Eco Food (інтенет магазин)

Створити маленький інтернет магазин продуктів.

Реалізувати:

* карусель
* модальне вікно
* меню
* посилання на соцмережі
* додавання продуктів в корзину
* зміну кількості продуктів в корзині
* сторінку перегляду деталей продукта
* сторінку для входу (log in)
* кнопку повернутись на попередню сторінку
* адаптувати дизайн під різні пристрої
* меню для мобільних (гамбургер)
* кнопку догори

Використовувати HTML5, CSS3, js, а також <https://validator.w3.org/> для перевірки на коректність сого коду.

# Завдання 1

Створити папку проекта EcoFood та відкрити в редакторі.

Створити першу та головну сторінку - index.html (пишемо тут лише html код).

Розбиваємо сторінку на логічні частини: header, main, footer.

* hader: лого, кнопки: увійти, корзина та очисти корзину, меню.
* footer: три відомості про мед, лінки на соцмережі, всі права захищені
* main: решта.

Поки верстаємо без каруселі та модального вікна.

Для іконок використовуємо fontAwesom -   
<script defer src="<https://use.fontawesome.com/releases/v5.15.2/js/all.js>"></script>

# Завдання 2

Вестаємо (лиш html код) сторінку деталей - product.html та login.html ( валідація, regex).

# Завдання 3

Додаємо CSS.

Ділимо всі стилі по відповідних файликах:

* style.css - загальні стилі для всіх сторінок
* home.css - стилі для головної части index.html
* product.css - стилі до сторінки з деталями
* login.css - стилі для сторінки логування

Для hover ефектів використати - *transition: all 0.5s;*

Шрифт -

<link rel="preconnect" href="https://fonts.gstatic.com">

<link href="https://fonts.googleapis.com/css2?family=**Montserrat:wght@100;400;700**&display=swap" rel="stylesheet">

css - font-family: 'Montserrat', sans-serif;

Колір кнопки - #91ac27

# Завдання 4

Додаємо Bootstrap модальне вікно та карусель.

# Завдання 5

Створюємо адаптивний дизайн. Додаємо гамбургер.

1. @media screen and ( min-width: 320px) and(max-width: 425px)
2. @media screen and ( max-width: 320px){

* розміщуємо продукти в одну колонку на головній сторінці
* інфоблок з медом - в одну колонку, у всіх футерах

}

Стилі для звичайного меню та гамбургера можна взяти з файла menu.css.

# Завдання 6

Починаємо додавати js.

Додати колекцію продуктів на головну сторінку динамічним HTML.

Для цього видалити усі <article>, які стосуються продуктів, залишити пусту секцію без заголовка.

<section class="products">

</section>

Створити новий js файл в якому динамічно додати продукти.

let jsonProducts = '{"products": [{"id": 1, "name": "Lemon", "price": 7.6, "imgUrl": "./img/lemon.jpg"}, {"id": 2, "name": "Cauliflower", "price": 1.2, "imgUrl": "./img/cauliflower.jpg"}, {"id": 3, "name": "Corn", "price": 2.0, "imgUrl": "./img/corn.jpg"}, {"id": 4, "name": "Cucumber", "price": 4.5, "imgUrl": "./img/cucumber.jpg"}, {"id": 5, "name": "Kiwi", "price": 10.0, "imgUrl": "./img/kiwi.jpg"}, {"id": 6, "name": "Lime", "price": 9.6, "imgUrl": "./img/lime.jpg"}, {"id": 7, "name": "Peach", "price": 15.5, "imgUrl": "./img/peach.jpg"}, {"id": 8, "name": "Pumpkin", "price": 3.3, "imgUrl": "./img/pumpkin.jpg"}]}';

let productsObj = JSON.parse(jsonProducts);

# Завдання 7

Додати window.location - для переходу на сторінку product.html, а також кнопку на попередню сторінку - window.history.back();

Перхід на сторінку продукти:

window.location = 'product.html?productName=' + product.id;

Додати кнопку скрол вверх з позиції 800.

js файлик:

let buttonTop = document.getElementById('buttonTop');

