwebtoken");  
  
const auth = (req, res, next) => {  
  try {  
    const token = req.header("Authorization");  
    if (!token) return res.status(400).json({ msg: "Invalid Authentication" });  
  
    jwt.verify(token, process.env.ACCESS_TOKEN_SECRET, (err, user) => {  
      if (err) return res.status(400).json({ msg: "Invalid Authentication" });  
  
      req.user = user;  
      next();  
    });  
  } catch (err) {  
    return res.status(500).json({ msg: err.message });  
  }  
};  
  
module.exports = auth;
```

3.4.5 Middleware перевіряючий роль

Роль адміністратора зберігається в базі даних у полі `role` зі значенням 1, а звичайного користувача – 0.

```
const Users = require("../models/userModel");

const authAdmin = async (req, res, next) => {
  try {
    // Get user information by id
    const user = await Users.findOne({
      _id: req.user.id,
    });
    if (user.role === 0)
      return res.status(400).json({ msg: "Admin resources access denied" });

    next();
  } catch (err) {
    return res.status(500).json({ msg: err.message });
  }
};

module.exports = authAdmin;
```

3.4.6 Створення проекту на React

Create React App – зручна середовище для вивчення React та можливість почати створення нового додатку на React. Для створення проекту потрібно виконати команди:

- `npx create-react-app client;`
- `cd client;`
- `npm start.`

3.4.7 Маршрутизація на клієнтській стороні

React має систему маршрутизації, котра дозволяє створювати запити до застосунку з окремими компонентами. Для використання маршрутизації

необхідно використовувати модуль `react-router-dom`, котра включає в себе функціонал для роботи з маршрутизацією при роботі в браузері.

```
import { Switch, Route } from "react-router-dom";
```

Кожен маршрут представляє собою об'єкт `Route`. Завдяки цьому компоненту з'являється можливість рендерити елемент в залежності від вказаного шляху.

Компонент `Route` має три пропси:

- `render`
- `component`
- `children`

На даному етапі розробки використовувався тільки один пропс – `component`, який при успішному зіставленні з `path`, повертає переданий `component`

```
{/*Main Page */}  
<Route path="/" exact component={Home} />
```

3.4.8 Навігаційне меню

Функція `adminRouter`, яка повертає посилання на сторінки «Створити продукт» та «Категорії», які доступні тільки адміністраторам.

```
const adminRouter = () => {  
  return (  
    <>  
      <li className="navbar-link">  
        <Link to="/create_product">Create Product</Link>  
      </li>  
      <li className="navbar-link">  
        <Link to="/category">Categories</Link>  
      </li>  
    </>  
  );  
};
```

Функція `loggedRouter`, яка повертає посилання на сторінки «Історія» та `Logout`, які доступні тільки авторизованим користувачам.

```
const loggedRouter = () => {
  return (
    <>
      <li className="navbar-link">
        <Link to="/history">History</Link>
      </li>
      <li className="navbar-link">
        <Link to="/" onClick={logoutUser}>
          Logout
        </Link>
      </li>
    </>
  );
};
```

3.4.9 Відображення 3D моделі на сторінках

Команди для завантаження необхідних пакетів

- npm i react-three-fiber;
- npm i drei або npm i @react-three/drei;
- npm i three.

Імпортування необхідних компонентів

```
import React, { useRef, Suspense } from "react";

import { OrbitControls } from "three/examples/jsm/controls/OrbitControls";

import { Canvas, useFrame, extend, useThree } from "react-three-fiber";

import { useGLTFLoader } from "drei";

extend({ OrbitControls });
```

Налаштування освітлення та тіні на сцені

```
const Lights = () => {

  return (

    <>
```

```

    { /* Ambient Light illuminates lights for all objects */ }

    <ambientLight intensity={0.3} />

    { /* Direction light */ }

    <directionalLight position={[12, 10, 5]} intensity={1} />

    <directionalLight

        castShadow

        position={[0, 15, 0]}

        intensity={1.5}

        shadow-mapSize-width={1024}

        shadow-mapSize-height={1024}

        shadow-camera-far={50}

        shadow-camera-left={-20}

        shadow-camera-right={20}

        shadow-camera-top={20}

        shadow-camera-bottom={-20}

    />

    { /* Spotlight Large overhead light */ }

    <spotLight intensity={1} position={[1000, 0, 0]} castShadow />

</>

);

