

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка та Web-дизайн»

Група: 4КГ-05

Дипломний проект

**здобувача освіти денної форми навчання
КГ.05.13.000.ДП**

***ЛИТВИНЕНКА
КОСТЯНТИНА
ОЛЕКСАНДРОВИЧА***

**м. Одеса
2022 р.**

Міністерство освіти і науки України

ВСП «Одеський технічний фаховий коледж ОНТУ»

Відділення _____ Комп'ютерних систем _____ Комісія _____ КТ і ПІ _____
Спеціальність _____ 123 «Комп'ютерна інженерія» _____
Освітня програма _____ «Комп'ютерна графіка і Web-дизайн» _____

ЗАТВЕРДЖУЮ:

Заст. директора з НВР _____

_____ Беркань І.В.

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачу _____ Литвиненко К.О. _____

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка інтернет-магазину з продажу фотографій.

затверджена наказом по коледжу від “ _____ ” _____ 20__ р. № _____

2. Термін здачі студентом закінченого проекту (роботи)

3. Вихідні данні до проекту (роботи) Обґрунтування технології програмування. Розробка структури веб-сайту. Створення веб-сайту засобами HTML. мовами програмування PHP, JavaScript, технологіями HTML, CSS.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1. Вступ. Аналіз сучасних технологій розробки Web-сайтів

2. Конструкторський розділ. Обґрунтування методики створення Web-сайту.

3. Технологічний розділ. Створення Web-сайту із використанням мови програмування PHP, JavaScript, технологіями HTML, CSS. Впровадження веб-сайту.

4 Економічний розрахунок.

5.Охорона праці. Висновки. Перелік використаних джерел

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Мультимедійна презентація

Консультанти по проекту (роботі), із зазначенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Розділи ДП	Кунуп Т.В.		
Економічні розрахунки	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
ЄСКД	Петрашова В.І.		

7. Дата видачі завдання _____ Кунуп Т.В.

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	<i>Аналіз теми ДП та огляд літературних джерел</i>	18.05.2022 р.	
	<i>Аналіз технічного завдання</i>		
2	<i>Розробка технологічного розділу ДП</i>	21.05 2022 р.	
	<i>Постановка задачі при проектуванні Web-сайту</i>		
3	<i>Розробка конструкторського розділу ДП</i>	26.05.2022 р.	
	<i>Розробка структури Web-сайту</i>	01.06.2022 р.	
4	<i>Створення веб-сайту</i>		
5	<i>Виконання розділу «Економічні розрахунки»</i>	04.06.2022 р.	
	<i>Розрахунок економічних показників НДР</i>		
6	<i>Виконання розділу «Охорона праці»</i>	06.06.2022 р.	
7	<i>Виконання пояснювальної записки ДП</i>	07.06 2022 р.	
8	<i>Виконання мультимедійної презентації ДП</i>	10.06 2022 р.	
9	<i>Малий захист</i>	17.06.2022 р.	
10	<i>Захист роботи</i>	21.06.2022р.	

Дипломник _____

(підпис)

Керівник проекту _____

(підпис)

[illegible]

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **123 «Комп'ютерна інженерія»**

Освітня програма: **«Комп'ютерна графіка та Web-дизайн»**

Група: **4КГ-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка інтернет-магазину з продажу фотографій

Проектний матеріал складається з пояснювальної записки на _____ сторінках та графічного (презентаційного) матеріалу на _____ аркушах (слайдах).

Дипломник _____ (Литвиненко К.О.)

Керівник _____ (Кунуп Т.В.)

Консультанти:

з економічної частини _____ (Копайгородська Т.Г.)

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Голова циклової комісії _____ (Скорнякова О.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « ____ » _____ 2022 р. Протокол ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

ЗМІСТ

Вступ.....	7
1. Технологічна частина	
1.1. Створення головної сторінки інтернет-магазину.....	8
1.2. Створення розділу зі слайдером	9
1.3. Створення решти сторінок.....	13
1.4. Створення випадаючого меню.....	15
1.5. Створення кнопки меню	20
1.6. Програмування кнопки отримання консультації.....	24
1.7. Валідація форм	25
1.8. Обробка замовлення.....	32
1.9. Локальний сервер.....	34
2. Економічна частина.....	36
3. Охорона праці.....	41
Висновок.....	45
Використана література.....	46

					КТ.05.13.001.00 ДП ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

Вступ

Актуальність теми дипломної роботи. У важкий для світу час, створення веб-сторінок, стало навіть більш затребуваним, ніж це було раніше. Через всесвітню пандемію більшість бізнесів мала закрити свої офіси у реальному світі і перейти у онлайн режим. До цього моменту багато бізнесів мала офіційні сторінки, але й досі залишалися багато компаній, які не мали свого сайту.

У цій ситуації, компанії, які мали свої веб-сторінки зазнали меншої шкоди, ніж ті, які працювали тільки у режимі офлайн. Тож підприємства, шукаючи вихід із ситуації з закриттям своїх філій на невизначений термін, вирішили перейти у цифровий світ. Компанії почали використовувати програми для дистанційної роботи, а бізнеси, які займалися продажом, майже усі, які досі не мали свого інтернет-магазину зробили усе можливе для їх скорішого створення. Станом на 2020 рік, більшість веб-сторінок, які було створено були саме інтернет-магазинами і ця цифра продовжує зростати.

Інтернет-магазин – це веб-додаток на якому розміщуються товари для їхнього продажу або послуги, які надає компанія, володіюча інтернет-магазином. Зазвичай ці веб-сторінки схожі один на одного, існують різні варіанти дизайну, але функціонал однаковий.

Створення інтернет-магазину – це простий спосіб створити бізнес у сфері торгівлі, так як не потребує багато вкладень. Створюючи інтернет-магазин не потрібно витрачати гроші для оренди приміщення, але орендуючи приміщення це не єдине, що потрібно зробити для відкриття магазину у реальному світі, для цього приміщення потрібно встановити систему пожежної безпеки, систему безпеки самого приміщення і т.д. Тож збиток від того, що бізнес не зможе виконати поставлених цілей значно менше, ніж від офлайн магазинів.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Технологічна частина

1.1 Створення головної сторінки інтернет-магазину

Головна сторінка інтернет магазину повинна містити навігацію, вітальний розділ, розділ з популярними товарами, частину з посиланням на інші сторінки з товарами за певною категорією, контакти, та у нижній частині сайту буде знаходитися посилання на розробника сайту.

Для початку потрібно створити у папці src файл index.html у якому ми створюємо структуру HTML-документа завдяки плагіну Emmet, попередньо встановленому у Visual Studio Code, потрібно написати знак оклику і натиснути клавішу «Tab». Між відкриваючим та закриваючим тегом head будуть знаходитися файли стилів, які ми будемо підключати до цієї сторінки та Fav-іконку, яку користувач зможе побачити на вкладці у браузері перед назвою сторінки.

Між тегом body буде розміщатися увесь контент, який ми хочемо додати на сторінку сайту. Перед закриваючим тегом body повинні розміщуватися файли JavaScript скриптів, які підключаються до сайту.

Для кожного розділу на сайті створюється окремий файл стилів Scss у папці blocks, який підключається до головного файлу стилів style.scss, вже з якого перетворюється у один файл style.min.css, який ми підключаємо до HTML документу.

Спочатку створюється навігація, вона буде розміщуватися зверху сторінки, та коли користувач прокручує сторінку, навігаційне меню повинне залишатися зверху екрану. У самій навігації знаходиться логотип, посилання на інші сторінки, випадаюче меню з елементами, при натисканні на які користувач буде переміщатися між розділами на самій сторінці, та кнопка, при натисканні на яку, буде з'являтися діалогове вікно, у якому користувач зможе звернутися по допомогу. Також, при наведенні на елемент навігації, який відповідає за

					КГ.05.13.001.00 ДП ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

посилання на інші сторінки проєкту, знизу нього буде плавно з'являтися біла полоса, шириною на весь елемент. Це створюється для більш привабливого візуального складника. Також у навігації буде знаходитися випадаюче меню, тому що усі елементи не помістяться без цього. У розділі навігації також буде знаходитися кнопка меню, яке буде використовуватися як навігація при малих розмірах екрану.

У вітальному розділі буде знаходитися назва магазину, кнопка, яка веде до розділу з посиланнями на інші сторінки. Також цей розділ буде розміром на всю висоту екрану користувача.

1.2 Створення розділу зі слайдером

На головній сторінці інтернет-магазину буде розташовуватися слайдер, який буде містити посилання на інші розділи інтернет-магазину.

Для створення слайдеру я використовую технологію Swiper API, завдяки її можливості створення унікальних анімацій для слайдерів у порівнянні з Tiny slider та slick slider, які не дуже підходять до цього проєкту.

