ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОУ ВПО НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Р.Е. АЛЕКСЕЕВА

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ

ТЕХНОЛОГИЙ

Кафедра "Вычислительные системы и технологии"

**WEB-технологии**

**Отчёт**

**По лабораторной работе №1**

Выполнил студент группы 22-ПО

Перевалов Даниил Дмитриевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_«\_\_\_» \_\_\_\_\_\_\_ 20\_\_\_ г. (личная (личная подпись) (дата)

Проверил

Скорынин Сергей Сергеевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_«\_\_\_» \_\_\_\_\_\_\_ 20\_\_\_ г. (лич (личная подпись) (дата)

Нижний Новгород 2024

**Постановка задачи**

Создать страницу со скриптом, в котором создается объект Холодильник с 4-мя свойствами: цвет, марка, модель, цена. Вывести по нажатию кнопки все значения свойств на экран.

**Стек используемых технологий**

HTML – язык гипертекстовой разметки, необходим для описания иерархии объектов.

CSS – язык таблиц стилей, необходим для описания внешнего вида страницы в целом и объектов в отдельности.

JS – скриптовый язык программирования, необходим для описания алгоритма динамического изменения страницы и внутренней логики работы сайта.

**Описание проекта**

Проект состоит из 3х файлов:

index.html

Отвечает за базовую структуру и отображение интерфейса. Основной HTML-документ с интерфейсом для управления холодильником. Содержит три кнопки: для создания холодильника, отображения/скрытия его характеристик, и удаления. Есть контейнер для изображения холодильника и блока с информацией.

При загрузке страницы кнопки для управления холодильником скрыты.

По нажатию на кнопки вызываются функции из logic.js, которые динамически изменяют видимость элементов и наполняют их данными.

look.css

Делает интерфейс более привлекательным и удобным для восприятия.

Стилизует внешний вид приложения:

* Фон градиентный.
* Элементы (заголовок, кнопки, изображения) имеют стили для выравнивания, цветов, размера и анимации.

Оформление кнопок: меняют цвет при наведении. Заголовок выделен белым текстом с черным обводом. Изображение холодильника и детали оформлены как отдельный блок с плавными стилями. Используется CSS-анимация для кнопок через transition. Блок с характеристиками полупрозрачный, что улучшает читаемость на фоне.

logic.js

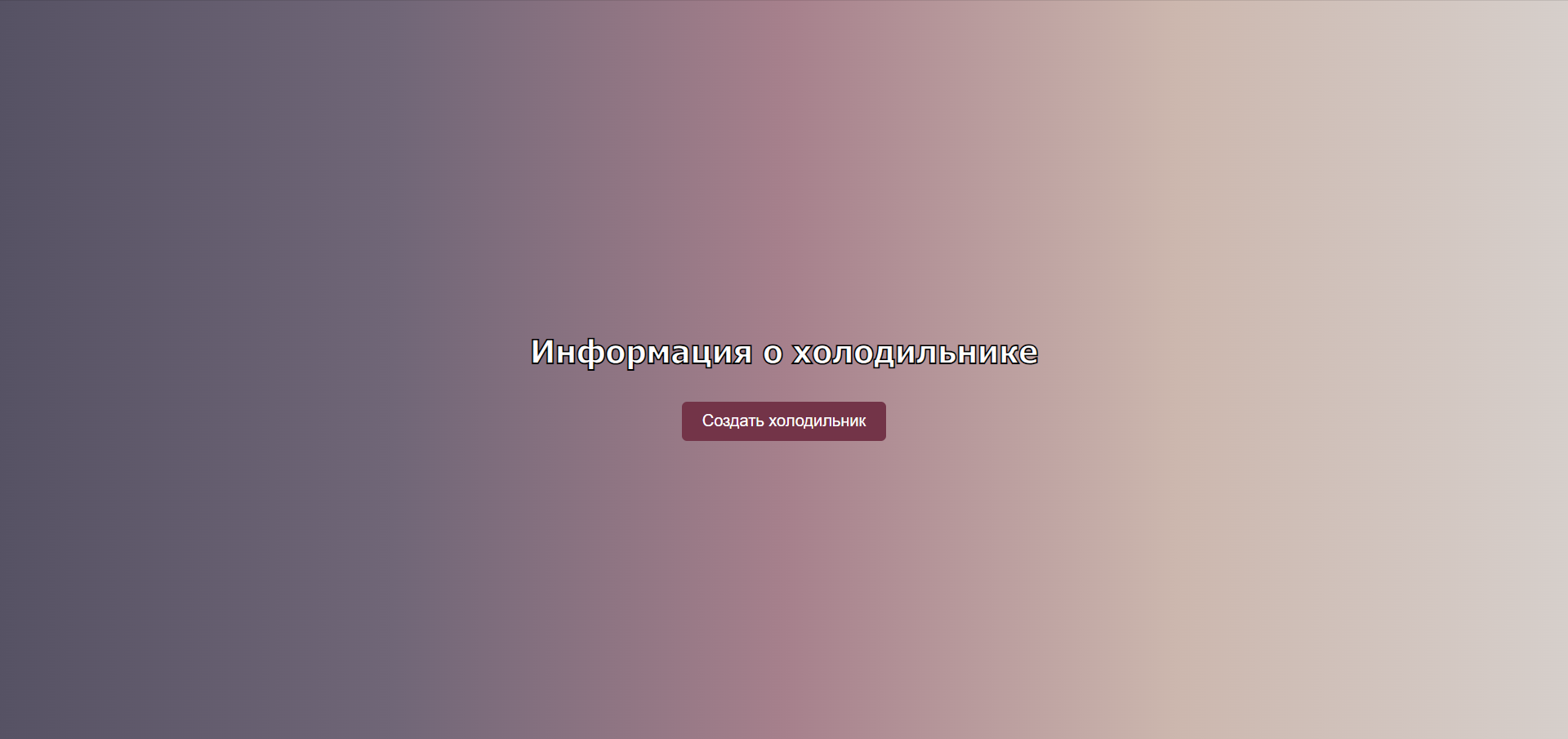
Отвечает за интерактивность интерфейса. Управляет состоянием элементов в зависимости от действий пользователя. Реализована логика создания, отображения/скрытия характеристик и удаления холодильника.

