МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

**Создание веб-сайта**

(наименование темы проекта или работы)

ОТЧЕТ

по лабораторной работе

по дисциплине

Информатика и компьютерные технологии

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Савкин А.Е. .

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хасанова К.К. .

(подпись) (фамилия, и.,о.)

24-ИСз .

(шифр группы)

**Задание к выполнению лабораторной работы № 2**

**Задание на лабораторную работу:**

Создание веб-сайта

**Цель работы:**

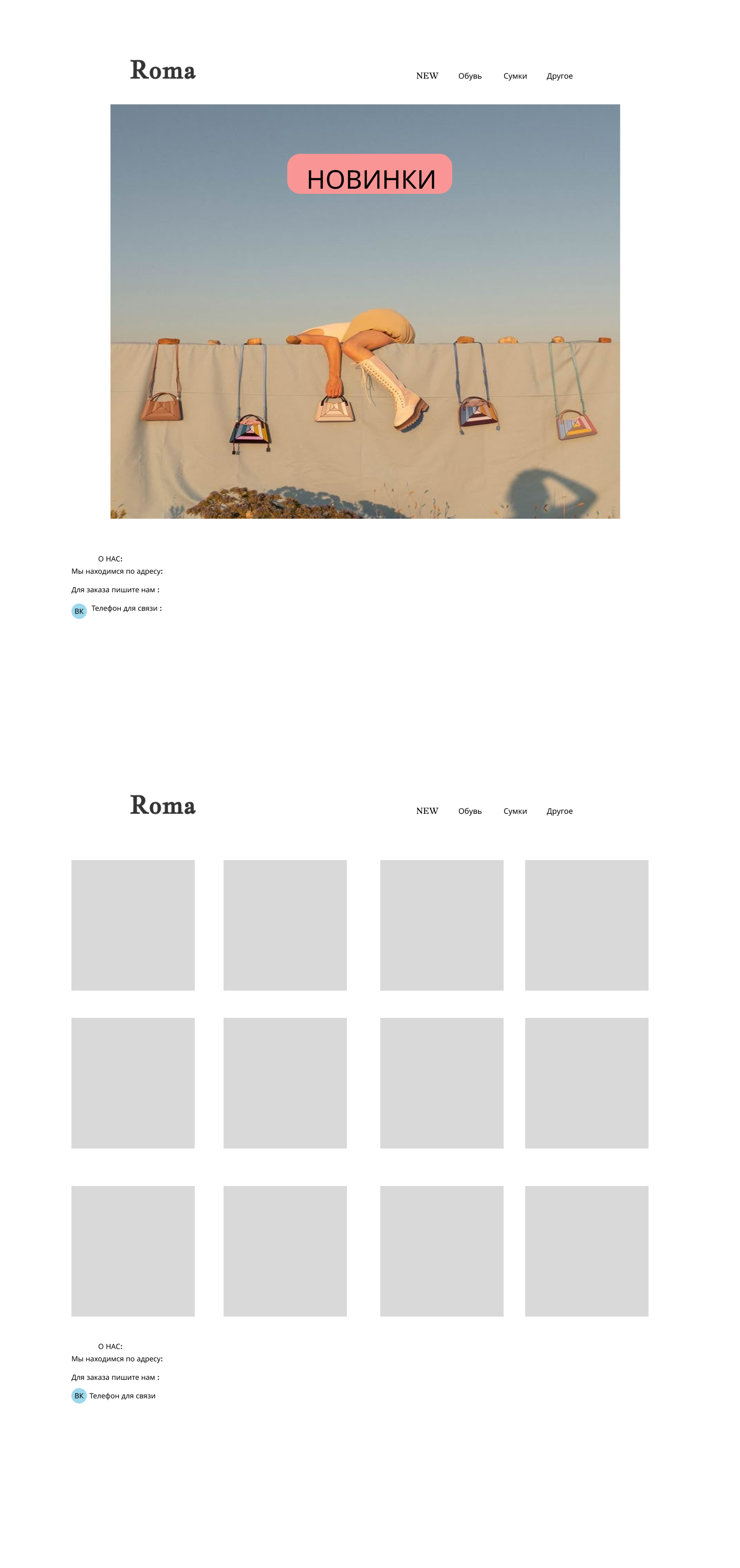
Научится создавать веб-сайт используя html, css, js.