Але недолік цієї технології полягає у тому, що вона передбачена для більш новітніх версій платформ.

Для підключення у тезі `<script>` вказуємо джерело, з якого ми підключаємо файл для роботи слайдеру, а після додаємо посилання на файл, у якому ми будемо модифікувати слайдер.

```
<script src="https://unpkg.com/swiper@8/swiper-bundle.min.js"></script>  
<script src="js/slider.js"></script>
```

Рисунок. 1.2.1 – Скріншот підключення файлів роботи для слайдера

У файлі slider.js створюємо змінну для слайдеру, у якій ми вказуємо налаштування для роботи слайдеру.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

```

1  const swiper = new Swiper('.swiper', {
2      slidesPerView: 1,
3      spaceBetween: 10,
4      loop: true,
5      centeredSlides: true,
6      keyboard: {
7          enabled: true,
8      },
9      autoplay: {
10         delay: 2500,
11         disableOnInteraction: false,
12     },
13     breakpoints: {
14         320: {
15             slidesPerView: 1,
16         },
17         576: {
18             slidesPerView: 2,
19             centeredSlides: false,
20         },
21         768: {
22             slidesPerView: 3,
23         },
24         992: {
25             slidesPerView: 3,
26             spaceBetween: 30,
27         },
28     },
29 },
30
31     pagination: {
32         el: '.swiper-pagination',
33         type: "bullets",
34         clickable: true,
35     },
36     navigation: {
37         nextEl: ".swiper-button-next",
38         prevEl: ".swiper-button-prev",
39     },
40
41 });

```

Рисунок 1.2.2 – Скріншот налаштування слайдера

У параметрі `slidesPerView` ми вказуємо кількість слайдів, які будуть відображатися на сайті за замовченням. Вказано один, тому що це потрібно для подальшої адаптації під пристрої з меншою шириною екрану. Цей параметр ми змінили при досягненні ширини екрану більше ніж 992 пікселі, та при ширині менше 576 та 320 пікселі. Властивість `spaceBetween` вказує відстань між слайдами у пікселях. Завдяки властивостям `loop` та `keyboard`, ми зможемо перемикатися між слайдами завдяки клавіатурі, та наш слайдер буде зациклений, тобто його можна буде крутити і після останнього слайду знов буде йти перший.

					КГ.05.13.001.00 ДП ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

Якщо користувач не буде переключатися між слайдами, то слайдер буде самостійно кожні 2,5 секунди перемикає слайд на наступний, завдяки властивості autoplay. Властивість centeredSlides буде відображати поточний слайд по центру, але при меншій ширині вікна браузера це налаштування буде вимикатися, тому що воно впливає на адаптивність слайдера.

Сам слайдер на сторінці буде містити картинку та текст до неї, що буде показувати, до якої категорії товарів ми переходимо. При натисканні на картинку користувач буде переходити до сторінки з обраною категорією.

```

51 <section class="trend">
52   <h2 class="title">Check most viewed items</h2>
53   <div class="swiper">
54     <div class="swiper-wrapper">
55       <div class="swiper-slide"></div>
57       <div class="swiper-slide"></div>
58       <div class="swiper-slide"></div>
59     </div>
60     <div class="swiper-pagination"></div>
61     <div class="swiper-button-prev"></div>
62     <div class="swiper-button-next"></div>
63   </div>
64 </section>
65

```

Рисунок 1.2.3 – Фрагмент HTML коду слайдеру.

Для того, щоб додати привабливий зовнішній вигляд для нашого слайдеру, ми створюємо scss файл, у якому ми будемо додавати стилі до слайдеру, файл стилів буде мати назву, яка збігається з назвою секції, для того, щоб було зрозуміло у якому файлі і що саме ми модифікуємо. Після створення файлу, потрібно підключити scss файл секції до основного scss файлу.

```

1  .trend {
2      position: relative;
3      min-height: 800px;
4      color: #fff;
5      margin: 0 40px;
6      .swiper {
7          width: auto;
8          height: 450px;
9          margin: 150px 20px;
10         &-wrapper {
11             display: flex;
12             align-self: start;
13         }
14         &-slide {
15             border-radius: 10px;
16             overflow: hidden;
17         }
18         &-button-next,
19         &-button-prev {
20             position: absolute;
21             color: $bright-color;
22         }
23         &-pagination {
24             &-bullet {
25                 width: 20px;
26                 background: $bright-color;
27                 opacity: 0.4;
28                 border-radius: 3px;
29                 &-active {
30                     opacity: 1;
31                 }
32             }
33         }
34     }
35 }

```

Рисунок 1.2.4 – Скріншот властивостей для слайдера.

Наступним розділом буде розділ з посиланнями на інші сторінки проєкту. У цьому розділі будуть знаходитися картки, при наведенні на які вони будуть плавно збільшуватися, це дасть змогу користувачу зрозуміти, що на ці елементи можна натискати. Коли користувач буде натискати на елемент його буде перенаправляти на відповідну сторінку.

Розділ з контактами містить текстову складову та картинку. У текстовій складовій зазвичай вказується адреса підприємства, посилання на різні соціальні

мережі і так далі. У моєму проєкті у контактах знаходиться посилання на сторінку в соціальній мережі Instagram, посилання на веб-сервіс для розробників GitHub та месенджер Telegram. При наведенні на іконки з посиланням, вони повинні підійматися, це також зроблено для створення кращої візуальної складової сайту.

Останнім розділом на головній сторінці буде розділ, який прийнято називати «підвалом» сайту. У цьому розділі зазвичай розміщують умови використання ресурсу, поширення інформації з ресурсу та інформацію про розробнику. У моєму разі цей розділ буде використовуватися через останню причину. При натисканні на іконку Github користувач буде переходити на мій профіль, а при наведенні на іконку вона буде повертатися на 360 градусів, також при натисканні на текст користувач зможе перейти на Github ресурс.

1.3 Створення решти сторінок

Після створення головної сторінки сайту, з якої користувач має змогу дізнатися про категорії на сайті, популярні товари, а також відвідати сторінки у соціальних мережах, створюємо сторінки за відповідними категоріями.

Усі сторінки з товарами будуть зроблені дотримуючись однакового дизайну, для того, щоб користувач мав змогу після відвідування однієї сторінки розуміти як працювати з іншими.

Перша сторінка з товарами буде мати назву Nature і на ній буде розміщуватися фотографії природи, які користувач буде мати змогу придбати.

Коли користувач переходить до певної категорії, у навігації буде з'являтися пряма лінія, яка буде залишатися під назвою категорії, яку користувач переглядає, ця лінія буде такою самою, як і при наведенні на певний елемент навігації. У навігації також буде випадаюче меню, яке буде містити посилання на головну сторінку проєкту, на розділ з контактами поточної сторінки та на розділ з товарами, для більш простої навігації з будь-якого розділу сторінки, так як навігація завжди залишається на екрані користувача. Також, при малому

					КГ.05.13.001.00 ДП ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

розмірі екрану користувача, навігація стилізується та програмується, щоб при натисканні на елемент кнопки меню з'являлася навігація для пристроїв з малим екраном, на ній також під назвою кожної сторінки буде лінія, як і на пристроях з великим розміром екрану.

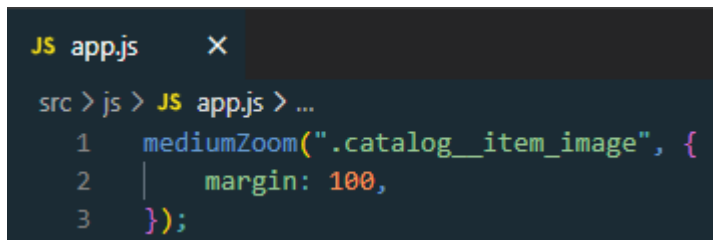
У вітальному розділі знаходиться назва магазину, під якою знаходиться назва категорії, у якій зараз перебуває користувач та кнопка, яка, також як і кнопка у випадаючому меню, направляє користувача до розділу з товарами. У цьому розділі також буде знаходитися фотографія, яка відповідає категорії сторінки, на яку користувач перейшов, ця фотографія буде розміщена як задній фон розділу.

Розділ з товарами містить назву категорії товару, до якого вони належать. На самих картках з товаром знаходиться ціна на товар, його назва, кнопка, при натисканні на яку, відкривається діалогове вікно, у якому знаходиться назва товару, форма для отримання контактів користувача, таких як ім'я, номер телефону та адресу електронної пошти та кнопка, яка відправляє форму. У діалоговому вікні також повинен знаходитися хрестик, при натисканні на який, діалогове вікно буде закриватися. У картці також буде знаходитися фотографія, форма картки буде нагадувати рамку з фотографією, так як це інтернет-магазин з продажу фотографій.