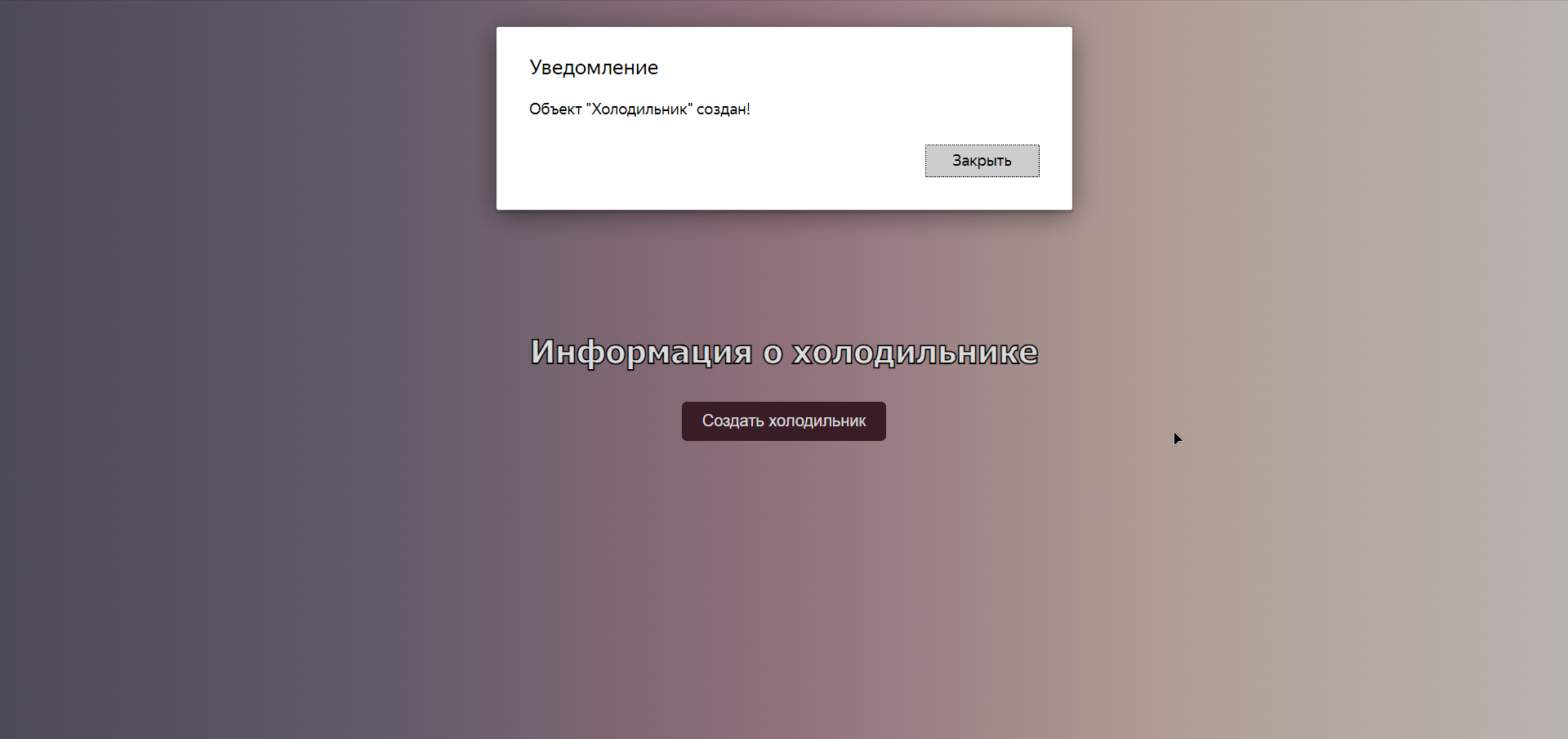
Алгоритм работы:

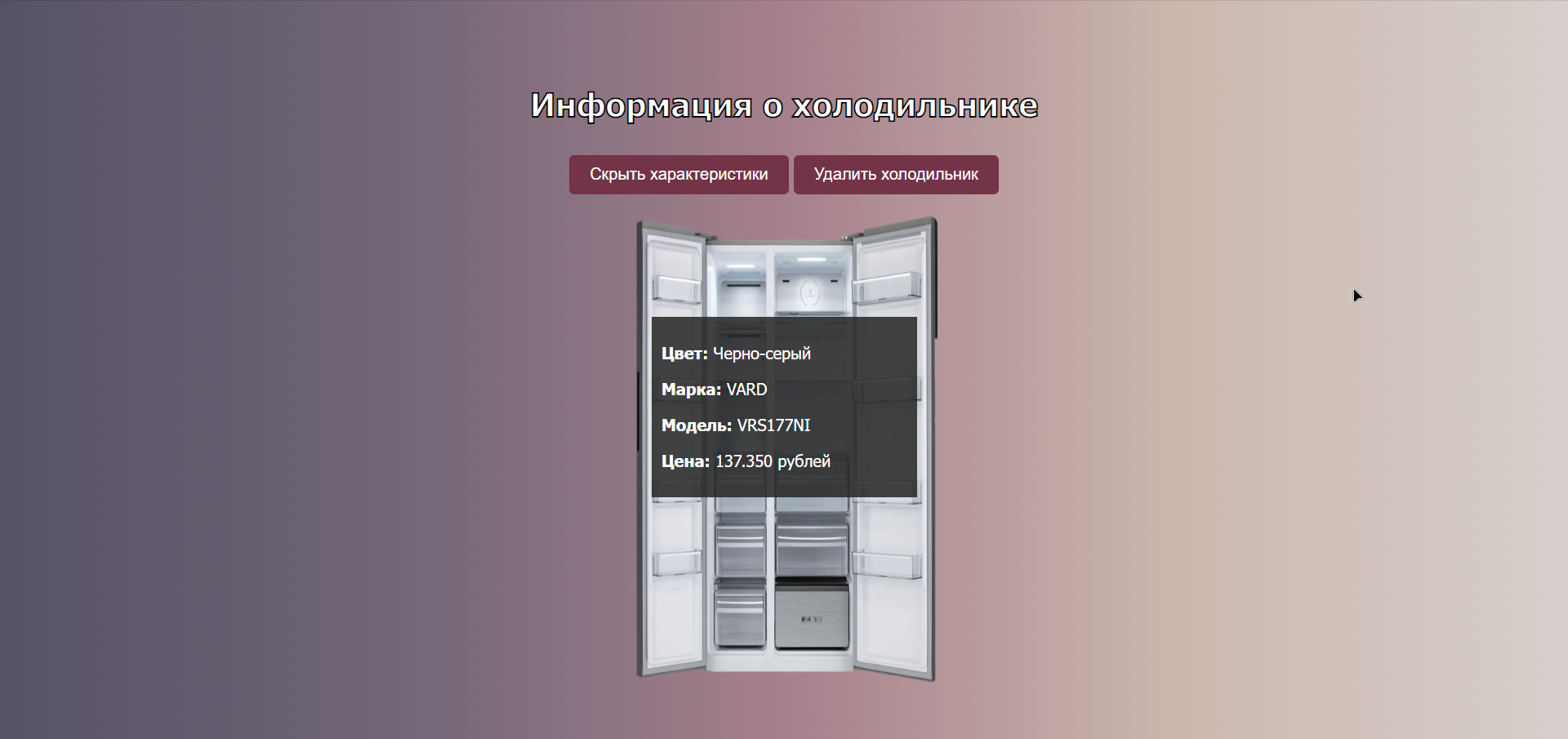
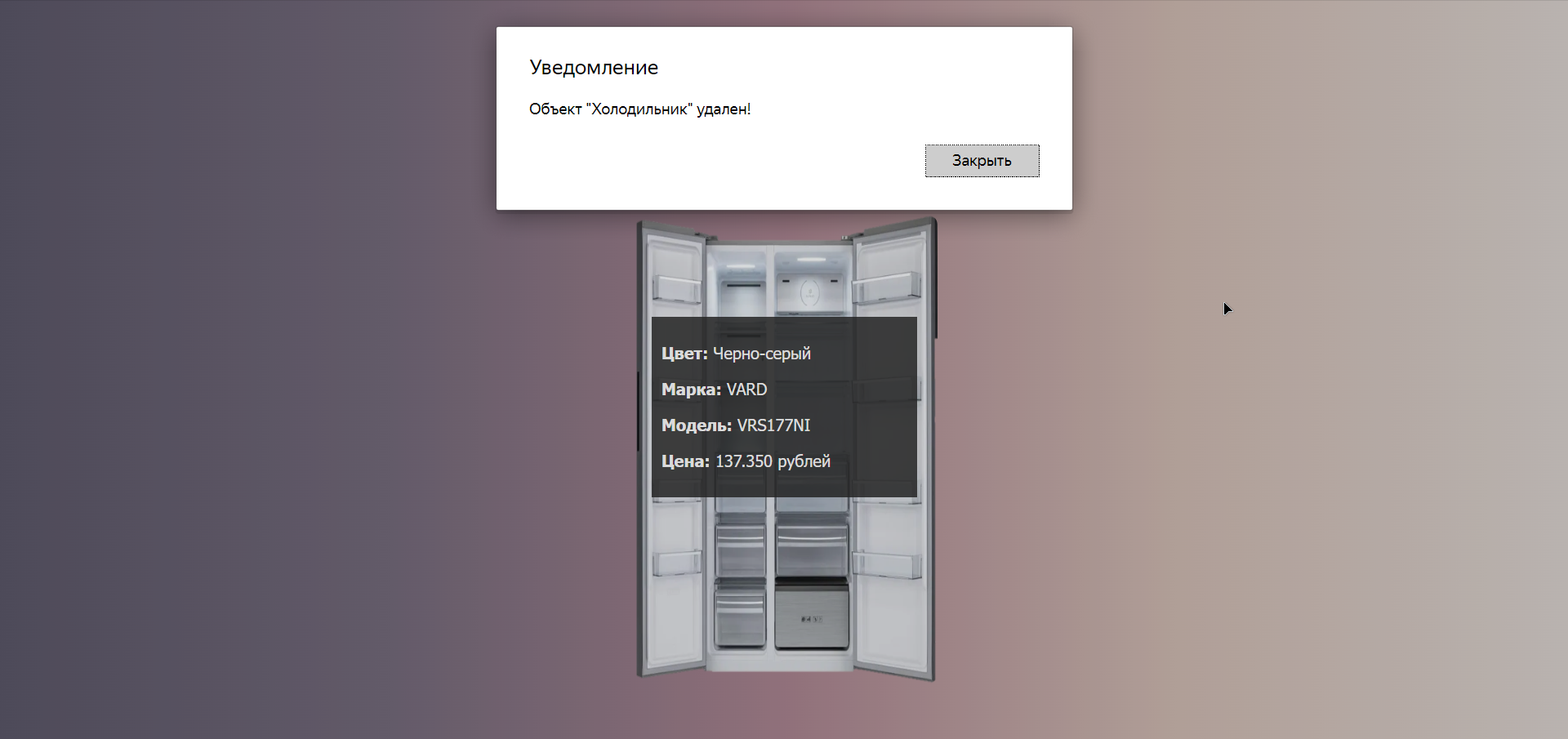
1. Создание холодильника:
   * Объект с данными (fridge) заполняется информацией о холодильнике.
   * Показываются изображение и кнопки управления.
2. Переключение характеристик:
   * Меняется изображение (открытое/закрытое).
   * Характеристики отображаются/скрываются в зависимости от состояния.
3. Удаление холодильника:
   * Удаляется объект, сбрасывается состояние интерфейса.