};

```

Налаштування камери, розташування моделі та її відображення

```
const HTMLContent = () => {  
  
  const ref = useRef();  
  
  //обертання відносно осі Y  
  
  useFrame(() => (ref.current.rotation.y += 0.01));  
  
  const {  
  
    camera,  
  
    gl: { domElement },  
  
  } = useThree();  
  
  return (  
  
    <Section factor={1.5} offset={1}>  
  
      <group position={[0, 250, 0]}>  
  
        <mesh ref={ref} position={[0, -35, 0]}>  
  
          <Model />  
  
          <orbitControls args={[camera, domElement]} />  
  
        </mesh>  
  
      </group>  
  
    </Section>  
  
  );  
  
};
```

Обернення всіх функцій в одну

```
const Canvass = () => {
```

```
return (  
  <Canvas colorManagement camera={{ position: [0, 0, 120], fov: 70 }}>  
    { /* Lights Component */}  
    <Lights />  
    <Suspense fallback={null}>  
      <HTMLContent />  
    </Suspense>  
  </Canvas>  
);  
};
```

Висновки

В процесі роботи було створено веб-сайту для продажу дизайну інтер'єру з інтерактивним оглядом продукту у 3D. Після процесу ретельного дослідження існуючих аналогічних ресурсів, було вирішено розробити удосконалений ресурс.

Для виконання поставлених задач був використаний стек MERN та мови програмування: JS, HTML, CSS. Також використовувалися такі бібліотеки, як express.js, three.js, react-three-fiber, drei, framer-motion. Вибір цього стеку був зумовлений зручністю у використанні, так як в роботі були наведені багато переваг обраних ресурсів для розробки клієнт-серверного застосування.

Результатом роботи стало створення робочої версії веб-застосунку для продажу, який може бути використаним для не тільки для продажу елементів дизайну інтер'єру, а ще для й будь-яких товарів(техніка, взуття, антикваріат тощо). Моделлю оплати були обрано PayPal, завдяки якому здійснювалась оплата за обраний продукт

Список використаної літератури та електронних ресурсів

1. Документація бібліотеки React.js. [Електронний ресурс]. – Режим доступу: <https://uk.reactjs.org/>
2. Навчальні матеріали для роботи з react-router. [Електронний ресурс]. – Режим доступу: <https://reacttraining.com/react-router/>
3. Документація бібліотеки React Bootstrap. [Електронний ресурс]. – Режим доступу: <https://react-bootstrap.github.io/>
4. Навчальні матеріали для роботи з React Hooks. [Електронний ресурс]. – Режим доступу: <https://ru.reactjs.org/docs/hooks-overview.html>
5. Документація бібліотеки Three.js. [Електронний ресурс]. – Режим доступу: <https://threejs.org/docs/>
6. Документація платформи Node.js. [Електронний ресурс]. – Режим доступу: <https://nodejs.org/uk/docs/>
7. Документація бази даних MongoDB. [Електронний ресурс]. – Режим доступу: <https://docs.mongodb.com>
8. Офіційна сторінка інтернет-магазину. [Електронний ресурс]. – Режим доступу: <https://rozetka.com.ua>
9. Документація бібліотеки React.js. [Електронний ресурс]. – Режим доступу: <https://metanit.com/web/react/>
10. Документація бібліотеки React.js. [Електронний ресурс]. – Режим доступу: <https://www.codecademy.com/learn/react>

Додатки