При наведенні на елемент з фотографією, сам елемент збільшиться та при наведенні на картинку з'явиться лупа, при натисканні лівої кнопки миші, користувач зможе більш детально роздивитися фотографію це можна створити завдяки бібліотеці JavaScript, яка називається Medium Zoom [1]. Ця бібліотека проста у підключенні до сайту та у використанні. Підключати цю бібліотеку до проєкту я буду завдяки додаванні її до файлу проєкту, файл бібліотеки буде мати назву medium.js, а його налаштування будуть знаходитися у файлі app.js. У налаштуванні вказується елемент, з яким виконуються дії та які саме дії будуть виконуватися, в моєму разі це елемент картинки товару, та властивість, яка відповідає за простір за межами зображення. Ця бібліотека також надає змогу

					КГ.05.13.001.00 ДП ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

повернути картинку у звичайний розмір натиснувши по ній ще один раз лівою кнопкою миші або просто прокрутити сторінку вниз.



```
JS app.js X
src > js > JS app.js > ...
1   mediumZoom(".catalog__item_image", {
2   |     margin: 100,
3   });
```

Рисунок 1.3.1 – Скріншот Налаштування Medium Zoom

Наступним буде після розділу з товарами буде йти розділ з контактами. Цей розділ не змінюється на всіх сторінках, так як потрібно, щоб відвідувач мав змогу зв'язатися з інтернет-магазином або відвідати соціальні мережі.

Після розділу з контактами користувач побачить процес виконання замовлення, а саме його замовлення, підтвердження та відправлення, а також картинку до цих дій.

Останнім йде наш «підвал» сайту, який однаковий на усіх сторінках та містить посилання на GitHub сторінку.

1.4 Створення випадаючого меню

Випадаючі меню потрібні, якщо потрібно додати інформацію, яка не завжди потрібна, або потрібна тільки якщо є додаткова інформація. Випадаюче меню з'являється коли користувач натискає на елемент та дає змогу обрати одну опцію з набору.

Для створення випадаючого меню також використовується мова програмування JavaScript. При створенні якогось функціоналу за допомогою будь-якої мови програмування, спочатку потрібно зрозуміти завдання, створити план та написати псевдокод.

«Розв'язання проблеми — це написання оригінальної програми, яка виконує певний набір завдань і відповідає всім встановленим обмеженням». - Думай як програміст.

					КГ.05.13.001.00 ДП ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

Написання псевдокоду перед створенням справжнього коду – це корисна звичка, яка дає змогу створити логіку програми, при цьому навіть не потрібно знати ніяку мову програмування, тому що псевдокод це написання коду зрозумілою людині мовою. Псевдокод також допомагає розділити написання складної програми на частини, які легше створити.

У такому разі, для створення випадаючого меню потрібно: створити елемент, при натисканні на який, буде з'являтися меню, у якому можна буде обрати потрібну опцію, а при повторному натисканні меню буде пропадати.

Меню буде знаходитися у навігації сайту, тож воно, як і інші елементи навігації буде елементом списку, та мати свій клас, всередині елемента буде знаходитися ще один список з опціями, який за замовчуванням буде мати властивість `display` зі значенням `none` для того, щоб воно з'являлося тільки при натисканні на елемент випадаючого меню. Ці та інші властивості, які пов'язані з елементом навігації повинні знаходитися у відповідному `scss` файлі.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

```

<header class="header">
  <div class="container">
    <nav class="nav">
      <div class="nav__logo">
        <a href=""></a>
      </div>
      <div class="nav__links">
        <ul class="ul">
          <li class="nav__links_item"><a href="../../pages/wild.html">Wild</a></li>
          <li class="nav__links_item"><a href="../../pages/pets.html">Pets</a></li>
          <li class="nav__links_item"><a href="../../pages/landscape.html">Landscape</a></li>
          <li class="nav__dropdown">
            <p>More </p>
            <ul id="dropDown">
              <li><a href="../../index.html">Main</a></li>
              <li><a href="#contacts">Contacts</a></li>
              <li><a href="#">Items</a></li>
            </ul>
          </li>
        </ul>
      </div>
      <button class="btn" data-modal="consultation">Help Me!</button>
      <div class="burger" id="burger">
        <span></span><span></span><span></span>
      </div>
    </nav>
  </div>
</header>

```

Рисунок 1.4.1 – Фрагмент HTML коду випадаючого меню

Для стилізації випадаючого меню, ми для кожного потрібного елементу додали клас. Щоб додати стилі ми відкриваємо файл стилів scss, який відноситься до відповідної секції, у нашому разі header, та додаємо потрібні властивості, такі як ефект курсору, розташування, колір, а також анімацію для стрілки, щоб вона мала змогу повертатися, та мала анімацію повороту. Також кожен елемент списку повинен збільшуватися при наведенні курсору для того, щоб користувач знав, яку саме опцію він обирає. Треба пам'ятати про те, що само меню та його елементи будуть з'являтися лише, коли користувач натискає на меню, тож потрібно встановити властивість display у позицію none, та коли випадаюче меню має клас активності властивість змінюється на значення block, щоб користувач мав змогу його побачити, а стрілка у меню повернулася у іншу сторону.

```

_header.scss X
src > sass > blocks > _header.scss > .header > .nav > &_links > .ul
78
79 .nav__dropdown {
80     cursor: pointer;
81     &::after {
82         display: none;
83     }
84     ul {
85         display: none;
86         position: absolute;
87         list-style-type: none;
88         width: 100px;
89         text-align: center;
90         margin: 0;
91         padding: 0;
92         background-color: $bg-color;
93         border: 1px solid $bright-color;
94         border-radius: 10px;
95         margin-top: 5px;
96         li {
97             padding: 5px;
98             &:hover {
99                 transform: scale(1.1);
100                 &::after {
101                     display: none;
102                 }
103             }
104             a {
105                 display: block;
106                 width: 100%;
107             }
108         }
109     }
110     img {
111         width: 15px;
112         height: 15px;
113         margin-left: 5px;
114         transition: transform .5s;
115     }
116     &_active {
117         ul {
118             display: block;
119         }
120         img {
121             transform: rotate(180deg);
122         }
123     }
124 }

```

Рисунок 1.4.2 –Скріншот Scss властивостей для випадаючого меню

Реалізація функціоналу випадаючого меню при допомозі мови програмування JavaScript. Створюємо змінну у якій буде зберігатися елемент зі списку навігації, який відповідає за появу випадаючого меню та змінну у якій буде зберігатися випадаюче меню. Ці змінні ми ініціалізуємо завдяки методам

getElementById та getElementsByClassName. Метод getElementsByClassName повертає колекцію, яка подібна на масив, тож нам потрібно вказати який саме елемент потрібен, у нашому випадку там тільки один елемент, який має індекс 0, тож ми звертаємося до цього елемента і створюємо подію натискання лівої кнопки миші, після чого до наших елементів, кнопки випадаючого меню, та списку додаються класи активності, які ми попередньо стилізували, при повторному натисканні на кнопку випадаючого списку, клас активності прибирається та наш елемент повертається у первісний стан.

```
const dropdown = document.getElementById("dropdown");
const navDropdown = document.getElementsByClassName("nav__dropdown");
navDropdown[0].addEventListener("click", () => {
  dropdown.classList.toggle("nav__dropdown_active");
  navDropdown[0].classList.toggle("nav__dropdown_active");
});
```

Рисунок 1.4.3 – Фрагмент JavaScript коду реалізації випадаючого меню

Коли користувач натискає на елемент випадаючого меню, випадаюче меню активується.