**Ход работы:**

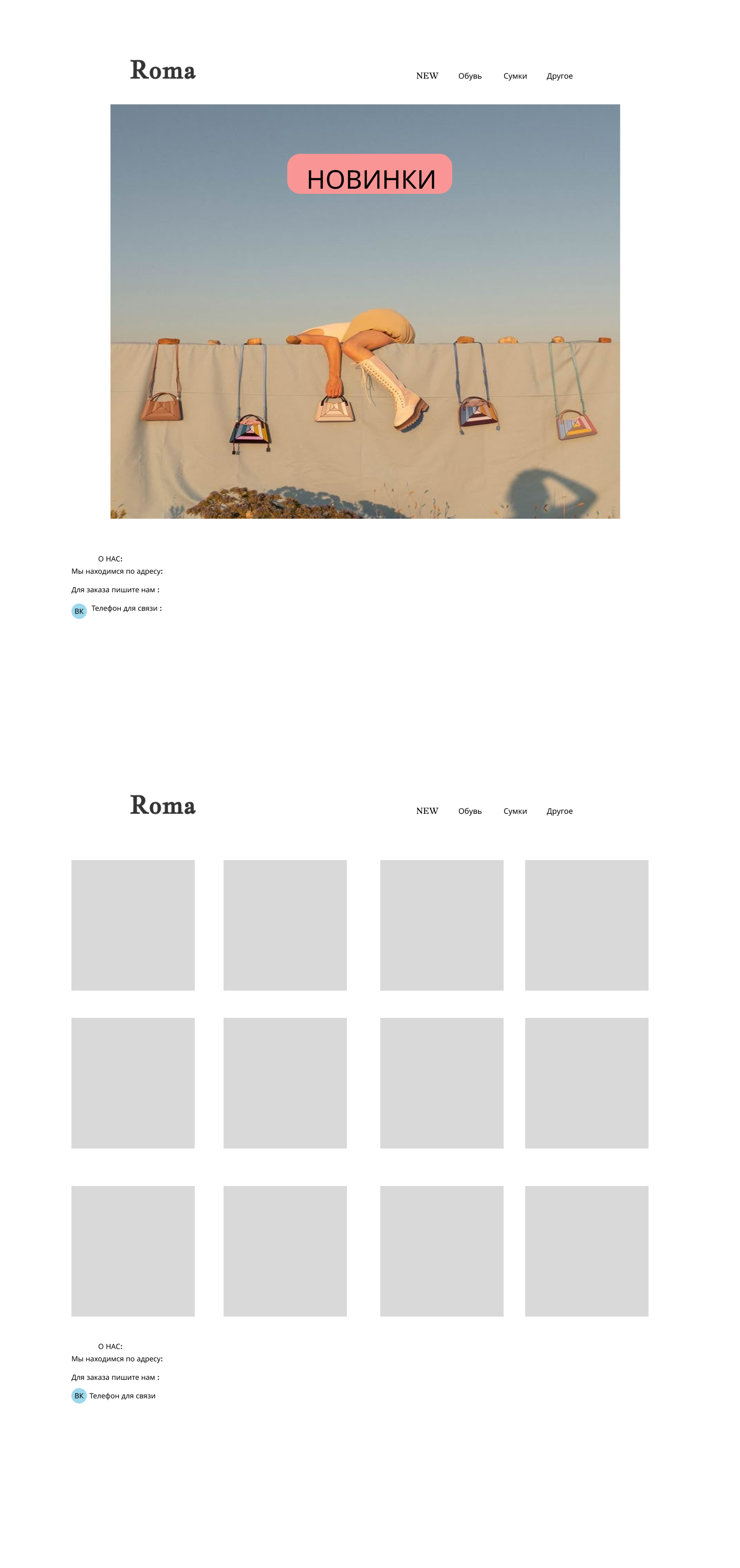
Для реализации данной лабораторной работы я выбрала тему создание каталога товаров магазина.