// When the user scrolls down 800px from the top of the document,we will show the button

window.onscroll = function() {

if (document.documentElement.scrollTop > 800) {

buttonTop.classList.add('show');

}

else {

buttonTop.classList.remove('show');

}

};

buttonTop.addEventListener("click", scrollToTop);

function scrollToTop(event) {

document.documentElement.scrollTo({

top: 0,

behavior: "smooth"

});

}

css файлик:

#buttonTop {

background-color: #FF9800;

width: 50px;

height: 50px;

border-radius: 50%;

position: fixed;

bottom: 50px;

right: 60px;

transition: background-color .3s,

opacity .5s, visibility .5s;

opacity: 0;

visibility: hidden;

z-index: 1000;

display: flex;

justify-content: center;

align-items: center;

}

#buttonTop:hover {

cursor: pointer;

background-color: #333;

color: white;

}

#buttonTop:active {

background-color: #555;

color: white;

}

#buttonTop.show {

opacity: 1;

visibility: visible;

}

html файлик: id ='buttonTop'

# Завдання 8

Отримати параметром назву продукта, змінити html так щоб назва продукта та картинка додавалась динамічно в залежності від принятого параметра.

let queryString = new Array();

window.onload = function () {

if (queryString.length == 0) {

if (window.location.search.split('?').length > 1) {

var params = window.location.search.split('?')[1].split('&');

for (var i = 0; i < params.length; i++) {

var key = params[i].split('=')[0];

var value = decodeURIComponent(params[i].split('=')[1]);

queryString[key] = value;

}

}

}

if (queryString["productName"]) {

document.getElementById('product-name').innerHTML = queryString["productName"];

document.getElementById('product-img').src = `./img/${queryString["productName"]}.jpg`;

document.getElementById('product-img').src = './img/' + queryString["productName"] + '.jpg';

document.getElementById('product-img').alt = queryString["productName"];

}

};

# Завдання 9

Створюємо функціонал addToCart:

* знаходимо всі кнопки Add to Cart, getElementsByClassName
* додаємо слухача addEventListener
* додаємо функцію addToCart

addToCart:

* збільшити лічильник загальної кількості продуктів на головній сторінці
* додати в масив продуктів новий продукт, якщо його немає або збільшити лічильник, якщо продукт вже існує в масиві

# Завдання 10

Створюємо функціонал clearCart:

* знайти елемент clear-cart, додати слухача
* створити функцію clearCart
* обнулити лічильник count
* очистити масив продуктів

# Завдання 11

Створюємо функціонал showCart:

* знайти елемент show-cart, додати слухача на нього
* створити функцію showCart
* за допомогою js створюємо рядки та стовпці таблиці (ім’я, ціна, кількість, загальна ціна по продукту)

Продовжуємо реалізацію. Додаємо можливість змінити кількість продуктів у корзині.

* шукаємо елемент кількості продукту, додаємо слухача на input

На завершення нашої функції додаємо функцію, яка б рахувала загальну ціну корзини.

# Завдання 12

Створюємо функцію для зміни ціни по продукту відносно нової кількості - priceByProduct.

* отримати нову кількість по продукту та його ід
* знайти цей продукт в масиві та змінити на відповідну кількість
* змінити загальну ціну по продукту в таблиці
* змінити загальну ціну корзини

# Завдання 13

Створити функцію totalPrice.

Пробігтись по масиву продуктів та порахувати загальну вартість, записати в таблицю.

# Завдання 14 (додаткове)

Реалізувати видалення продукта з корзини: додати кнопку видалити та якщо кількість змінено до нуля також видалити продукт.

# Завдання 15 (додаткове)

Запитати користувача чи він дійсно хоче видалити продукт з корзини.

# Завдання 16 (додаткове)

Заливаємо проект на github.

<https://docs.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line>

# Додаткові матеріали

<https://docs.emmet.io/cheat-sheet/> - emmet

<https://validator.w3.org/> - validator HTML

<https://jigsaw.w3.org/css-validator/> - validator CSS

<https://eslint.org/> - перевіряє код на помилки перед git commit