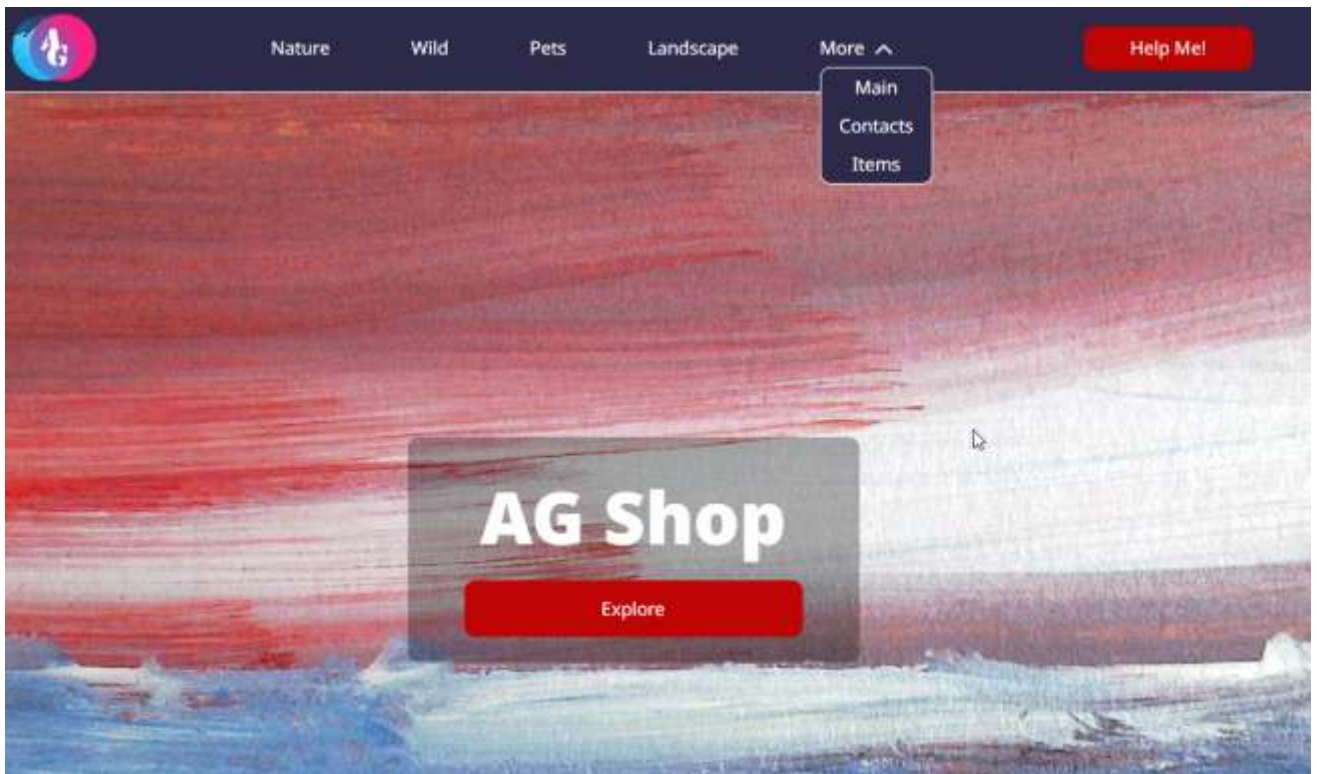


Рисунок 1.4.4 – Скріншот роботи випадаючого меню

					КТ.05.13.001.00 ДП ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

1.5 Створення кнопки меню

У разі, коли навігація сайту стає більше ніж ширина екрану, а це завжди трапляється, коли на веб-сторінку заходять з мобільного пристрою, планшету, та навіть деяких ноутбуків, вміст сторінки спотворюється, що створює додаткові проблеми при читанні інформації та натискання на посилання, використовують спеціальний елемент, при натисканні на яку з'являється меню [2].

Кнопка меню складається з трьох рисочок, які нагадують відому їжу з ресторанів швидкого харчування. Таке меню використовується у випадках, коли весь розділ навігації не вміщується у розміри екрану пристрою. Функціонал такого меню залежить від бажань клієнта, але найчастіше функціонал такої кнопки такий: при зменшенні розміру екрана користувача до розмірів, при яких навігація більше вміщується у розміри екрану, меню навігації ховається, та на її місці з'являється елемент кнопки меню, натискаючи на який, з лівої сторони екрану у користувача з'являється меню навігації у вертикальній та зменшеній формі, що дає змогу вільно її читати та натискати на посилання.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

```

@media(max-width: 768px) {
  .header {
    .nav__links {
      display: none;
      .ul {
        flex-direction: column;
        li {
          margin-top: 15px;
        }
      }
      &_active {
        position: fixed;
        top: 0;
        left: 0;
        display: flex;
        flex-direction: column;
        align-items: stretch;
        justify-content: center;
        width: 30%;
        height: 100vh;
        background-color: rgba($color: #000000, $alpha: .8);
      }
    }
    .nav__dropdown {
      ul {
        background-color: rgba($color: #000000, $alpha: .8);
      }
    }
    .btn {
      display: none;
    }
    .burger {
      display: flex;
    }
  }
}

```

Рисунок 1.5.1 – Стилізація навігації при малому розмірі екрану

Створення функціоналу меню виконується завдяки можливостям мови програмування JavaScript. Писати код для реалізації роботи кнопки меню у тому ж JavaScript файлі, що й код для випадаючого меню, так як ці елементи будуть на кожній сторінці і не буде виникати помилки при пошуку потрібного елемента.

```

const burger = document.getElementById("burger");
const navLinks = document.getElementsByClassName("nav__links");
burger.addEventListener("click", () => {
  burger.classList.toggle("burger_active");
  navLinks[0].classList.toggle("nav__links_active");
});

```

Рисунок 1.5.2 – Створення функціоналу для кнопки меню

					КГ.05.13.001.00 ДП ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

В першу чергу потрібно ініціалізувати змінні у яких буде знаходитися елемент кнопки меню та елементи навігації. Після цього, створюємо метод `addEventListener`, функція у якому буде виконуватися кожен раз, коли користувач натискає на елемент кнопки меню. У цьому методі зазвичай використовується стрілочна функція, так як у цьому випадку не потрібно вказувати ім'я функції, воно і не потрібно при створенні цього методу, так як ми прописуємо логіку у тілі методу.

Після натискання на елемент меню, йому буде надаватися клас активності `burger__active` для того, щоб змінити його вигляд, а також при натисканні на елемент кнопки меню елементам навігації також потрібно додати клас активності для їх стилізації на малих екранах. Для того, щоб отримати елементи навігації ми вказуємо у квадратних дужках цифру 0, так як метод `getElementsByClassName` повертає HTML колекцію [3] з усіма елементами з зазначеним у дужках класом, у нашому разі тільки один елемент має клас `nav__links`, тож він буде мати індекс 0. Для елемента навігації встановлюється клас активності `nav__links_active`, завдяки чому навігація буде мати стилі, які вказані на рисунку 1.4.1. В цьому разі, зовнішній вигляд кнопки змінюється завдяки стилям и перетворюється у хрестик, при натисканні на який навігаційне меню закривається.



Рисунок 1.5.3 – Скріншот головної сторінки при активованій кнопці меню

					КТ.05.13.001.00 ДП ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

1.6 Програмування кнопки отримання консультації.

Якщо у користувача виникнуть питання, щодо замовлення товару, його отримання, оплати і тому подібне, користувачу потрібно буде отримати консультацію, для цього буде створена кнопка, при натисканні на яку, з'явиться вікно, де користувач зможе залишити свої контактні данні, натиснути кнопку відправлення, після чого його данні перейдуть на електронну адресу адміністратора сайту.

Кнопка отримання консультації буде знаходитися у навігації сайту та розділу з контактами, а у разі малого розміру екрану користувача тільки у розділі з контактами.

```
<section class="contacts" id="contacts">
  <div class="contacts__text">
    <h2 class="title">Our contacts</h2>
    <div class="descr">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Aperiam odio quas perferendis nobis impedit, asperiores quasi mollitia nisi pariatur dicta ducimus praesentium ab similique reiciendis eligendi? Nesciunt beatae temporibus maxime? Suscipit eaque officia fuga, facere dolorem eum, vero asperiores molestiae possimus rem facilis eos doloribus ad ea quibusdam incidunt molestias debitis dicta praesentium earum blanditiis eius? Ad dolore atque suscipit. Repudiandae dolorum accusantium corrupti quasi at molestiae placeat, quod ut earum! Qui recusandae quasi distinctio expedita tempore ea cupiditate aspernatur aut doloribus enim sequi accusamus culpa, suscipit voluptates officia rerum? Sint blanditiis dignissimos aspernatur! Ut consectetur nesciunt distinctio dolorum animi beatae repudiandae nemo ea neque exercitationem, voluptatum asperiores similique magnam hic? Rerum, consectetur ducimus. Sunt, unde. Rerum repellat harum dolorem! Cupiditate, pariatur quo? Pariatur fugit expedita recusandae veritatis, suscipit illo impedit doloribus ratione doloremque laudantium commodi facilis harum earum itaque tempora sed delectus enim rem non esse ad ea quibusdam.</div>
    <div class="contacts__links">
      <a class="contacts__links_item" href="https://www.instagram.com/" target="_blank"></a>
      <a class="contacts__links_item" href="https://github.com/TheKostyak" target="_blank"></a>
      <a class="contacts__links_item" href="https://web.telegram.org/" target="_blank"></a>
    </div>
    <button class="btn contacts_btn" data-modal="consultation">Help Me!</button>
  </div>
  <div class="contacts__image">
    
  </div>
</section>
```

Рисунок 2.6.1 - Фрагмент HTML коду розділу з контактами

					КТ.05.13.001.00 ДП ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

Програмування появи діалогового вікна при натисканні на кнопку буде відбуватися у файлі `app.js`, так як цей файл буде підключений до кожної сторінки інтернет-магазину. Для реалізації появи діалогових вікон на сайті буде використовуватися jQuery API [4].