*1.Создание макета сайта в Figme*

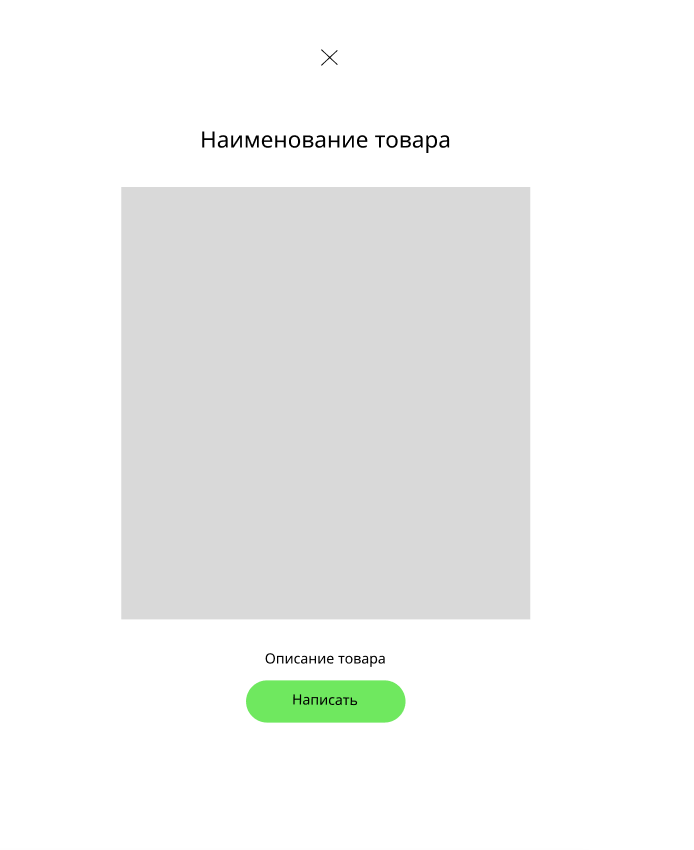
1.1 Главная страница



1.2 Визуал вкладок:



1.3 Карточка товара



*2 Написание кода в Visual Studio Code:*

2.1 Создание структуры каталогов

- папка img содержит все изображения ;

- index.html содержимое главной страницы;

- styles.css стиль , визуал сайта;

- new.html новинки;

- bag.html сумки;

- shoes.html обувь;

- other.html другое;

- script.js модальное окно - карточка товара.

2.2 Создание панели навигации

<header>

    <a href="index.html" class="logo" onclick="handleLogoClick(event)">Roma</a>

    <nav class="nav">

          <div class="item"><a href="new.html">NEW</a></div>

          <div class="item"><a href="shoes.html">Обувь</a></div>

         <div class="item"><a href="bag.html">Сумки</a></div>

          <div class="item"><a href="other.html">Другое</a></div>

    </nav>

</header>

<header> - контейнер для верхней части веб-страницы.

<a> - элемент ссылки, который позволяет пользователю перейти на другую страницу

class="logo": Присвоение класса для стилизации логотипа.

onclick="handleLogoClick(event)" - JS-обработчик события, который реагирует на клик по логотипу.

<nav class="nav"> - Это элемент навигации, который содержит все ссылки на другие страницы.

<div class="item">: Обертка для каждого элемента меню.Класс item используется для стилизации элементов меню

<a href="new.html">NEW</a>: Ссылка на страницу с новинками и т.д.