Используется флаг fridgeOpen для отслеживания состояния (открыт/закрыт). Привязка событий к кнопкам выполнена через addEventListener. Сообщения пользователю выводятся через alert.

**Пример работы**





**Код проекта**

index.html

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Холодильник</title>

<link rel="stylesheet" href="look.css">

<script defer src="logic.js"></script>

</head>

<body>

<h1>Информация о холодильнике</h1>

<div class="form-container">

<button id="createFridge">Создать холодильник</button>

<button id="toggleDetails" style="display: none;">Показать характеристики</button>

<button id="deleteFridge" style="display: none;">Удалить холодильник</button>

</div>

<div id="fridgeContainer" class="fridge-container" style="display: none;">

<img id="fridgeImage" src="closed\_fridge.png" alt="Закрытый холодильник" class="fridge-image">

<div id="fridgeDetails" class="fridge-details" style="display: none;"></div>

</div>

</body>

</html>

look.css

body {

font-family: 'Tahoma', sans-serif;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

margin: 0;

background: linear-gradient(to right, #565264, #706677, #a6808c, #ccb7ae, #d6cfcb);

}

h1 {

text-align: center;

color: #ffffff;

-webkit-text-stroke: 1.5px #000000;

font-size: 32px;

border-radius: 5px;

padding: 10px 20px;

display: inline-block;

}

.form-container {

text-align: center;

}

button {

padding: 10px 20px;

font-size: 16px;

background-color: #733448;

color: #fff;

border: none;

cursor: pointer;

border-radius: 5px;

transition: background-color 0.1s ease;

}

button:hover {

background-color: #3d1c27;

}

.fridge-container {

position: relative;

margin-top: 20px;

}

.fridge-image {

width: 300px;

height: 460px;

border-radius: 10px;

}

.fridge-details {

position: absolute;

top: 100px;

left: 20px;

color: #fff;

font-size: 16px;

background-color: rgba(36, 36, 36, 0.85);

padding: 10px;

width: 80%;

}

logic.js

let fridge = null; // Переменная для хранения объекта холодильника

let fridgeOpen = false; // Статус холодильника (открыт/закрыт)

// Функция для создания объекта холодильника

function createFridge() {

fridge = {

color: "Черно-серый",

brand: "VARD",

model: "VRS177NI",

price: "137.350 рублей"

};

// Показываем контейнер с изображением холодильника и кнопки управления

document.getElementById("fridgeContainer").style.display = "block";

document.getElementById("toggleDetails").style.display = "inline-block";

document.getElementById("deleteFridge").style.display = "inline-block";

// Скрываем кнопку "Создать холодильник"

document.getElementById("createFridge").style.display = "none";

alert('Объект "Холодильник" создан!');

}

// Функция для показа или скрытия характеристик холодильника

function toggleFridgeDetails() {

const fridgeImage = document.getElementById("fridgeImage");

const fridgeDetails = document.getElementById("fridgeDetails");

// Переключаем состояние холодильника (открыт/закрыт)

fridgeOpen = !fridgeOpen;

if (fridgeOpen) {

// Меняем изображение на "открытое"

fridgeImage.src = "open\_fridge.png";

fridgeImage.alt = "Открытый холодильник";

// Отображаем характеристики холодильника

fridgeDetails.innerHTML = `

<p><strong>Цвет:</strong> ${fridge.color}</p>

<p><strong>Марка:</strong> ${fridge.brand}</p>

<p><strong>Модель:</strong> ${fridge.model}</p>

<p><strong>Цена:</strong> ${fridge.price}</p>

`;

fridgeDetails.style.display = "block";

document.getElementById("toggleDetails").textContent = "Скрыть характеристики";

} else {

// Меняем изображение на "закрытое"

fridgeImage.src = "closed\_fridge.png";

fridgeImage.alt = "Закрытый холодильник";

// Скрываем характеристики

fridgeDetails.style.display = "none";

document.getElementById("toggleDetails").textContent = "Показать характеристики";

}

}

// Функция для удаления холодильника

function deleteFridge() {

fridge = null; // Удаляем объект холодильника

fridgeOpen = false; // Сбрасываем состояние холодильника

// Скрываем контейнер холодильника и кнопки управления

document.getElementById("fridgeContainer").style.display = "none";

document.getElementById("toggleDetails").style.display = "none";

document.getElementById("deleteFridge").style.display = "none";

// Показать кнопку "Создать холодильник"

document.getElementById("createFridge").style.display = "inline-block";

alert('Объект "Холодильник" удален!');

}

// Назначаем функции на кнопки

document.getElementById("createFridge").addEventListener("click", createFridge);

document.getElementById("toggleDetails").addEventListener("click", toggleFridgeDetails);

document.getElementById("deleteFridge").addEventListener("click", deleteFridge);