Спочатку потрібно використовувати функцію `$(document).ready()`, для того, щоб безпечно маніпулювати DOM-деревом, тіло цієї функції буде виконуватися тільки після того, як буде створено об'єктна модель документа. В тілі функції пишемо з яким елементом ми будемо взаємодіяти та які саме взаємодії будуть відбуватися. Конструкція `$(“ідентифікатор”)` у jQuery потрібна для вказання з яким елементом відбувається взаємодія. В нашому разі при натисканні на елемент (метод `.on(“click”, function(){})`) буде виконуватися функція, яка відображає діалогове вікно. Для цього використовується jQuery метод `.fadeIn()`.

У разі якщо користувач хоче закрити діалогові вікна, йому потрібно натиснути хрестик у правому верхньому кутку діалогового вікна. Це робиться так само як відображення діалогового вікна але замість методу `.fadeIn()` використовується метод `.fadeOut()`.

```
$(document).ready(function() {
    $('[data-modal=consultation]').on("click", function() {
        $(".overlay, #consultation").fadeIn();
    });
    $(".modal__close").on("click", function() {
        $(".overlay, #consultation, #thanks, #order").fadeOut();
    });
});
```

Рисунок 1.6.2 – Скріншот JavaScript коду реалізації діалогового вікна

1.7 Валідація форм.

Для отримання контактних даних користувача потрібно, щоб користувач заповнив всі поля для введення інформації, а також заповнив їх коректно. У мові гіпертекстової розмітки для поля введення є атрибут `required`, який запобігає відправленню форми, якщо поле введення порожнє, але вказавши цей атрибут

					КГ.05.13.001.00 ДП ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

користувач зможе відправити форму, якщо заповнить поле будь-якою інформацією, будь то правильно введена інформація, або тільки набір символів. Для валідації форм існує багато різних API, такі як imask.js [5], jquery validate [6] та інші. Ці два плагіна не підходять під вимоги проєкту, так як imask.js резервує цифру 0, як пустий символ для введення номеру, що не підходить під код країни, який містить 0. У разі використання jquery validate зарезервована цифра 9, що не заважає у цьому випадку, але коли користувач натискає на поле введення курсор з'являється у тому місті, де він натиснув, замість появи на початку вводу номеру, а також jquery validate давно не оновлювався, тож можна не очікувати нових функцій або виправлення старих. Але jquery validate добре підходить для валідації адреси електронної пошти та виведення повідомлень виникнення помилок.

Для створення маски введення номеру, щоб налаштувати інші зарезервовані цифри потрібно змінювати налаштування API. Інший спосіб це вибір серед багатьох інших API, наприклад, cleave.js [7].

В першу чергу потрібно додати основний файл cleave.js, у якому написані функції для роботи API, після цього, якщо потрібно використовувати маску для номерів телефонів підключається додатковий файл з функціями для валідації телефону, у моєму разі потрібен префікс «+380», для цього створюємо файл JavaScript у якому буде знаходитися валідація форми, вказуємо селектори елементів, які будуть проходити валідацію, а саме поле з іменем, номером телефону та адресою електронної пошти. Для валідації номеру телефону використовуємо Cleave.js, де вказуємо елемент, який буде проходити валідацію, вказуємо, що це поле з номером телефону, країну та префікс, з якого починається номер.

Наступним кроком буде валідація поля з іменем та адресою електронної пошти, для цього вказуємо селектор форми, а також що це поле обов'язкове для заповнення, для поля з електронною поштою ще потрібно вказати, що у цей елемент буде вводиться адреса електронної пошти і потрібно дотримуватися

					КГ.05.13.001.00 ДП ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

стандарту введення електронної пошти. Для поля с телефоном додатково вказуємо мінімальну кількість символів для вводу, враховуючи пробіли.

```
$(document).ready(function() {
    $('input[name=phone]').toArray().forEach(function(field) {
        var cleave = new Cleave(field, {
            phone: true,
            phoneRegionCode: 'UA',
            prefix: '+380',
        });
    });
    $("#consultation .feed-form").validate({
        rules: {
            name: "required",
            email: {
                required: true,
                email: true
            },
            phone: {
                required: true,
                minlength: 16
            }
        }
    });
});
```

Рисунок 1.7.1 – Фрагмент коду валідації форми

Якщо користувач спробує відправити порожню або наполовину заповнену форму, йому висвітиться помилка, яка попросить заповнити форму.

Leave your contacts
and we will phone you

Your name

This field is required.

+380

Please enter at least 16 characters.

E-mail

This field is required.

Help me!

Рисунок 1.7.2 – Скріншот спроби відправлення порожньої форми

Після заповнення форми, користувач повинен відправити дані на сервер, для цього потрібно використовувати технологію AJAX, яка дає змогу передавати дані на сервер без перезавантаження сторінки, так як за замовчуванням при заповненні форма відправляється та перезавантажує сторінку.

Щоб змінити стандартну поведінку форми ми обираємо потрібну форму, використовуючи її селектор та використовуємо метод `.submit`, у тілі функції записуємо поведінку, як параметр у функцію будемо передавати «e», що від слова Event, тобто подія, яка відбувається. Наступним кроком для параметру, який передається у функції, а саме подія відправлення форми, змінюємо стандартну поведінку, використовуючи метод `.preventDefault()`, також потрібно перевірити, чи пройшла форма валідацію, якщо ні, то форма не відправиться. Це зроблено для того, щоб не відправляти порожню форму. Далі, використовуємо AJAX для відправлення даних. Потрібно вказати тип даних POST, якщо дані відправляються на сервер та GET, якщо отримуємо дані з серверу, в моєму разі використовуємо метод POST, також потрібно вказати який файл буде обробляти запит та його розташування. Для даних, що надсилаються використовуємо

					КГ.05.13.001.00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

ключове слово `this`, яке вказує на дані, з якими ми працюємо та для них використовуємо метод `.serialize()`, який готує дані для відправки на сервер. Після того, як форма відправилась потрібно щоб з'явилося діалогове вікно, яке сповіщає користувача про успішне відправлення даних. Для реалізації цієї функції використовується метод `.done()`, в дужках пишеться функція, яка буде виконуватися після відправлення даних. У тілі цієї функції вказуємо форму, з якою ми працюємо, та використовуючи метод `find`, в якому вказуємо всі елементи поля введення, для них вказуємо метод `val` з порожнім значенням, для того, щоб очистити поля введення. Наступним буде використання методів `.fadeOut` та `.fadeIn` для приховування одного діалогового вікна та появи іншого, останньою дією потрібно скинути усі форми.

```
$('#email-help').submit(function(e) {
    e.preventDefault();

    if (!$('this').valid()) {
        return;
    }

    $.ajax({
        type: "POST",
        url: "mail/email-help.php",
        data: $('this').serialize()
    }).done(function() {
        $('this').find("input").val("");
        $('#consultation').fadeOut();
        $('#overlay, #thanks').fadeIn();
        $('form').trigger('reset');
    });
    return false;
});
```

Рисунок 1.7.3 – Фрагмент JavaScript коду обробки відправлення форми

Потрібно налаштувати відправлення повідомлення на електронну адресу сайту, для цього переходимо у файл, до якого ми вказали шлях у JavaScript файлі. Для відправлення повідомлення використовується мова програмування PHP та бібліотека `PHPMailer` [8], яка дозволяє відправляти зашифровані повідомлення з підтримкою HTML структури та дозволяє прикріплювати

					КГ.05.13.001.00 ДП ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

вкладення до повідомлення, що не дозволяє зробити вбудована функція у PHP - mail().

Так як ми передаємо дані з форми при допомозі JavaScript використовуючи метод POST у файлі PHP потрібно прийняти ці дані та записати їх до змінних.

```
$name = $_POST['name'];  
$phone = $_POST['phone'];  
$email = $_POST['email'];  
$curr_date = date("d-m-y, H:i:s");  
require_once('phpmailer/PHPMailerAutoload.php');  
$mail = new PHPMailer;  
$mail->CharSet = 'utf-8';  
$webPage = 'AGshop';
```

Рисунок 1.7.4 – Фрагмент PHP коду зі створенням змінних

Як раз для того, щоб записати правильні дані у змінні у тега input потрібно вказувати атрибут name, тому що у PHP значення беремо з поля, у якому вказано атрибут name з відповідним значенням. Потрібно знати час, коли клієнт залишив свої дані, для цього використовуємо функцію PHP date, яка повертає отформатовану згідно з вказаними параметрами дату та час. Також потрібно підключити файли бібліотеки PHPMailer, після чого ми встановлюємо кодування символів стандарту UTF-8. Створюємо змінну, у якій буде зберігатися адреса сайту, на якому користувач залишив заявку, це потрібно буде під час відправлення повідомлення на електронну пошту сайту, так як компанії можуть володіти декількома сайтами одночасно.