В СSS файле прописана:

Структура и оформление хедера:

Правило для header позволяет разместить элементы навигации и логотип горизонтально, а также задает фон, отступы и тень, что улучшает визуальную составляющую.

Логотип:

Класс .logo используется для стилизации логотипа, задавая ему шрифт, размер и цвет.

Навигационная панель:

Класс .nav и вложенные .item позволяют создать эффективный и визуально аккуратный интерфейс для навигации. Они обеспечивают правильный шрифт и отступы между пунктами.

Убраны подчеркивания у ссылок:

Правило .item a убирает подчеркивание у ссылок.

header {

    display: flex; /\* Используем флекс для горизонтального расположения элементов \*/

    justify-content: space-between; /\* Распределяем пространство между элементами \*/

    align-items: center; /\* Выравниваем элементы по центру по вертикали \*/

    background-color: #f8f8f8; /\* Цвет фона хедера \*/

    padding: 20px 4cm; /\* Отступы 20px сверху и снизу, 4 см по бокам \*/

    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); /\* Тень под хедером \*/

}

.logo {

    font-family: 'Jacques Francois', cursive; /\* Шрифт для логотипа \*/

    font-size: 48px; /\* Размер шрифта логотипа \*/

    color: #333; /\* Цвет текста логотипа \*/

    text-decoration: none;/\* убираем подчеркивание у ссылок\*/

}

.nav {

    font-family: 'Inknut Antiqua', serif; /\* Шрифт для навигации \*/

    font-size: 18px; /\* Размер шрифта для навигационных элементов \*/

    display: flex; /\* Используем флекс для горизонтального расположения пунктов меню \*/

    gap: 20px; /\* Отступ между пунктами меню \*/

}

.item a {

    text-decoration: none;/\* убираем подчеркивание у ссылок\*/

}

.item {

    margin-top: 1cm; /\* Отступ сверху для каждого элемента меню \*/

    margin-bottom: 20px; /\* Отступ снизу для каждого элемента меню \*/

}

* 1. Нижняя панель (информация)

<div class="info-column">

    <div>

        О НАС:<br>

        Мы находимся по адресу: Казанское шоссе, 11, Нижний Новгород<br>

        Для заказа пишите нам:<br>

    </div>

    <a href="https://vk.com/roma\_boutique\_nn?search\_track\_code=9242a0ba8eNg8Mm2fJgbnHi4UfUspF2\_wibChH\_te32vYeYd7\_GpK6lIEQCxCtpKTosVoXjuDNHwJtyDF-wW&from=search" target="\_blank">

        <img src="img/Group 1 (1).png" alt="Связаться с нами" class="contact-image">

    </a>

    Телефон для связи: +7 (910) 131 00 13<br>

</div>

В СSS файле необходимо прописать:

Стиль информации о магазине:Правило для .info-column определяет, как будет выглядеть текст в этом блоке.

Изображение контакта: Класс .contact-image позволяет настроить изображение, которое будет использоваться для связи . При нажатии на который открывается страница магазина в ВК.

.info-column {

    font-family: 'Inknut Antiqua', serif; /\* Шрифт для информации о магазине \*/

    font-size: 14px; /\* Размер шрифта для информации \*/

    text-align: left; /\* Выровнять текст по левому краю \*/

    padding: 20px; /\* Отступы вокруг информации \*/

    margin-top: 20px; /\* Отступ сверху для информации \*/

}

.contact-image {

    cursor: pointer; /\* Изменяем вид курсора на "указатель" при наведении \*/

    margin-top: 10px; /\* Отступ сверху для изображения контакта \*/

    width: 24px; /\* Ширина изображения для контактов \*/

    height: 24px; /\* Высота изображения для контактов \*/

}

2.4 Содержание главной страницы файл index.html.

<div class="image-container"><!-- Контейнер для изображения новинки -->

    <img src="img/1 1.png" alt="Новинка" class="responsive-image">

    <a href="new.html" class="btn-new">Новинка</a> <!-- Кнопка "Новинка" -->

</div>

<div class="image-container"> - контейнер для изображения с кнопкой.

Тег <img> отвечает за отображение изображения на веб-странице.

Атрибут src указывает путь к изображению. Атрибут alt предоставляет текстовое описание изображения. Класс responsive-image -изображение будет адаптироваться под размеры контейнера.

<a>: - создаёт гипертекстовую ссылку, которая направляет пользователя на страницу new.html .

Класс btn-new используется для стилизации кнопки.

В CSS файле необходимо прописать:

.responsive-image {

    width: 80%; /\* растяжение изображения \*/

    height: auto; /\* Сохраняем пропорции изображения \*/

}

.image-container {

    position: relative; /\* Устанавливаем относительное позиционирование для вложенных элементов \*/

    width: 100%; /\* Ширина контейнера 100% \*/

    text-align: center; /\* Центруем содержимое внутри контейнера \*/

}

.btn-new {

    position: absolute; /\* Абсолютное позиционирование кнопки \*/

    top: 40px; /\* Отступ от верхнего края изображения \*/

    left: 50%; /\* Центрируем по горизонтали \*/

    transform: translateX(-50%); /\* Смещение для точного выравнивания по центру \*/

    background-color: #F99595; /\* Цвет фона кнопки "Новинка" \*/

    color: white; /\* Цвет текста кнопки \*/

    padding: 10px 20px; /\* Отступы внутри кнопки \*/

    border: none; /\* Убираем границу у кнопки \*/

    border-radius: 25px; /\* Скругление углов кнопки \*/

    font-size: 24px; /\* Размер шрифта кнопки \*/

    cursor: pointer; /\* Курсор при наведении над кнопкой \*/

    text-align: center; /\* Центрируем текст в кнопке \*/

    text-decoration: none; /\* Убираем подчеркивание текста в кнопке \*/

}

.btn-new:hover {

    background-color: #d68282; /\* Цвет кнопки при наведении \*/

}

2.5Создание вкладок новинки, обувь, сумки, другое:

На каждой из вкладок в теле прописывается (изменяется только путь к фотографии товара, наименование и описание при желании):

<div class="gallery">

    <img src="img/product1.jpg" alt="Товар 1" onclick="openModal('Сумка-все всегда с собой', 'Ты готова сделать этот шаг', 'img/product1.jpg')">

    <img src="img/product2.jpg" alt="Товар 2" onclick="openModal('Сумка- настроение весна', 'Ты готова сделать этот шаг', 'img/product2.jpg')">

    <img src="img/product3.jpg" alt="Товар 3" onclick="openModal('Балетки', 'Ты готова сделать этот шаг', 'img/product3.jpg')">

    <img src="img/product4.jpg" alt="Товар 4" onclick="openModal('Обвес', 'Ты готова сделать этот шаг', 'img/product4.jpg')">

    <img src="img/product5.jpg" alt="Товар 5" onclick="openModal('Колье', 'Ты готова сделать этот шаг', 'img/product5.jpg')">

    <img src="img/product6.jpg" alt="Товар 6" onclick="openModal('Обвес', 'Ты готова сделать этот шаг', 'img/product6.jpg')">

    <img src="img/product7.jpg" alt="Товар 7" onclick="openModal('Колье', 'Ты готова сделать этот шаг', 'img/product7.jpg')">

    <img src="img/product8.jpg" alt="Товар 8" onclick="openModal('Товар 6', 'Ты готова сделать этот шаг', 'img/product8.jpg')">

</div>

<div class="gallery"> - контейнером для всех изображений. Класс gallery используется для группировки изображений и применения к ним определённых стилей.

Каждый элемент <img> отображает изображение товара. Атрибут src указывает путь к файлу изображения, а атрибут alt — текстовое описание.

Атрибут onclick вызывает JavaScript-функцию openModal при нажатии на изображение, передавая параметры, которые можно использовать для открытия модального окна с информацией о товаре.