Наступним потрібно налаштувати використання SMTP протоколу для відправлення повідомлення з електронної пошти. Для цього потрібно вказати поштовий клієнт, в нашому разі це клієнт електронної пошти gmail. Також потрібно вказати, що використовується ідентифікація, а саме адреса електронної пошти та пароль від застосунку. Для захисту повідомлення використовується протокол SSL який використовує асиметричну криптографію для аутентифікації ключів обміну. Також потрібно вказати порт, який використовує поштовий сервіс. Такі самі налаштування потрібно вказати на локальному сервері для перевірки відправлення повідомлення.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Потрібно також вказати адресанта та адресата, тут потрібно виходити з того, чи потрібно відправляти повідомлення користувачу про його запит, чи тільки повідомлення буде надходити до адміністрації сайту. Дотримуючись другого варіанту у проєкті заявки користувачів будуть надходити тільки адміністрації сайту, для цього вказуємо поштову адресу адміністрації як адресата. Для створення структури повідомлення використовуємо мову гіпертекстової розмітки, щоб мати змогу використовувати теги HTML у повідомленні.

При створенні повідомлення вказуємо тему повідомлення, це буде назва сайту, де користувач залишив заяву. У тілі повідомлення вказуємо адресу сайту, дату залишення даних, ім'я користувача, телефонний номер та електронну адресу.

Останнім, що потрібно зробити при відправленні повідомленні, це додати конструкцію обробки помилок, якщо вони з'являться.

```

13  $mail->isSMTP();
14  $mail->Host = 'smtp.gmail.com';
15  $mail->SMTPAuth = true;
16  $mail->Username = '';
17  $mail->Password = '';
18  $mail->SMTPSecure = 'ssl';
19  $mail->Port = 465;
20
21  $mail->setFrom('', 'AG Shop');
22  $mail->addAddress($mail->Username);
23  $mail->isHTML(true);
24
25  $mail->Subject = 'AG Shop';
26  $mail->Body = '
27  The request has been left on: ' . $webPage . '<br> Date: ' . $curr_date . '<br>
28  Name: ' . $name . ' <br>
29  Phone number: ' . $phone . ' <br>
30  E-mail: ' . $email . ' ';
31
32  if(!$mail->send()) {
33      return false;
34  } else {
35      return true;
36  }

```

Рисунок 1.7.5 – Фрагмент відправлення повідомлення

					КГ.05.13.001.00 ДП ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

Після правильного заповнення та відправлення форми, на пошту адміністрації сайту приходить повідомлення з даними користувача.

AG Shop <kostaa900@gmail.com>
кому: мне ▼
The request has been left on: www.AGshop.com
Date: 11-06-22, 12:59:51
Name: test
Phone number: +380 312 312 312
E-mail: test@test

Рисунок 1.7.6 – Скріншот повідомлення на електронній пошті

1.8 Обробка замовлення.

Як і при замовленні консультації потрібно зробити діалогове вікно при замовленні товару.

У елементі з товаром буде знаходитися кнопка, при натисканні на яку повинно з'являтися діалогове вікно з назвою товару, та полями для введення даних користувача. Поля введення будуть такі ж самі як і при заповненні форми для отримання консультації, але потрібно реалізувати динамічне змінення назви товару при виборі різних товарів, тобто у діалоговому вікні буде з'являтися назва товару, яка вказана на самому товарі. Для реалізації такої функції використовується мова програмування JavaScript.

У файлі app.js, де ми до цього реалізували натискання на кнопки консультації та кнопки закриття. Потрібно вказати селектор кнопки, при натисканні на яку будуть виконуватися функції появи діалогового вікна замовлення товару, а також створити функцію, яка при натисканні на будь-яку кнопку з товаром, буде брати назву обраного товару і додавати її до елемента з назвою у діалоговому вікні, а також до прихованого елемента введення, щоб потім додати до атрибута value і передати на сервер, щоб адміністратор сайту бачив, який товар обрав користувач.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

```

$(".btn_buy").on("click", function() {
    $(".overlay, #order").fadeIn();
});
$(".btn_buy").each(function(i) {
    $(this).on("click", function() {
        $("#order .modal__subtitle").text($(".catalog__item_title").eq(i).text());
        $("#order .input__subtitle").text($(".catalog__item_title").eq(i).text());
    });
});

```

Рисунок 1.8.1 – Фрагмент JavaScript коду натискання на кнопку замовлення товару

Створюємо JavaScript файл, у якому реалізуємо валідацію форми, цей файл буде мати назву email-order.js, щоб відрізнити його від файлу з консультацією.

Валідація форм проходить так само, як і у файлі з формою для консультації, але потрібно ще додати фрагмент коду у якому буде створена змінна, яка відповідає за елемент з назвою товару та ініціалізувати цю змінну елементом з селектором класу input__subtitle, так як метод `getElementsByClassName` повертає колекцію з елементів, потрібно звертатися до першого елементу з індексом 0. Для того, щоб можна було отримати назву товару на сервері, потрібно додати назву елемента до атрибуту `value` у полі введення, для цього ми беремо значення з текстового поля тегу `input` та поміщаємо значення у атрибуту `value`.

```

let inputValue = document.getElementsByClassName("input__subtitle");
inputValue[0].value = inputValue[0].textContent;

```

Рисунок 1.8.2 – Фрагмент коду додавання значення до атрибуту `value`

Також потрібно змінити шлях до файлу PHP який відправляє повідомлення про замовлення товару на файл з назвою `order.php`

У файлі `order.php` так само створюємо структуру повідомлення, але ще потрібно додати змінну, у якій буде зберігатися назва товару, який користувач обрав.

```
Order has been placed on: ' . $webPage . '<br> Date: ' . $curr_date . '<br>
Name: ' . $name . '<br>
Phone number: ' . $phone . '<br>
Order: ' . $order . '<br>
E-mail: ' . $email . '';
```

Рисунок 1.8.3 – Скріншот структури заказу

1.9 Локальний сервер.

Після створення сайту потрібно протестувати його роботу, але так як при створенні сайту ми використовували планувальник задач Gulp та плагін BrowserSync до нього, ми не мали змогу відправляти повідомлення на електронну пошту або виконувати інші можливості серверної частини, потрібно встановити додаткове програмне забезпечення Open Server, яке дозволяє створити локальний сервер.

Після встановлення програми потрібно налаштувати її так, щоб вона мала доступ до електронної пошти, яка буде використовуватися для повідомлень. Для цього потрібно у налаштуваннях програми у вкладці пошта вказати такі ж дані, що й у файлах PHP, які відповідають за відправлення повідомлень, а саме smtp сервер, порт, адресу електронної пошти, яка буде використовуватись та пароль для застосунку.

Далі потрібно запустити сервер та перевірити як відправляються повідомлення.

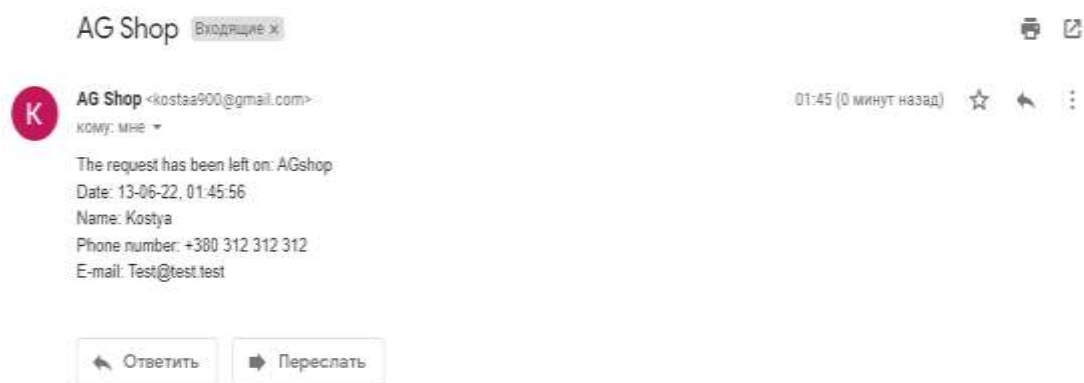


Рисунок 1.9.1 – Скріншот повідомлення заяви про отримання консультації

					КТ.05.13.001.00 ДП ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

Також, при замовленні товару, буде надходити повідомлення з назвою товару та даними користувача, а також датою замовлення.

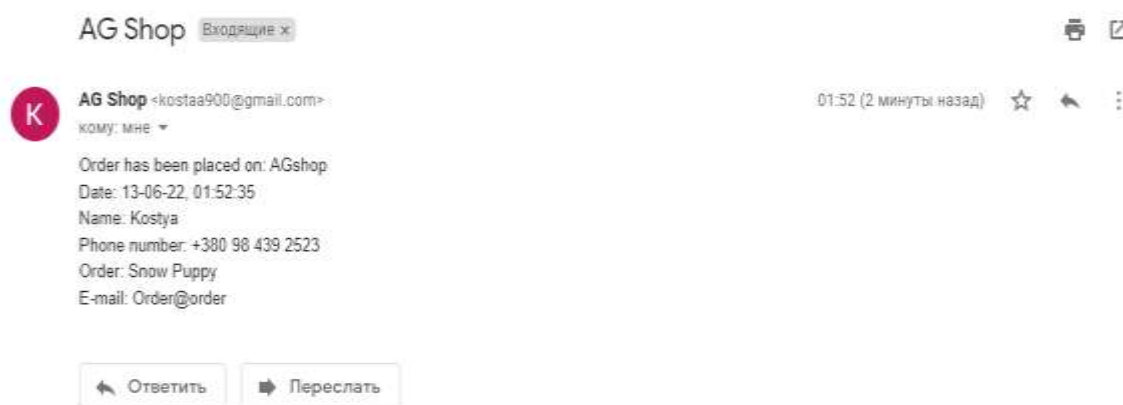


Рисунок 1.9.2 – Скріншот повідомлення заяви замовлення товару

					КТ.05.13.001.00 ДП ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

2. ЕКОНОМІЧНИЙ РОЗРАХУНОК

В дипломному проєкті був створений сайт інтернет-магазину. Інтернет-магазин це просте рішення при створенні нового бізнесу, він не потребує багато витрат, у порівнянні зі звичайним магазином простий у створенні та підтримці.

Розрахунок економічної ефективності розробки створеного web-сайту.

Загальні витрати (B_3) на створення сайту складаються з декількох параметрів:

$$B_3 = B_p + B_v + B_e$$

де B_p – витрати на розробку сайту;

B_v – витрати на впровадження сайту;

B_e – витрати на експлуатацію сайту;

Витрати на розробку сайту (B_p) є одноразовими та складаються з вартості наступних видів робіт зі створення сайту:

1. Розробка дизайну сайту: розробка логотипу.
2. Реалізація на сайті меню: випадаючого меню, «бургер» меню.
3. Підготовка сторінок-шаблонів
4. Наповнення сайту інформацією: наповнення та форматування web-сторінок; обробка малюнків для публікації на web-сторінках, верстка (переклад в HTML-формат) web-сторінок
5. Програмна розробка сайту: створення програмного коду сайту, програмування динамічних елементів.
6. Налаштування модулів: модуля новин, модуля документів (створення лістингу документів), модуля каталогу товарів, модулів інтернет-магазину (каталог, замовлення), модуля конструктора форм, модуля e-mail форм, модуля форми телефону.

Для визначення витрат на розробку сайту (B_p) розраховуємо оплату праці виконавців, безпосередньо притягнених до її виконання. Реалізацією та розробкою інтернет-магазину займається програміст, дизайнер, верстальник.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

Для визначення трудомісткості розробки сайту (Вр) складено план-графік по розробці web-сайту і тривалості виконання робіт. Розподіл робіт по етапах і видах виконавців наведено в таблиці 3.1.

Таблиця 3.1 – План-графік по розробці Web-сайту

№	Назва етапу	Час виконання (годин)	Посада виконавця
1	Розробка логотипу	2	Дизайнер
2	Реалізація меню	6	Верстальник
3	Підготовка шаблонів	12	Верстальник
4	Наповнення сайту інформацією	12	Верстальник
5	Програмна розробка сайту	50	Програміст
6	Налаштування модулів	30	Програміст
ВСЬОГО:		112	

Розрахунок трудомісткості здійснений в наступній послідовності:

1. Складений перелік всіх етапів і видів робіт, які необхідно виконати в ході даної розробки. Після узгодження з керівником проекту допущено виключення, доповнення, об'єднання окремих етапів і видів робіт;

2. По кожному виду робіт визначений кваліфікаційний рівень виконавців. В разі виконання однієї роботи виконавцями різної кваліфікації, робота розподілена на ряд паралельних конкретних робіт для кожної категорії виконавця.

В умовах відсутності нормативної бази тривалість виконання окремих робіт розраховуємо на основі вірогідних оцінок робіт.

Розмір заробітної плати розраховуємо виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а також їх середньої заробітної плати (ставки) за годину.

При визначенні вартості виконуваних робіт орієнтуємося на мінімальну заробітну плату, встановлену відповідно до «Закону про Державний бюджет

					КГ.05.13.001.00 ДП ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

України» (станом на 1.01 поточного року), враховуючи кваліфікацію виконавців, Витрати на заробітну плату приведені в таблиці 3.2. (6500 грн – з/п , 39,60 - в годину)

Таблиця 3.2 – Витрати на заробітну плату

№	Персонал	Етапи розробки	Кількість робочих годин	Погодинна ставка грн.	Заробітна плата, грн.
1	Дизайнер	Розробка логотипу	2	39,60	72,20
2	Верстальник	Реалізація меню	6	39,60	237,60
3	Верстальник	Підготовка шаблонів	12	39,60	475,20
4	Верстальник	Наповнення сайту інформацією	12	41	492
5	Програміст	Програмна розробка сайту	50	39,60	1980
6	Програміст	Налаштування модулів	30	50	1500
ВСЬОГО:					$V_{зп} = 4757$

До складу витрат на оплату праці також включаються податки, збори і інші обов'язкові платежі, встановлені системою оподаткування що діє. Розмір єдиного соціального внеску складає 22% від заробітної плати, розраховується за наступною формулою:

$$V_{есв} = V_{зп} \times 0,22 = 4757 \times 0,22 = 1046,54$$

Загальні витрати (V_p) на розробку веб-сайту розраховуються як сума витрат на заробітну плату праці персоналу ($V_{зп}$) та єдиного соціального внеску ($V_{есв}$):

$$V_p = V_{зп} + V_{есв} = 4757 + 1046,54 = 5803,54$$

Витрати на впровадження сайту (V_v)

У цю категорію відноситься витрати на реєстрацію доменного імені на 1 рік. Назва доменного імені буде com.ua, що буде коштувати 280,19 гривень

Витрати на впровадження $V_v = 280,19$ гривень

Загальні витрати (V_3) на розробку, впровадження та експлуатацію веб-

сайту розраховуються за наступною формулою:

$$B_3 = B_p + B_v = 5803,54 + 280,19 = 6083,73$$

Економічна ефективність за рік (E_p) визначається як сукупність коштів, вивільнених за рахунок впровадження сайту:

$$E_p = E_{p1} + E_{p2}$$

Вивільнення коштів будуть досягнуті за рахунок:

E_{p1} – скорочення витрат коштів на поширення прайс-листів, візитних карток та буклетів. За створення дизайну кожної візитки підприємство витрачає приблизно 450 гривень, за печать візиток кількістю 500 одиниць підприємство заплатить 600 гривень. Дизайн буклету залежно від його дизайну, кількості сторінок коштує в середньому 650 гривень, та його печать коштує приблизно 800 гривень на 500 копій. Також скорочуються кошти на рекламу, так як добре налаштований сайт буде висвічуватися користувачу одним з перших при запиті, що дає змогу заощадити приблизно 1500 гривень на рекламу, знизивши кількість придбаної реклами.

E_{p2} – скорочення засобів завдяки скорочуванню робітників, які займаються рекламою та дизайном або залучення їх для інших проєктів, що заощадить підприємству 6500 гривень.

Таким чином економічна ефективність $E_p = 450 + 600 + 650 + 800 + 1500 + 6500 = 10500$ гривень.

Коефіцієнт економічної ефективності за перший рік вираховується таким чином:

$$K_e = E_p / B_3 = 10500 / 6083,73 = 1,73$$

Функціональна ефективність проявляється:

- в забезпеченні повноти, точності і доступності інформації про організацію, її діяльність, товари і послуги у будь-який слушний для користувача час доби;

- в оптимізації технологічних процесів (процесу обробки замовлень на товари і послуги)

					КТ.05.13.001.00 ДП ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

Соціальна ефективність виражається у інформуванні найбільшого числа зацікавлених осіб про діяльність організації і про умови укладення договору на виконання робіт цією організацією, оптимізація рекламної діяльності організації, пошук нових клієнтів і ділових партнерів.

					КТ.05.13.001.00 ДП ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

3 Охорона праці

Вступ

Охорона праці на виробництві завжди була дуже важлива, отже саме завдяки рекомендаціям з охорони праці, персонал, який працює на підприємстві створює алгоритм виконання робочих завдань з чітким дотриманням рекомендацій. Основне завдання охорони праці – це створення та проведення заходів, спрямованих на захист життя, працездатності та здоров'я людини у процесі трудової діяльності.

При роботі з комп'ютером, як і в багатьох інших галузях, повинні враховуватись нормативи освітлення, температура, відносна вологість і сили вібрації. Але при роботі у приміщенні з комп'ютером найважливішим є дотримання правил пожежної безпеки, це вогнестійкість приміщення, також рівень звукового шуму, характеристики електромагнітних, ультрафіолетових та інфрачервоних полів.

Для аналізу охорони праці у дипломному проєкті досліджується безпека розробника веб-сторінок у офісному приміщенні.

1. АНАЛІЗ ТА БЕЗПЕКА УМОВ ПРАЦІ ПРАЦІВНИКА НА РОБОЧОМУ МІСЦІ

Під час будь-якого виду роботи за комп'ютером, працівник наражає себе на небезпечні фактори виробничого середовища, а саме: фізичні та психофізіологічні небезпечні й шкідливі виробничі фактори.

Серед фізичних небезпечних факторів, найпоширеніші це підвищена температура повітря робочої зони, підвищений рівень шуму, знижена вологість повітря – це звичайні фактори, які виникають при роботі у приміщеннях з комп'ютерами, через їх роботу на робочому місті підіймається температура та знижається вологість повітря. Окрім цього, комп'ютер випромінює електростатичні та електромагнітні поля у діапазоні від 5 Гц до 2 кГц та від 2 до 400 кГц, тож робота за комп'ютером включає ще підвищений рівень

					КГ.05.13.001.00 ДП ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

електромагнітний випромінювання та підвищений рівень статичної електрики. У офісних приміщеннях не завжди є достатня кількість природного освітлення у такому разі присутня велика кількість штучного освітлення, яке у свою чергу не завжди правильно налаштоване, з цього виникає, що світло може бути недостатньо яскравим або дуже яскравим.

Психофізіологічні виробничі небезпечні фактори поділяються на фізичні перевантаження та нервово-психічні перевантаження, при роботі з комп'ютером найчастіше друге. У нервово-психічних перевантаженнях програміст зазнає перенапругу аналізаторів та монотонність праці, інколи, ще й розмовну перенапругу, коли розробнику потрібно складати технічне завдання разом з клієнтом.

Розробка заходів з охорони праці

Виробниче освітлення

Штучне освітлення в приміщеннях з робочими місцями, обладнаними ВДТ має здійснюватися системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, у разі переважної роботи з документами, допускається застосування системи комбінованого освітлення (крім системи загального освітлення, додатково встановлюються світильники місцевого освітлення).

Мікроклімат

Також, при роботі у приміщеннях з великою кількістю комп'ютерів, приміщення з якими класифікуються як приміщення з підвищеною небезпекою електротравм, температура повітря влітку може становити більше 35 С, що дуже погано впливає на здоров'я людини, тож у таких приміщеннях повітря повинне охолоджуватись та понижена вологість повітря повинна регулюватись спеціальним обладнанням.

Відповідно до норм ДСН 3.3.6.042-99 температура повітря в офісі повинна становити 22-25 С, вологість повітря 40-60%, швидкість руху повітря не більше 0,1 м/с. Якщо ці норми перевищені, робочий день працівника повинен бути скорочений на 10%.

					КГ.05.13.001.00 ДП ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

Пожежна безпека

Під час роботи на підприємстві на працівників впливають різні фактори, серед яких електричний струм, освітленість робочого місця та дуже важливо – забезпечення пожежної безпеки на об'єкті праці.

Для персоналу на підприємстві повинні організовувати тренінги поведінки з електротехнікою, евакуаційні тренінги, на яких розповідають і показують де знаходяться аварійні виходи та як їх знайти у складних ситуаціях, так як найбільший відсоток смертей під час пожежі пов'язаний з панікою і отруєнням чадним газом через неможливість знайти вихід з офісу.

Проводити тренінги повинна спеціально навчена людина з охорони праці. У разі здійснення змін щодо модернізації приміщення, перепланування приміщення потрібно заздалегідь провести експертизу, щодо відповідності актам з питань пожежної безпеки у органах державного пожежного нагляду.

Прохід до аварійних виходів повинен бути вільний, не менше 1 метру, у разі великої кількості горючих відходів потрібно використовувати відведені сміттєзбірники. Електроприлади повинні використовуватися тільки для їхнього прямого призначення, а у разі пошкодження приладів, слід вимкнути їх живлення та привести до пожежобезпечного стану.

Знання щодо використання вогнегасників дуже важливе у ситуаціях виникнення пожежі. Для вдалого гасіння пожежі потрібно правильно обрати тип вогнегасника та їх кількість, відстань від найвіддаленішого місця у офісі до вогнегасника не повинна перевищувати 20 метрів. Згідно Правил на кожному поверсі будинку адміністративного призначення повинно знаходитися не менше двох вогнегасників з масою заряду вогнегасної речовини 5 кг і більше. Експлуатація вогнегасників без призначення відповідального за організацію цієї роботи не допускається. Раз на місяць відповідальний повинен проводити огляд вогнегасників при їх експлуатації, перевірити пломби, пошкодження, маркування, корозію, тиск. Відповідальний повинен мати ліцензію відповідно до ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги».

					КГ.05.13.001.00 ДП ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

Забороняється палити на підприємстві, крім спеціально виведених для цього місцях, забороняється зберігати легкозаймисті матеріали, такі як папір ближче ніж 1 метр від електрощитів, 0,15 м від приладів центрального водяного опалення та 0,6 м від сповіщувачів автоматичної пожежної сигналізації, також документація повинна зберігатися у спеціально відведених для цього шафах.

Для запобігання розповсюдження пожежі встановлюють протипожежні системи, які складаються з датчиків, звукових сповіщувачів, аварійних кнопок, приймально-контрольної панелі, яка виступає як аналізатор інформації, яку отримали датчики і відправляє ці данні на пульт пожежної охорони. Протипожежна сигналізація призначення для виявлення пожежі на початковому етапі.

Підприємство крім установки пожежної сигналізації на своєму об'єкті, має укласти договір на обслуговування даної системи з фірмою, що має на це ліцензію. В обслуговування входить проведення встановлених нормами регламентних робіт, а так само усунення несправностей в роботі системи. Періодичність перевірки узгоджується з замовником, але повинна бути не рідше ніж один раз на місяць.

У разі, якщо пожежі не вдалось уникнути, необхідно:

1. терміново повідомити пожежну охорону по телефону 101, вказати при цьому адресу, кількість поверхів, місце виникнення пожежі, наявність людей, своє прізвище;
2. організувати евакуацію людей та матеріальних цінностей;
3. повідомити про виникнення пожежі адміністрацію та чергового (за його наявності);
4. вимкнути, у разі необхідності, струмоприймачі та вентиляцію;
5. розпочати гасіння пожежі наявними первинними засобами пожежогасіння;
6. організувати зустріч підрозділів пожежної охорони й надати їм консультаційну та іншу допомогу в процесі гасіння пожежі.

					КГ.05.13.001.00 ДП ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновок

Під час дипломного проєктування було створенно інтернет-магазин з продажу фотографій використовуючи мову гіпертекстової розмітки, каскадні таблиці стилів, використовуючи препроцесор Sass, мову програмування JavaScript та PHP. Для перевірки роботи сайту після створення використовувалось програмне забезпечення Open Server, яке дозволяє створити локальний сервер. Під час виконання диплому були дотримані усі правила з охорони праці.

					<i>КТ.05.13.001.00 ДП ПЗ</i>	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

Перелік використаних джерел

1. <https://github.com/francoischalifour/medium-zoom>
2. <https://uxplanet.org/the-ultimate-guide-to-the-hamburger-menu-and-its-alternatives-e2da8dc7f1db>
3. https://medium.com/@layne_celeste/htmlcollection-vs-nodelist-4b83e3a4fb4b
4. <https://api.jquery.com/>
5. <https://imask.js.org/>
6. <https://jqueryvalidation.org/>
7. <https://github.com/nosir/cleave.js>
8. <https://github.com/PHPMailer/PHPMailer>
9. <https://zakon.rada.gov.ua/laws/show/2694-12#Text>
10. <https://saucelabs.com/resources/articles/selenium-tips-css-selectors>
11. <https://swiperjs.com/>
12. <https://nosir.github.io/cleave.js/>

					КТ.05.13.001.00 ДП ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		