В СSS файле необходимо прописать:

Оформление галереи: Правило для .gallery помогает организовать изображения в аккуратную сетку.

Адаптивность изображений: Правила для .gallery img делают изображения адаптивными, обеспечивая, что они будут занимать всю доступную ширину контейнера и сохранять свои пропорции.

Эффект наведения: Добавление эффекта увеличения изображения при наведении (transform: scale(1.05)) .

.gallery {

    display: grid; /\* организация элементов \*/

    grid-template-columns: repeat(4, 1fr); /\* 4 колонки одинаковой ширины \*/

    gap: 20px; /\* Отступы между элементами галереи \*/

    padding: 20px; /\* Отступы вокруг галереи \*/

}

.gallery img {

    width: 100%; /\* Ширина изображений 100% от родительского элемента \*/

    cursor: pointer; /\* Курсор при наведении на изображения \*/

    border-radius: 8px; /\* Закругленные края изображений \*/

    transition: transform 0.2s; /\* Анимация для трансформации при наведении \*/

}

.gallery img:hover {

    transform: scale(1.05); /\* Увеличение изображения при наведении \*/

}

Увеличение изображения:

<div id="myModal" class="modal">

    <div class="modal-content">

        <span id="close" onclick="closeModal()" style="cursor:pointer;">&times;</span>

        <h2 id="modal-title">Название товара</h2>

        <img id="modal-image" src="" alt="Изображение товара" style="width: 100%; border-radius: 8px;">

        <p id="modal-description">Описание товара</p>

        <button onclick="window.open('https://vk.com/roma\_boutique\_nn?from=search', '\_blank')">Заказать</button>

    </div>

</div>

<div id="myModal" class="modal"> — основной контейнер для карточки товара.

<div class="modal-content"> — содержит все элементы внутри модального окна.

<span id="close" onclick="closeModal()"> — закрывает модальное окно при нажатии.

: <h2 id="modal-title"> — отображает название товара.

<img id="modal-image" src="" alt="Изображение товара"> — показывает изображение товара.

<p id="modal-description"> — текст описания товара.

<button onclick="window.open('...')">Заказать</button> — открывает ссылку для заказа товара в новой вкладке.

В CSS необходимо прописать:

Стили для модального окна: Правила для .modal позволяют создать фон для модального окна, и скрыто по умолчанию.

Стили для содержимого модального окна: Правила для .modal-content определяют, как будет выглядеть содержимое.

Стиль кнопки заказа: Правила для кнопки внутри модального окна modal-content button установлен цвет, размеры, и эффект при наведении, чтобы пользователи сразу понимали, что эта кнопка является кликабельным элементом.

/\*модальное окно\*/

.modal {

    display: none; /\* Модальное окно скрыто по умолчанию \*/

    position: fixed; /\* Фиксированное позиционирование окна относительно окна браузера \*/

    z-index: 999; /\* Уровень слоя, чтобы окно отображалось поверх других элементов \*/

    left: 0; /\* Прикрепляем окно к левому краю \*/

    top: 0; /\* Прикрепляем окно к верхнему краю \*/

    width: 100%; /\* Заполнить всю ширину экрана \*/

    height: 100%; /\* Заполнить всю высоту экрана \*/

    background-color: rgba(0, 0, 0, 0.5); /\* Полупрозрачный черный фон \*/

}

.modal-content {

    position: absolute; /\* Абсолютное позиционирование для контента модального окна \*/

    top: 50%; /\* Центрируем по вертикали \*/

    left: 50%; /\* Центрируем по горизонтали \*/

    transform: translate(-50%, -50%); /\* Смещение для точного центрирования \*/

    background-color: white; /\* Фоновый цвет контента модального окна \*/

    padding: 20px; /\* Отступы вокруг контента \*/

    border-radius: 8px; /\* Закругление углов контента \*/

    width: 80%; /\* Ширина контента 80% от родителя \*/

    max-width: 500px; /\* Максимальная ширина контента \*/

    display: flex; /\* Используем Flexbox для расположения элементов \*/

    flex-direction: column; /\* Вертикальное расположение элементов \*/

    align-items: center; /\* Центрируем элементы по горизонтали \*/

    justify-content: center; /\* Центрируем элементы по вертикали \*/

}

/\*КНОПКА В КАРТОЧКЕ\*/

.modal-content button {

    background-color: #4CAF50; /\* Зеленый цвет для кнопки \*/

    border: none; /\* Убираем границу у кнопки \*/

    color: white; /\* Цвет текста кнопки \*/

    padding: 10px 20px; /\* Отступы внутри кнопки \*/

    text-align: center; /\* Центрируем текст в кнопке \*/

    text-decoration: none; /\* Убираем подчеркивание текста в кнопке \*/

    display: inline-block; /\* Позволяет задавать ширину и высоту кнопки \*/

    font-size: 16px; /\* Размер шрифта для кнопки \*/

    margin-top: 10px; /\* Отступ сверху \*/

    cursor: pointer; /\* Курсор при наведении на кнопку \*/

    border-radius: 5px; /\* Скругление углов кнопки \*/

}

.modal-content button:hover {

    background-color: #3e8e41; /\* Более темный зеленый цвет кнопки при наведении \*/

}

Так же необходимо прописать в файле script.js

function openModal(title, description, imageSrc) {

    // Устанавливаем заголовок модального окна

    document.getElementById('modal-title').innerText = title;

    // Устанавливаем описание в модальном окне

    document.getElementById('modal-description').innerText = description;

    // Устанавливаем источник изображения в модальном окне

    document.getElementById('modal-image').src = imageSrc;

    // Показываем модальное окно, устанавливая его стиль на "block"

    document.getElementById('myModal').style.display = "block";

}

function closeModal() {

    // Скрываем модальное окно, устанавливая его стиль на "none"

    document.getElementById('myModal').style.display = "none";

}

// Закрытие модального окна при клике вне его

window.onclick = function(event) {

    // Проверяем, был ли клик именно по фону модального окна

    if (event.target == document.getElementById('myModal')) {

        // Если да, то закрываем модальное окно

        closeModal();

    }

}

Функция openModal(title, description, imageSrc)

1. Аргументы функции:

title: Заголовок, который будет отображен в модальном окне.

description: Описание, которое будет показано пользователю в модальном окне.

imageSrc: Путь к изображению, которое будет отображено в модальном окне.

1. Установка заголовка:

document.getElementById('modal-title').innerText = title;

- ищем элемент с ID modal-title и устанавливаем его текст на значение переменной title.

1. Установка описания:

document.getElementById('modal-description').innerText = description;

- находим элемент с ID modal-description и устанавливаем его текст на значение переменной description.

1. Установка изображения:

document.getElementById('modal-image').src = imageSrc;

- находим элемент с ID modal-image и устанавливаем его атрибут src на значение imageSrc, чтобы загрузить нужное изображение.

1. Отображение модального окна:

document.getElementById('myModal').style.display = "block";

- находим модальное окно по ID myModal и меняем его стиль display на block, что делает его видимым на странице.

Функция closeModal()

1. Эта функция отвечает за закрытие модального окна.

document.getElementById('myModal').style.display = "none";

- находим модальное окно по ID и устанавливаем стиль display на none, что скрывает его.

Обработчик события window.onclick

1. Этот обработчик отслеживает клики по всему окну.
2. Проверка клика:

if (event.target == document.getElementById('myModal')) {

- проверяет, был ли клик именно по фону модального окна (то есть по элементу с ID myModal).

1. Закрытие модального окна:

Если условие истинно, вызывается функция closeModal(), которая скрывает модальное окно.

Вывод: Я научилась создавать сайт каталог с использование html, css, js.

Результат:

