АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ

Факультет: Информационные технологии и управление
Кафедра: Компьютерная инженерия

Предмет: «Веб-системы и технологии»

Курсовая работа

Тема «Разработка онлайн информационной системы для продуктовых магазинов»

Группа: 680.22

Курс: 3

Специальность: 050616 Информационные технологии

Студент: Агаев Нурлан

Руководитель: **асс. Халилов М. Э.**

Зав.кафедрой: **доц.Рагимова Н.А.**

Баку – 2025

АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ

АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ

ЗАДАНИЕ ПО КУРСОВОЙ РАБОТЕ

ФАКУЛЬТЕТ: Информационные технологии и управление
КАФЕДРА: Компьютерная инженерия

Группа 680.22 **курс** III

Специальность **050616** Информационные технологии

Студент Агаев Нурлан

Зачетная книжка _____

Руководитель курсовой работы **асс. Халилов М. Э.**
Срок выдачи _____

Дата сдачи _____

Тема курсовой работы «Разработка онлайн информационной системы для продуктовых магазинов»

Отзыв руководителя курсовой:

_____

_____

_____

Подпись студента _____

Подпись рук.курсовой _____

Подпись зав.кафедрой _____

Дата защиты курсовой _____ **Оценка** _____

Пред.комиссии _____(_____)

Члены комиссии: 1._____(_____)

2._____(_____)

3._____(_____)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Веб-системы и технологии относятся к различным инструментам, языкам и платформам, которые используются для разработки, развертывания и обслуживания веб-приложений. Эти технологии развивались на протяжении многих лет, чтобы не отставать от растущего спроса на веб-сервисы, что привело к появлению разнообразных языков программирования, фреймворков, библиотек и баз данных. Когда появились Web-технологии, пользоваться компьютерной техникой начали абсолютно новые категории граждан. Список социальных категорий населения, подсоединяющихся к сети интернет с целью поиска информации во всемирной паутине, постоянно увеличивается за счёт людей, которые не считают себя специалистами в сфере информационных технологий. Web-технология в корне изменила понятия об информационной обработке, как и работе с компьютерной техникой тоже.

Оказывается стандартные характеристики, описывающие прогресс в сфере вычислительного оборудования, а именно, производительность, пропускная способность, объём памяти, практически не включали в свой состав главное системное «узкое место», именуемое пользовательским интерфейсом. Устаревшая техника взаимоотношений пользователя с информационной системой тормозила продвижение передовых технологий и нивелировала преимущества от их внедрения. И лишь когда интерфейс пользователя, обеспечивающий взаимодействие человека и компьютера, разработчики довели до простого и естественного понимания обычными людьми, произошёл невообразимый взрывной рост интереса к достоинствам и возможностям компьютерного оборудования.

Web-технологиями является весь набор средств, позволяющих организовать WWW (World Wide Web), то есть всемирную паутину. Так как каждый сеанс является взаимодействием двух сторон, а именно, сервера и клиента, то и Web-технологии делятся на следующие группы:

1. Технологии серверной стороны (server-side).
2. Технологии клиентской стороны (client-side).

Технологии клиентской стороны включают в свой состав весь набор технологий по созданию веб-страниц (HTML, JavaScript, DHTML и другие), а технологии серверной стороны состоят из технологий доступа к информационным базам данных в сети интернет (CGI, PHP).

Серверные программы обеспечивают предоставление тех или иных ресурсов клиентским программам. Клиенты, когда им требуется какой-либо файл или просто какая-то информация от сервера, вырабатывают специальный запрос клиента и отправляют его серверу. Серверная программа выполняет обработку запроса и отправляет ответ сервера, который содержит запрошенную информацию или же извещение об ошибке, в случае недоступности требуемых данных. Данная компьютерная организация, или по другому, принципы формирования вычислительных систем или сетей, именуется архитектурой «клиент-сервер» или двухзвенной организацией. Как раз на базе двухзвенной архитектуры работают практически все интернет - сервисы, включая и WWW.

Данная курсовая работа рассматривает пример создания информационной системы для продуктовых магазинов.

Разработка онлайн-информационной системы будет включать в себя следующие этапы:

1. Сбор требований: Первым шагом будет сбор требований от клиента, таких как продукты, которые он хочет продавать, целевая аудитория, функции, которые он хочет видеть на веб-сайте.

2. Дизайн: Следующим шагом будет разработка дизайна веб-сайта, включая создание каркасов и макетов, выбор цветовых схем и шрифтов, а также определение общего макета и пользовательского интерфейса.

3. Разработка: Это предполагает создание веб-сайта с использованием HTML, CSS, JavaScript и других веб-технологий.

4. Тестирование: После завершения разработки веб-сайт необходимо будет протестировать, чтобы убедиться, что он должным образом функционирует в различных браузерах и устройствах.

WEB-сайт Интернет-магазина – это система, ориентированная большей частью на покупателя. Он должен предоставлять сервисы просмотра товара, удобного поиска товара, легкость и удобство при заказе товара.

И так, целью моей курсовой работы является создание сайта Интернет-магазина, обеспечивающего удобство, безопасность и интуитивную понятность интерфейса пользователю.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Web-программирование

Web-программирование (Web-разработка) - это бурно развивающийся раздел программирования, ориентированный на разработку динамических Интернет-приложений. Языки Web-программирования делятся на две группы: клиентские и серверные. Клиентские языки обрабатываются на стороне пользователя (в основном в браузере). Соответственно, обработка скрипта зависит от браузера пользователя, и пользователь имеет полномочия настроить свой браузер так, чтобы тот вообще игнорировал скрипты. При этом если браузер старый, он может не поддерживать тот или иной язык или версию языка, на которую опирался разработчик. С современными браузерами таких проблем возникать не должно, к тому же языки программирования не так уж часто кардинально обновляются (раз в несколько лет) и лучшие из них давно известны. Код клиентского скрипта может посмотреть каждый, выбрав в меню своего браузера «Исходный код страницы». Преимущество клиентского языка заключается в том, что обработка скриптов на таком языке может выполняться без отправки документа на сервер. Программа сразу проверит правильное заполнение формы перед отправкой, и, если необходимо, выведет ошибку. Отсюда же вытекает и то ограничение, что с помощью клиентского языка программирования ничто не может быть записано на сервер. Самым распространенным из клиентских языков признан JavaScript, разработчиками которого является компания Netscape совместно с компанией Sun Microsystems. Еще один популярный язык - это VBScript. Помимо этого в последнее время набрали популярность такие технологии, как AJAX, Adobe Flash, Microsoft Silverlight и др. Серверные языки программирования открывают перед программистом большие просторы для деятельности. Когда пользователь делает запрос на какую-либо страницу (переходит на нее по ссылке, или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере (то есть выполняются все программы, связанные со страницей) и только потом возвращается к посетителю по сети в виде файла. Этот файл может иметь расширения: HTML, PHP, ASP, Perl, SSI, XML, DHTML, XHTML.

1.2 HTML и XHTML

HyperTextMarkupLanguage (HTML) - это язык разметки документов во Всемирной паутине, принятый за стандартный. Большая доля всех Web-страниц в Интернете создана при помощи языка HTML (или XHTML), поэтому мы рассмотрим его подробно.

Язык HTML позволяет форматировать текст и другие элемента Web-страницы:

Цвет, жирность, стиль, название шрифта для текста.

Позволяет выделять фрагменты и символы например: ударение в слове, заголовок страницы или абзаца, сам абзац, пункт списка.

Гипертекстовые ссылки, позволяют переходить между документами и между фрагментами одного документа.

Формы для введения данных, как правило, данные из форм обрабатываются с помощью скриптов на языках программирования, ориентированных на Web,например PHP.

Отображение мультимедийных файлов, их может отображать сам браузер - изображения, аудиофайлы, или внешние приложения, взаимодействующие с браузером, например Flash-ролики, Java-апплеты.

HTML - язык разметки документов основанный на тэгах. Документ на языке HTML представляет собой набор элементов, при этом начало и конец каждого элемента обозначается служебными символами - тегами. Все тэги HTML начинаются с «<« (левой угловой скобки) и заканчиваются символом «>« (правой угловой скобки). Завершающий тег выглядит также, как начальный, и отличается от него прямым слэшем перед текстом внутри угловых скобок.

<HTML></HTML>

HTML регистронезависимый язык, теги могут быть написаны как строчными, так и заглавными буквами (в отличие от XHTML). Теги могут быть вложенными друг в друга.

<**HTML>**

<HEAD>

<TITLE>

Заголовок страницы

</TITLE>

</HEAD>

</HTML>

Тэги могут быть пустыми, то есть не содержать текста или других вложенных конструкций (например, <**br>** который переводит строку). Закрывающий тег в таком случае не указывается.

Также, элементы разметки могут иметь атрибуты, задающие их свойства (например, размер шрифта, цвет, расположение). Атрибуты задаются в начале тега.

<ahref=«http://www.yandex.ru**»>**Пример элемента с атрибутом **href.**

Теги можно разделить на следующие группы.

Гиперссылки

<AHREF=«filename**»target=«_self»>**текст ссылки**</A>**

где filename - имя файла (может быть и локальным) или адрес страницы вInternet, на который нужно совершить переход.

текст ссылки - текст гипертекстовой ссылки, который будет отображаться в браузере, как правило, выделяется подчеркиванием.

target - задает окно или фрейм, в котором будет открыт документ, при переходе по ссылке. Он может принимать значения:

_top - документ откроется в текущем окне

_blank - документ откроется в новом окне

_self - документ откроется в текущем фрейме

_parent - документ откроется в родительском фрейме

По умолчанию принимает значение _self.

Текстовые ссылки.

<H1></H1>,**<H2></H2>**, … ,**<H6></H6>** - заголовки 1-6 уровней. Применяются для выделения частей выводимого текста (заголовок 1 - будет выведен очень большим, 6 - будет размером сопоставимым с обычным текстом).

<P> - обозначает начало нового абзаца. Закрывающий тег**</P>**, не является необязательным.

**
** - переход на новую строку. Закрывающий тег**</br>**, отсутствует.

<HR> - горизонтальная линия.

<BLOCKQUOTE></BLOCKQUOTE> - цитата. Выделение заданного текста как цитаты.

<PRE></PRE> - режим предпросмотра. При этом текст заключается в рамку и выводиться не форматированным (то есть все теги, кроме </PRE>, игнорируются, но при этом переводы строки ставятся там, где они присутствуют в исходном документе).

<DIV></DIV> - блок текста (как правило, применяется для использования каскадных стилей CSS).

<SPAN></SPAN> - строка (как правило, применяется для использования каскадных стилей CSS).

Теги форматирования текста

<EM></EM> - выделение символа, на который падает ударение (обычно отображается курсивом).

<STRONG></STRONG> - выделение символа, на который падает усиленное ударение (обычно отображается жирным текстом).

<I></I> - выделение текста курсивом.

<B></B> - выделение текста жирным шрифтом.

<U></U> - подчёркивание текста

<S></S> - зачёркивание текста.

<STRIKE></STRIKE> - то же самое, что <S> … </S>

<BIG></BIG> - увеличение шрифта.

<SMALL></SMALL> - уменьшение шрифта.

<BLINK></BLINK> - мигающий текст.

<MARQUEE></MARQUEE> - сдвигающийся по экрану текст.

**** - вывод текста под строкой. Например, **H₂ O**тобразиться в виде текста H2O.

**** - вывод текста над строкой. Например, **E=mc²**отобразиться в виде текста E=mc2.

<FONT атрибуты></FONT> - задание атрибутов у используемого шрифта. Атрибуты могут быть следующими:

COLOR=color - указание цвета. Цвет может быть указан шестнадцатеричным числом в формате #rrggbb (первые 2 шестнадцатеричные цифры указывают интенсивность красного, следующие 2 - зелёного, последние 2 - синего) или названием самого цвета.

FACE= указываем имя шрифта.

SIZE= позволяет изменить размер шрифта. Размеры могут быть от 1 до 7, по умолчанию размер 3.

SIZE=+ размер или SIZE=-размер - размер больше или меньше стандартного. Например,SIZE=+2 указывает размер на 2 больше стандарта, то есть размер 5.

Списки.

Данная конструкция

<UL>

<LI> первый элемент списка </LI>

<LI> второй элемент списка </LI>

<LI> третий элемент списка </LI>

</UL>

создаёт список вида:

- первый элемент;
- второй элемент;
- третий элемент.

Также стоит отметить, что тегов есть параметры, позволяющие менять вид списка.

Объекты.

EMBED - вставка объектов различных типов

APPLET - вставка Java-апплетов

SCRIPT - вставка различных скриптов, например JavaScript

Изображения.

IMG - тег для вставки изображения. Это не закрывающийся тег.

SRC - имя локального файла или путь к нему в виде URL

ALT - текст картинки (отобразится, в виде текста, если не удалось отобразить картинку)

TITLE - подсказка (показывается при попадании курсора в область картинки)

WIDTH, HEIGHT - размеры изображения (выводимое изображение будет масштабировано до указанных размеров)

ALIGN - обтекание текста

Таблицы.

TABLE - тег создание таблицы. Тег имеет следующие параметры:

BORDER - задает толщину границу таблицы

CELLSPACING - задает расстояние от ячейки до ячейки

CAPTION - задает заголовок таблицы (необязательный тег)

TR - добавление строки в таблицы

TH - задает заголовок столбца (необязательный тег)

TD - добавление ячейки таблицы

WIDTH, HEIGHT - размеры таблицы

Формы. Формы ввода данных могут быть самыми разнообразными. Поэтому рассмотрим только основные теги:

FORM - тег для создания формы

INPUT - добавление элемента ввода

TEXTAREA - добавление текстового поля

SELECT - добавление списка (как правило, это выпадающее меню)

OPTION - пункт списка

Символы. Некоторые символы не могут быть выведены напрямую. Для их вывода требуется использовать их определения, например, символ апмерсанд& в коде HTML будет иметь вид **&amp**, символ меньше < будет иметь вид **&lt**, символ больше > будет **&gt**. Это ограничение введено, так как эти символы уже используются в языке HTML как служебные.

Любая HTML-страница должна иметь обозначение начала и конца документа обрамленные тегами**<html>** и **</html>** соответственно. Внутри них должны находиться теги заголовка **<head>** и **</head>**, итеги, обозначающие тело документа **<body>** и **</body>**. А внутри них могут быть произвольные комбинации из групп тегов описанных ранее.

Также рассмотрим ExtensibleHypertextMarkupLanguage (XHTML) это расширяемый язык разметки гипертекста. Стоит отметить, что язык XHTML это ни описание самого языка, а список отличий XHTML от HTML. Рассмотрим основные отличия HTML и XHTML. В XHTML все используемые теги должны иметь закрывающий тег. Теги, не имеющие закрывающего тега должны оканчиваться символом /. Например тег **
**, должен иметь закрывающий его тег **
. В XHTML допускается писать теги и их атрибуты только строчными буквами. В XHTML очень строгая проверка синтаксиса не допускается использовать< и &, даже в URL, вместо них должны быть **&lt и **&amp**. Браузеры, обнаружив ошибку синтаксиса в XHTML, должны прекратить его обработку и вывести ошибку на экран. В стандарте HTML браузер должен попытаться отобразить запрашиваемый документ. Стоит отметить, что XHTML расширяемый язык - за счет указания типа документа и возможности использовать свои теги.

Как мы видим, язык разметки HTML предоставляет широкие возможности для отображения информации, для этого в нем содержится большое количество тегов для различного форматирования выводимой информации. Язык XHTML очень похож на HTML, но более строгий, грамматические правила в XHTML менее сложные, и как следствие при созданий Web-страниц будет меньше ошибок.

1.3 Dynamic HTML

Dynamic HTML или DHTML - так принято называть связку языка HTML, каскадных таблиц стилей, скриптового языка и объектной модели документов. Скриптовым языком может выступать JavaScript или VisualBasic, но именно первый язык получил большую популярность и сегодня

используется повсеместно. При помощи DHTML можно создавать интерактивные Web страницы, он позволяет легко и гибко обрабатывать данные запроса и формировать динамический ответ. DHTML может быть использован для реализации интерфейса Drag'n'Drop. На его основе создаются игры и другие интерактивные сервисы. Стоит отметить, что для реализации принципов DHTML достаточно лишь браузера, который будет обрабатывать содержимое страницы. То есть, нет необходимости, например, в обращениях к базе данных.

1.3.1 Каскадные таблицы стилей

Рассмотрим подробнее каскадные таблицы стилей - CSS (CascadingStyleSheets). Это стандарт позволяющий задавать описание внешнего вида некоторых элементов страницы на HTML. CSS используется при создании Web-страниц для определения шрифта, цвета, расположения и прочих атрибутов, используемых в документе, элементов.

Основная цель использования CSS разграничить само содержимое страницы, созданное на языке HTML и описание оформления, которое написано на CSS. Такое разделение упрощает создание и изменение документа, документом более легко управлять, а также снижается избыточность в исходном коде. Также, при помощи CSS можно легко представить один и тот же документ, но с разным форматированием или использовать разные методы вывода, такие как отображение на экране, печать документа, чтение голосом и тому подобных.

Описание стиля на CSS при открытии страницы может быть взято из разных мест: оно может быть встроенным стилем - это блоки CSS внутри страницы на языке HTML.

```
<style type=«text/css»>
body {
color:green;
}
</style>
```

Из отдельного файла .css, в котором описывается таблица стиля, на этот файл делается ссылка внутри страницы:

<link rel=«stylesheet» **type=**«text/css» **href=**«style.css» **/>**

Это может быть Inline-стиль, при этом в HTML документе информация о стиле элемента указывается как атрибут style.

<p style=«font-size: 21px; color: green;»**>**Выводимый текст**</p>**

Также в любом браузере есть свой стандартный стиль, используемый по умолчанию.

Таблица стилей это набор описаний форматов. Каждое правило стиля, в свою очередь, имеет один или несколько селекторов, которые разделяются запятыми. В любом стиле также должен присутствовать блок определений, который обозначается фигурными скобками { }, и включает в себя набор свойств и их значений.

селектор, селектор {
свойство1: значение;
свойство2: значение;
свойство3: значение;
}

В стандарте CSS присутствуют приоритеты, согласно которым применяются правила стилей, например, если для элемента подходят свойства сразу нескольких правил. Это и называется каскадом. Стоит обратить внимание и на порядок расположения применяемых свойств - у свойства, указанного позже, будет более высокий приоритет.

До создания CSS оформление стиля Web-страницы производилось непосредственно внутри самого содержимого документа. Однако с использованием CSS появилась возможность принципиально разделить содержания и описание стиля документа. Благодаря этому стало возможным лёгкое применение единого стиля при создании похожих документов, а также это оформления можно изменить легко и быстро. Появилась возможность создать несколько вариантов дизайна страницы для разных устройств, на котором ее можно просмотреть. Например, на мониторе дизайн страницы

будет иметь большую ширину, при отправке документа на печать буду отбрасываться такие элементы как меню, а на КПК или сотовом будет масштабирование уменьшающее искажение текста страницы. За счет того, что таблицы стилей хранятся в отдельном файле CSS, уменьшается время загрузки страницы. В этом случае браузер получает только описание структуры документа и данные, находящиеся на Web-странице, а стиль оформления этих данных загружается браузером при первом обращении и потом берется из КЭШа. Последующее изменения дизайна сайта так же значительно упрощается. Нет необходимости изменять каждую страницу, лишь нужно внести изменения в CSS-файл. Но у CSS есть также и недостатки. Различное отображение вёрстки в различных браузерах, которые по-разному отображают одни и те же данные CSS.

Для работы с CSS есть большое количество различных программных продуктов, но они не получили большого распространения. Как правило, объем CSS, кода по соотношению с HTML, не такой большой. И почти все HTML-редакторы имеют поддержку синтаксиса CSS. Из отдельных решений можно выделить TopStyle, данный программный продукт предназначен для создания стилей CSS. Очень полезным в нем является модуль Stylechecker, с помощью него можно не только проверить правильность синтаксиса таблицы стилей, но и проверить формат представления данных в зависимости от браузера. Таким образом, при его помощи можно создавать стили, которые будут одинаково отображаться в разных браузерах. Также существуют визуальные CSS-редакторы, но из них сложно выделить какой-то наиболее достойный.

1.3.2 JavaScript

Несмотря на то, что JavaScript не является стандартом W3C, мы рассмотрим его в данном контексте, и как часть DHTML.

JavaScript - это объектно-ориентированный скриптовый язык программирования. JavaScript встраиваемый язык, используемый для доступа к объектам в приложениях. Он нашел очень широкое применение при

создании Web-страниц. JavaScript очень похож на язык Си, но все же имеет кардинальные отличия: Структуру объектов и тип объектов (например переменных, функций) можно определить в процессе выполнения кода. В JavaScript действует автоматическое приведение типов данных. Также автоматической является контроль памяти и защита от утечек. Функции в языке программирования JavaScript могут быть без имени. Все описания переменных, функции и других элементов регистрозависимы, названия переменных могут содержать буквы, символ подчёркивания, символ доллара и арабские цифры. При этом названия переменных не должны начинаться с цифры. Стоит отметить, что JavaScript не поддерживает области видимости, не умеет работать с файловой системой и потоками ввода/вывода, не имеет сетевых интерфейсов. Это сделано для обеспечения безопасности, так как скрипт на языке JavaScript легко запустить на любом компьютере, для этого будет достаточным открыть на нем Web-страницу, содержащую скрипт. Также для повышения безопасности скрипт не может получить доступ к свойствам другой страницы.

Использовать JavaScript на странице, можно при помощи тегов**<script></script>.** Например:

<**script type=**«text/javascript»**>**

alert ('Hello, World!'**);**

</script>

Также скрипт может быть сохранен в отдельном файле.

<scripttype=«text/javascript»src=http://Путь_к_файлу_со_скриптом**></scr ipt>**

Язык JavaScriptдает создателю страницы очень широкие возможности для выполнения скриптов и как следствие качественно новые горизонты для создания динамических элементов на Web-страницах. JavaScript позволяет автоматически заполнять данные в формах, изменять форматирование и стиль страницы. Скрывать ненужное содержимое и отображать нужное содержимое, менять поведение клиентской части Web-приложений, изменять элементы

управления на странице и так далее. При этом язык довольно простой для понимания, и большая часть функции выполняется автоматически.

Выделить какой-то редактор для JavaScript сложно, но отметим что в среде Eclipse, есть поддержка JavaScript при помощи подключаемого модуля, при этом становиться доступен весь функционал среды Eclipse. А также практически все современные браузеры имеют встроенные средства для отладки JavaScript-скриптов.

1.4 Bootstrap

Bootstrap — это открытый и бесплатный HTML-, CSS- и JS-фреймворк, который используют веб-разработчики для быстрой верстки адаптивных дизайнов сайтов и веб-приложений. Включает в себя CSS- и HTML-шаблоны оформления для веб-форм, меток, типографики, кнопок, блоков навигации и других компонентов веб-интерфейса.

Bootstrap используется, когда:

- у сайта много страниц;
- страницы собраны из простых базовых элементов — кнопок или таблиц;
- не будет глобального редизайна;
- шаблонность страниц окупается скоростью внедрения.

По сути, Bootstrap — это набор файлов. После их подключения к странице для верстки станет доступно большое количество готовых компонентов и классов. Они позволяют быстро и качественно создавать адаптивный дизайн сайта.

Классы в Bootstrap делятся на 3 большие группы:

- для создания сетки — адаптивного макета страницы;
- для стилизации контента — текста, изображений, кода, таблиц и прочей информации;

- служебные — для решения популярных вспомогательных задач, таких как отображение и скрытие элементов, выравнивание текста на странице, настройка цвета фона, отступов и пр.

Кроме классов, в Bootstrap есть компоненты (готовые объекты интерфейса). Это хлебные крошки (путь от начального элемента до уровня иерархии, который сейчас просматривает пользователь), кнопки, выпадающие списки и подсказки, модальные окна, формы, навигационные меню, всплывающие панели и пр.

Чтобы начать работать с Bootstrap, нужен текстовый редактор для работы с кодом (Visual Studio Code, Atom, Sublime Text и т.п.) или IDE — интегрированная среда разработки, а также браузер, в котором можно отслеживать изменения. Он должен быть обновлен до версии, которую поддерживает последний вариант Bootstrap.

Существует несколько способов установки Bootstrap. Получить актуальную версию фреймворка можно на официальном сайте.

1. Подключение скомпилированных файлов через BootstrapCDN — общедоступную сеть доставки контента. Достаточно создать шаблон HTML и разместить внутри него ссылку на фреймворк.
2. Скачивание скомпилированных файлов CSS и JS с их подключением к проекту через ссылки.
3. Скачивание исходных файлов. Для работы потребуются компилятор CSS и автопрефиксатор.
4. Установка исходных файлов с помощью менеджеров пакетов yarn, npm, RubyGems, NuGet, Composer.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Описание предметной области

Предметная область - область теории, рассматривасмая в пределах отдельного рассуждения, научной теории. Также под предметной областью принято понимать часть реального мира, подлежащего изучению для организации управления и ,в конечном счете, автоматизации.

Предметной областью данной курсовой работы, является онлайн информационная система для продуктовых магазинов.

Архитектура сайта – это систематизация информации и навигации по ней с целью помочь потенциальным заказчикам находить нужные им данные. Хорошо продуманная архитектура сайта обеспечивает меньшее время на поиск запрашиваемой информации и гарантирует популярность среди пользователей благодаря простоте использования сайта.

При разработке архитектуры сайта необходимо учитывать то, как представление информации повиляет на продвижение товаров и услуг на интернет-рынке. В процессе создания структуры нового сайта, либо оптимизации структуры уже существующего, необходимо представлять информацию таким образом, чтобы повысить посещаемость сайта и привлечь внимание потенциальных покупателей к наиболее важным его разделам, исходя их предлагаемых товаров и услуг.

Важным элементом в разработке интернет-магазина является интерфейс, так как с помощью него будет происходить взаимодействие с клиентом.

Благодаря грамотному расположению блоков на странице клиент сможет найти интересующую информацию о товарах или же предоставляемых услугах компании.

Для разработки были использованы следующие технологии:

- HTML – этот язык является базовым в области технологий создания сайтов, так как относительно легок в освоении.
- CSS – формальный язык, преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML.

2.2 Создание окружения для проекта

В качестве редактора кода будет использован Visual Studio Code. Создаем папку в нем, которую назовем “ KURSOVAYA”, в нем и будут все файлы проекта.

Сайт будет состоять из одной страниц. Создадим для нее файл index.html

Окружение готово и теперь мы можем приступить к самой разработке (рис. 1)

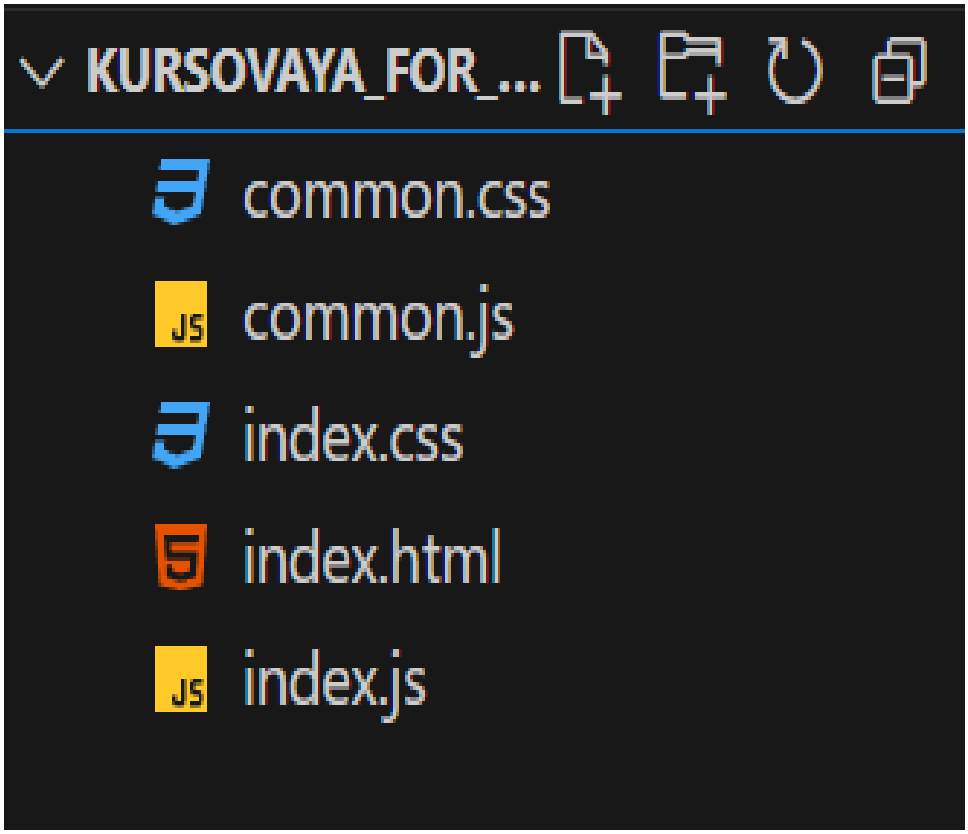


Рис. 1. Итоговый вид окружения

2.3 Верстка HTML страницы

Открываем ранее созданный “index.html” и пишем стандартный HTML5 код (рис. 2).

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Фермерский рынок "ЭкоФерма" - Свежие продукты от фермеров</title>
```

Рис. 2

Элемент <!DOCTYPE> предназначен для указания типа текущего документа — DTD (document type definition, описание типа документа). Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях.

Тег <html> является контейнером, который заключает в себе все содержимое веб-страницы, включая теги <head> и <body>. Открывающий и закрывающий теги <html> в документе необязательны, но хороший стиль диктует непременное их использование.

Элемент <head> содержит машиночитаемую информацию (metadata) о документе, например его заголовок, скрипты и страницы стилей.

Элемент <body> представляет собой контент (содержимое) документа HTML. В документе может быть только один элемент <body>.

Затем мы подключаем наши файлы через <link> в элементе <head> (рис. 3)

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Фермерский рынок "ЭкоФерма" - Свежие продукты от фермеров</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="./common.css">
    <link rel="stylesheet" href="./index.css">
</head>
<body>
```

Рис. 3

Затем прописываем в <head> : <meta name="viewport" content="width=device-width, initial-scale=1> . Этот код задает параметры масштабирования страницы для устройств с разными размерами экранов. Атрибут content определяет метаданные, которые могут быть использованы браузером для корректного отображения содержимого в окне браузера на разных устройствах.

- Атрибут width=device-width задает ширину страницы равной ширине экрана устройства.
- Атрибут initial-scale=1 устанавливает начальный масштаб страницы, который будет использоваться при первом загрузке страницы.

Теперь в <body> будем прописывать контент.

Создаем навбар сайта.
Для начала создаем <nav> в котором будет весь контент навбара.

Затем создаем через <a> лого:

```
<a href="index.html" class="logo">
                <i class="fas fa-leaf"></i>
                <span>ЭкоФерма</span>
    </a>
```

HTML-элемент <a> определяет гиперссылку для перехода на определённое место на странице или на другую страницу в Интернете. Также он может быть использован (в устаревшем варианте) для создания якоря — это место назначения для гиперссылок внутри страницы: так ссылки не ограничены только в перемещении между страницами.

Затем создаем меню через <ul> , <li>:

```
<nav class="nav" id="mainNav">
      <ul class="nav-list">
          <li class="nav-item">
              <a href="index.html" class="nav-link active">Главная</a>
          </li>
          <li class="nav-item">
              <a href="#" class="nav-link">О нас</a>
          </li>
         * <li class="nav-item">
              <a href="#" class="nav-link">Каталог</a>
          </li>
          <li class="nav-item">
              <a href="#" class="nav-link">Контакты</a>
          </li>
      </ul>
  </nav>
```

HTML-элемент <ul> используется для неупорядоченного списка - в частности для маркированного списка.

HTML-элемент <li> используется для создания элементов списка. Он также должен находиться в родительском элементе: упорядоченном списке (<ol>), неупорядоченном списке (<ul>), или меню (<menu>).

Затем создаем и помещаем туда иконки:

```
<div class="container header-container">
          <a href="index.html" class="logo">
              <i class="fas fa-leaf"></i>
              <span>ЭкоФерма</span>
          </a>

          <button class="mobile-menu-btn" id="mobileMenuBtn">
              <i class="fas fa-bars"></i>
          </button>

          <nav class="nav" id="mainNav">
```

```
            <ul class="nav-list">
                <li class="nav-item">
                    <a href="index.html" class="nav-link active">Главная</a>
                </li>
                <li class="nav-item">
                    <a href="#" class="nav-link">О нас</a>
                </li>
               * <li class="nav-item">
                    <a href="#" class="nav-link">Каталог</a>
                </li>
                <li class="nav-item">
                    <a href="#" class="nav-link">Контакты</a>
                </li>
            </ul>
        </nav>

        <div class="header-buttons">
            <div class="header-icon">
                <i class="fas fa-search"></i>
            </div>
            <div class="header-icon">
                <i class="fas fa-user"></i>
            </div>
            <div class="header-icon">
                <i class="fas fa-shopping-cart"></i>
                <span class="cart-count">3</span>
            </div>
        </div>
    </div>
```

После задания css стилей, которое будут описаны в части “2.3 Использование стилей CSS”, навбар примет такой вид. (Рис. 4)

(Рис. 4)

```
<main>
    <!-- Герой секция -->
    <section class="hero">
        <div class="container hero-content">
            <h1>Свежие продукты от фермеров</h1>
            <p>Мы предлагаем широкий ассортимент экологически чистых продуктов
напрямую от местных фермеров.</p>
            <div class="hero-buttons">
```

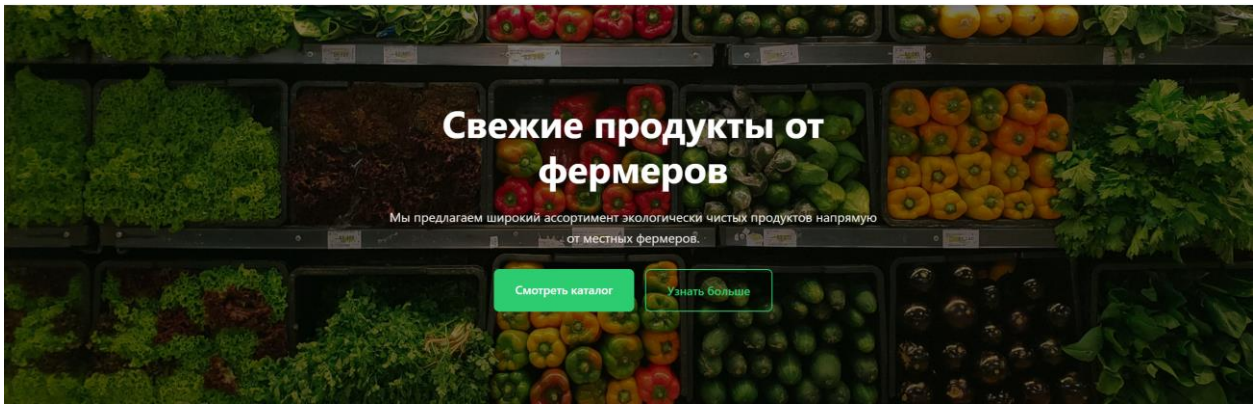
```
                <a href="#" class="btn btn-lg" id="viewCatalogBtn">Смотреть
каталог</a>
                <a href="about.html" class="btn btn-lg btn-outline">Узнать
больше</a>
            </div>
        </div>
    </section>
</main>
```

Назначение блока <main>
Элемент <main> обозначает основное содержимое страницы. В рамках семантической структуры HTML он используется для размещения ключевого контента, уникального для данной страницы. В данной работе он содержит "герой-секцию" — важную зону первого экрана сайта.
<section class="hero">
Блок <section> с классом hero представляет собой **герой-секцию (hero section)**. Это визуально заметный блок, располагающийся в верхней части страницы. Его цель — быстро донести до пользователя основное сообщение сайта. В данном случае — реклама фермерских продуктов.
** **
Элемент с классом container hero-content служит для **ограничения ширины контента** и **центровки содержимого** на странице. Класс container, как правило, используется для обеспечения отступов и выравнивания, а hero-content — для индивидуальной стилизации конкретной секции.

Блок кнопок .hero-buttons
Этот блок содержит две ссылки-кнопки, выполненные с помощью тегов <a>, которые стилизованы с помощью CSS-классов:

```
<a href="#" class="btn btn-lg" id="viewCatalogBtn">Смотреть каталог</a>
<a href="about.html" class="btn btn-lg btn-outline">Узнать больше</a>
```

btn — базовый класс кнопки (например, задает фон, отступы, скругление).
btn-lg — модификатор, увеличивающий размер кнопки (large).
btn-outline — вариант кнопки с обводкой вместо заливки.
id="viewCatalogBtn" — идентификатор для возможности **добавления JavaScript-функционала**, например, при клике можно плавно пролистывать к каталогу.
В итоге мы получим следующее:



Следующая часть секция категория:

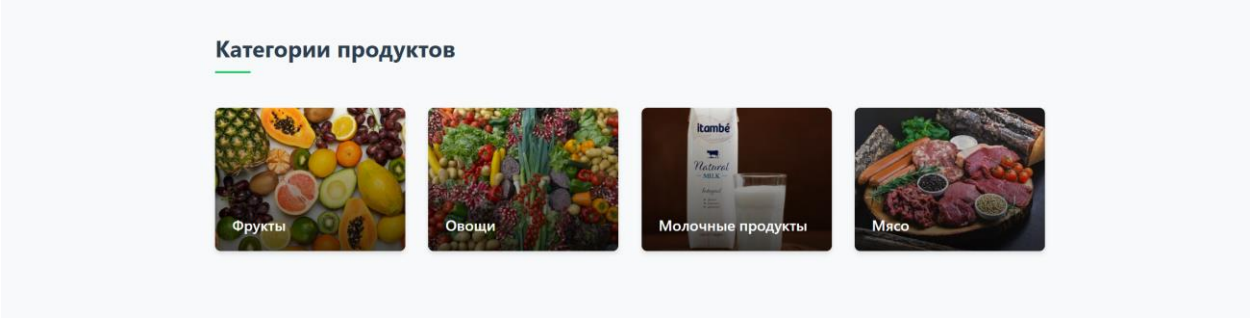
```
<!-- Категории -->
<section class="categories section">
    <div class="container">
        <h2 class="section-title text-center">Категории продуктов</h2>
        <div class="category-grid">
            <div class="category-card">
                <img src="https://images.unsplash.com/photo-1610832958506-aa56368176cf?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG9
                <div class="category-overlay">
                    <h3 class="category-name">Фрукты</h3>
                </div>
            </div>
            <div class="category-card">
                <img src="https://images.unsplash.com/photo-1566385101042-1a0aa0c1268c?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG9
                <div class="category-overlay">
                    <h3 class="category-name">Овощи</h3>
                </div>
            </div>
            <div class="category-card">
                <img src="https://images.unsplash.com/photo-1563636619-e9143da7973b?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by
                <div class="category-overlay">
                    <h3 class="category-name">Молочные продукты</h3>
                </div>
            </div>
            <div class="category-card">
                <img src="https://images.unsplash.com/photo-1607623814075-e51df1bdc82f?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG9
                <div class="category-overlay">
                    <h3 class="category-name">Мясо</h3>
                </div>
            </div>
        </div>
    </div>
</section>
```

Объяснение:

1. **Элемент <section>**:
   - Используется для выделения части контента. Здесь — секция категорий продуктов.
2. **Класс categories section**:
   - categories — основной класс для секции, вероятно, для стилизации через CSS.
   - section — добавляет семантический смысл, указывая, что это отдельная логическая часть страницы.
3. **Элемент <h2>**:
   - Заголовок, который объявляет, что следующая часть содержит **категории продуктов**.
4. **Класс category-grid**:
   - Это контейнер для карточек категорий. Он будет использовать **CSS Grid** или **Flexbox** для выравнивания карточек в сетке.
5. **Каждая карточка категории (category-card)**:
   - Включает изображение (например, фрукты, овощи) с классом category-img.

- Картинка обернута в контейнер с классом category-overlay, который может использоваться для стилизации, например, затемнения фона.
- Название категории (category-name) отображается поверх изображения, что позволяет быстро идентифицировать категорию.

В итоге мы получаем следующее:



Следующая секция **Популярные продукты:**

Напишем самое основное:

Фильтры продуктов:

```
<div class="product-filter">
    <button class="filter-btn active" data-category="all">Все</button>
    <button class="filter-btn" data-category="fruits">Фрукты</button>
    <button class="filter-btn" data-category="vegetables">Овощи</button>
    <button class="filter-btn" data-category="dairy">Молочные
продукты</button>
    <button class="filter-btn" data-category="meat">Мясо</button>
</div>
```

Эта часть содержит кнопки для фильтрации продуктов по категориям: Все, Фрукты, Овощи, Молочные продукты, Мясо.

Кнопки используют класс filter-btn, а активная кнопка имеет дополнительный класс active.

Сетка с продуктами:

```
<div class="product-grid">
    <div class="product-card" data-category="fruits">
        <!-- Продукт "Яблоки" -->
    </div>
    <div class="product-card" data-category="vegetables">
        <!-- Продукт "Морковь" -->
    </div>
    <div class="product-card" data-category="dairy">
        <!-- Продукт "Молоко" -->
    </div>
    <div class="product-card" data-category="meat">
        <!-- Продукт "Говядина" -->
    </div>
</div>
```

Каждая карточка продукта обернута в с классом product-card. Класс data-category указывает категорию продукта, чтобы фильтрация работала корректно.
В каждой карточке отображается изображение, информация о продукте и действия с продуктом.

Внутри каждой карточки:

- **Изображение продукта** (<img>): картинка продукта с классом product-img.
- **Действия с продуктом**: кнопки для добавления в избранное и просмотра.

```
<div class="product-actions">
    <div class="product-action-btn">
        <i class="far fa-heart"></i>
    </div>
    <div class="product-action-btn">
        <i class="fas fa-eye"></i>
    </div>
</div>
```

Рейтинг продукта:

- Отображаются звездочки рейтинга и количество отзывов.

```
<div class="product-rating">
    <div class="rating-stars">
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star"></i>
        <i class="fas fa-star-half-alt"></i>
    </div>
    <span class="rating-count">(24)</span>
</div>
```

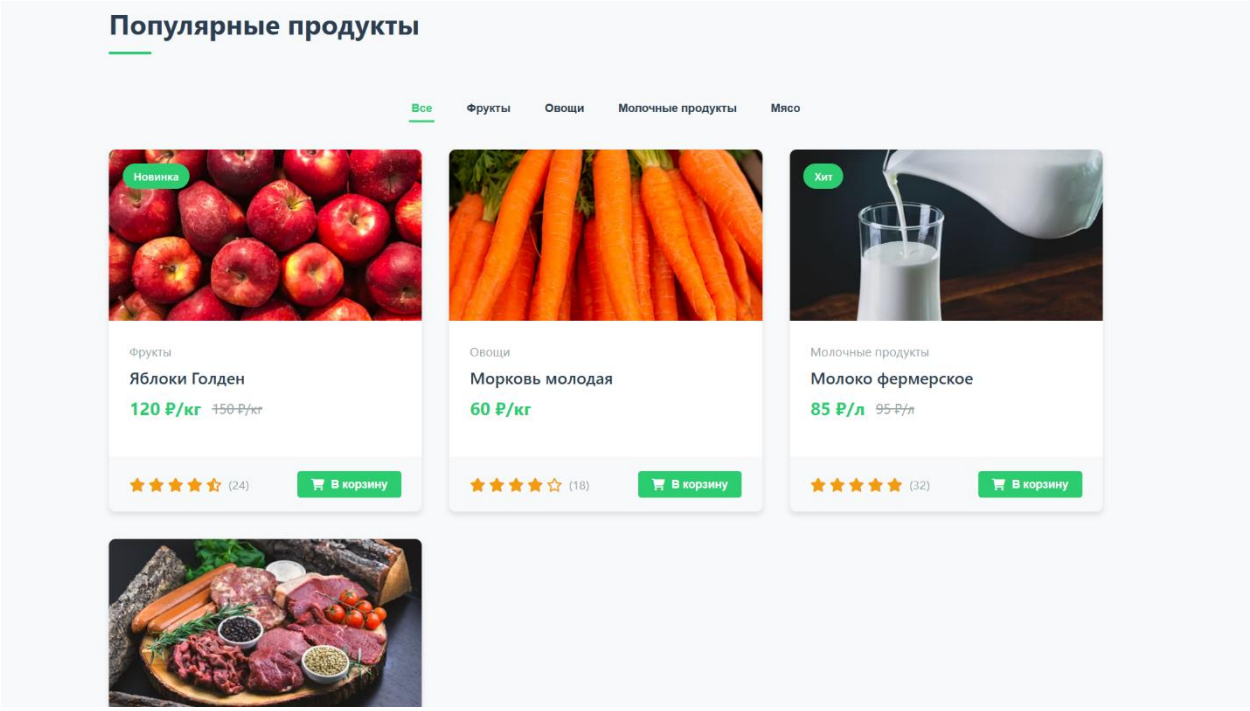
- **Кнопка добавления в корзину**:

```
<button class="add-to-cart">
    <i class="fas fa-shopping-cart"></i>
    <span>В корзину</span>
</button>
```

Важные моменты:

- **Фильтрация**: кнопки фильтров позволяют пользователю отфильтровывать продукты по категориям.
- **Рейтинг**: карточки показывают рейтинг продуктов, помогая пользователям делать выбор.
- **Интерактивность**: кнопки "добавить в корзину" и действия с продуктами делают секцию динамичной.
- **Визуальные элементы**: использование значков (сердце, глаз, корзина) и различных стилей придает странице современный и интерактивный вид.

Получаем следующее :



Важные моменты в секции **"Отзывы"**:

Секция отзывов:

- Секция начинается с тега <section> с классами testimonials и section.
- Это позволяет выделить раздел с отзывами и применить соответствующие стили через CSS.

Заголовок:

- Заголовок <h2> с классом section-title и text-center делает текст "Отзывы наших клиентов" крупным и выровненным по центру.

Сетка отзывов:

- Используется контейнер для организации отзывов в виде карточек.
- Каждый отзыв представлен в отдельной карточке , что позволяет стилизовать каждый отзыв как отдельный элемент.

Контент отзыва:

- Каждый отзыв включает текст отзыва в <p class="testimonial-text">.
- Информация об авторе отзыва (имя и роль) и изображение (внутри).

Изображение автора:

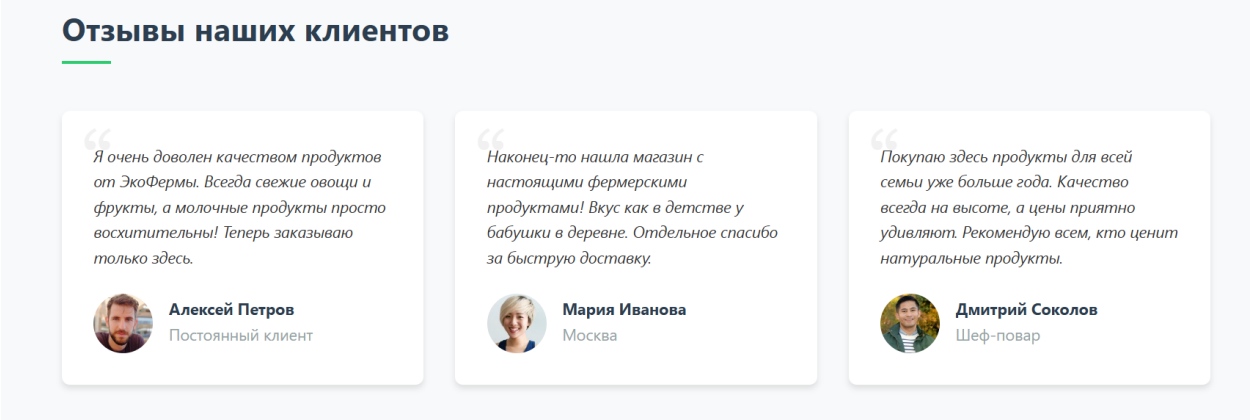
- Включает изображение автора отзыва, получаемое через API randomuser.me, для демонстрации фотографии.
- Это позволяет каждому отзыву быть персонализированным.

Визуальная структура:

- Сетка отзывов организована с использованием карточек, где каждый отзыв структурирован с текстом, изображением и информацией о пользователе, что улучшает восприятие и удобство для пользователя.

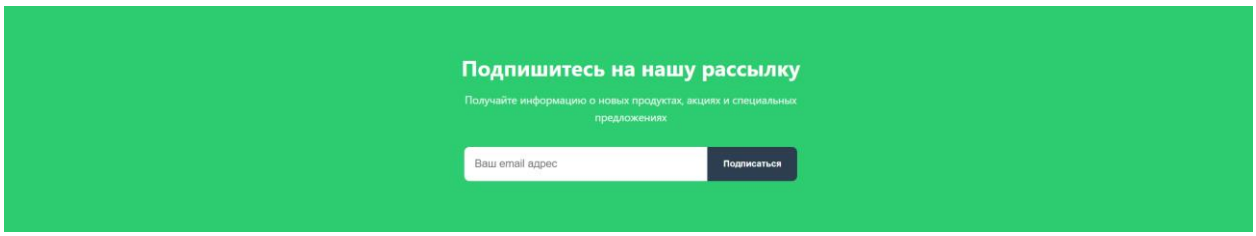
```
<section class="testimonials section">
    <h2 class="section-title">Отзывы наших клиентов</h2>
    <div class="testimonial-grid">
        <div class="testimonial-card">
            <p class="testimonial-text">Очень доволен...</p>
            <div class="testimonial-author">
                <img src="author.jpg" alt="Автор" class="author-img">
                <h4>Алексей Петров</h4>
            </div>
        </div>
        <!-- Другие отзывы -->
    </div>
</section>
```

В итоге получаем следующее:



Следующая секция ‘подписка на рассылку’

```
<!-- Подписка на рассылку -->
        <section class="newsletter">
            <div class="container newsletter-container">
                <h2>Подпишитесь на нашу рассылку</h2>
                <p>Получайте информацию о новых продуктах, акциях и
специальных предложениях</p>
                <form class="newsletter-form">
                    <input type="email" class="newsletter-input"
placeholder="Ваш email адрес" required>
                    <button type="submit" class="newsletter-
btn">Подписаться</button>
                </form>
            </div>
        </section>
```



Секция "Подписка на рассылку":

- Содержит заголовок, описание и форму для ввода email-адреса.
- Форма включает поле для ввода email и кнопку для отправки.

Основные моменты:

- **Тема**: "Подписка на рассылку" с призывом подписаться на новости.
- **Контент**: Текстовое описание для подписки и форма с полем для ввода email.
- **Структура**: Используется section для оформления блока, контейнер для центрации контента, форма с обязательным полем для ввода email и кнопкой отправки.

Следующая часть ‘footer’:

Объяснение кода:

Этот код представляет **подвал** (footer) веб-страницы, который включает несколько секций для различных информационных блоков и социальных ссылок. Подвал помогает пользователям найти ключевую информацию о сайте и компании. Давайте разберем его по частям:

1. **Основная структура подвала (footer)**:
   - Весь подвал находится внутри блока <footer class="footer">.
   - Внутри подвала есть несколько колонок (используется класс footer-col), которые содержат разные секции.
2. **Колонка "О нас"**:
   - Заголовок: <h3>О нас</h3>.
   - Описание компании с информацией об ЭкоФерме.
   - Социальные ссылки (Facebook, Instagram, Telegram, VK), которые представлены с помощью иконок (из библиотеки FontAwesome).

```
<div class="footer-col footer-about">
    <h3>О нас</h3>
    <p>ЭкоФерма – это онлайн-магазин фермерских продуктов высочайшего
качества. Мы сотрудничаем с лучшими фермерами, чтобы предложить вам самые
свежие и экологически чистые продукты.</p>
    <div class="social-links">
        <a href="#" class="social-link"><i class="fab fa-facebook-f"></i></a>
        <a href="#" class="social-link"><i class="fab fa-instagram"></i></a>
        <a href="#" class="social-link"><i class="fab fa-telegram"></i></a>
        <a href="#" class="social-link"><i class="fab fa-vk"></i></a>
    </div>
</div>
```

Колонка "Быстрые ссылки":

- Содержит список ссылок на страницы сайта, такие как "Главная", "О нас", "Каталог" и другие.

- Для каждой ссылки используется иконка и текст.

```
<div class="footer-col footer-links">
  <h3>Быстрые ссылки</h3>
  <ul>
    <li><a href="index.html"><i class="fas fa-chevron-right"></i> Главная</a></li>
    <li><a href="about.html"><i class="fas fa-chevron-right"></i> О нас</a></li>
    <li><a href="#"><i class="fas fa-chevron-right"></i> Каталог</a></li>
    <li><a href="#"><i class="fas fa-chevron-right"></i> Контакты</a></li>
    <li><a href="#"><i class="fas fa-chevron-right"></i> Доставка и оплата</a></li>
    <li><a href="#"><i class="fas fa-chevron-right"></i> Политика конфиденциальности</a></li>
  </ul>
</div>
```

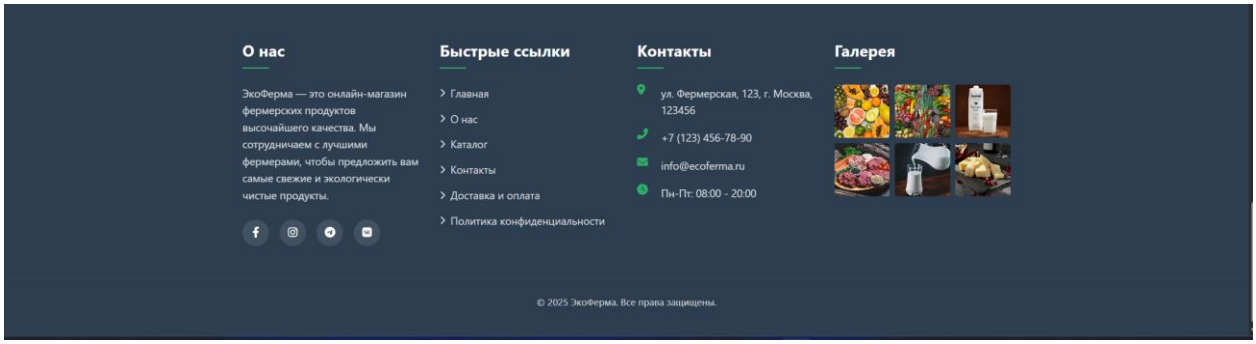
Самые важные моменты кода для копирования:

1. Подвал состоит из 4 колонок:
   - "О нас" с социальной ссылкой.
   - "Быстрые ссылки" с навигацией по сайту.
   - "Контакты" с адресом и телефонным номером.
   - "Галерея" с изображениями.
2. Внизу размещен **copyright** блок.

```
<footer class="footer">
    <div class="container footer-container">
        <div class="footer-col footer-about">
            <h3>О нас</h3>
            <p>ЭкоФерма — это онлайн-магазин фермерских продуктов
высочайшего качества...</p>
            <div class="social-links">
                <a href="#" class="social-link"><i class="fab fa-
facebook-f"></i></a>
                <a href="#" class="social-link"><i class="fab fa-
instagram"></i></a>
                <a href="#" class="social-link"><i class="fab fa-
telegram"></i></a>
                <a href="#" class="social-link"><i class="fab fa-
vk"></i></a>
            </div>
        </div>
        <div class="footer-col footer-links">
            <h3>Быстрые ссылки</h3>
            <ul>
                <li><a href="index.html"><i class="fas fa-chevron-
right"></i> Главная</a></li>
                <li><a href="about.html"><i class="fas fa-chevron-
right"></i> О нас</a></li>
                <li><a href="#"><i class="fas fa-chevron-right"></i>
Каталог</a></li>
            </ul>
        </div>
        <div class="footer-col footer-contact">
            <h3>Контакты</h3>
            <p><i class="fas fa-map-marker-alt"></i> ул. Фермерская,
123, г. Москва, 123456</p>
            <p><i class="fas fa-phone-alt"></i> +7 (123) 456-78-90</p>
        </div>
```

```
            </div>
            <div class="copyright">
                <div class="container">
                    <p>&copy; 2025 ЭкоФерма. Все права защищены.</p>
                </div>
            </div>
</footer>
```

Получаем:



2.4 Использование стилей CSS

На всем сайте будет использован шрифт “ 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif”. Для этого берем нужные вариации этого шрифта из Google Fonts и подключаем в <head>

```
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family= sans-serif:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">
```

И затем задаем всему документу этот шрифт.

```
*{
    font-family: sans-serif;
}
```

Чтобы страница выглядела лучше задаем значение 0 тэгам padding,margin и свойство border-box(border-box говорит браузеру учитывать любые границы и внутренние отступы в значениях, которые вы указываете в ширине и высоте элемента).

```
*{
    font-family: sans-serif;
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
```

Убираем маркировку:

```
ul{
    list-style: none;
}
```

Убираем подчеркивания:

```
a{
    text-decoration: none;
}
```

1. **Переменные CSS (Root Variables)**
   Файл ‘common.css’ где находятся основные части любой страницы

```
:root {
    --primary: #2ecc71; /* Основной цвет */
    --primary-dark: #27ae60; /* Темный основной цвет */
    --secondary: #f39c12; /* Вторичный цвет */
    --dark: #2c3e50; /* Темный цвет для текста */
    --light: #ecf0f1; /* Светлый цвет фона */
    --gray: #95a5a6; /* Серый цвет */
    --danger: #e74c3c; /* Цвет ошибки */
    --success: #27ae60; /* Цвет успеха */
    --white: #ffffff; /* Белый цвет */
    --shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /* Тень для элементов */
    --radius: 8px; /* Радиус скруглений */
}
```

Объяснение: В этих переменных заданы основные цвета и параметры, такие как тени и радиус скруглений. Это позволяет использовать их в разных частях сайта, что упрощает редактирование и стилизацию.

2. **Сброс стилей и базовая настройка**

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
```

Объяснение: Этот код сбрасывает все отступы и внутренние отступы (margin и padding) по умолчанию, а также устанавливает схему расчета размеров элементов с использованием box-sizing: border-box. Это предотвращает нежелательные отступы и делает верстку более предсказуемой.

3. **Типография**

```
h1, h2, h3, h4, h5, h6 {
    font-weight: 700;
    line-height: 1.2;
    margin-bottom: 1rem;
    color: var(--dark);
}
```

Объяснение: Все заголовки (h1-h6) имеют жирный шрифт и одинаковый интервал между строками. Также задан отступ снизу, чтобы разделить заголовки от последующего контента, и цвет шрифта соответствует темному основному цвету сайта.

4. **Кнопки**

```
.btn {
    display: inline-block;
    padding: 0.8rem 1.5rem;
    background-color: var(--primary);
    color: white;
    border: none;
    border-radius: var(--radius);
    font-weight: 600;
    cursor: pointer;
    transition: all 0.3s ease;
    text-align: center;
}

.btn:hover {
    background-color: var(--primary-dark);
    transform: translateY(-2px);
    box-shadow: var(--shadow);
}
```

Объяснение: Кнопки получают стили для фона, цвета текста и скругления углов. При наведении на кнопку изменяется цвет фона и добавляется тень, а также добавляется эффект подъема с помощью transform.

5. **Шапка сайта (Header)**

```
.header {
    background-color: white;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    position: sticky;
    top: 0;
    z-index: 1000;
}
```

Объяснение: Шапка сайта фиксируется в верхней части страницы (при прокрутке) с помощью position: sticky. Также ей добавляется тень для выделения.

6. **Адаптивность**

```
@media (max-width: 768px) {
    .header-container {
        flex-wrap: wrap;
    }

    .nav {
        flex-basis: 100%;
        display: none;
        margin-top: 1rem;
    }

    .nav.active {
        display: block;
    }
}
```

Объяснение: В этом медиа-запросе для экранов с шириной меньше 768px делается шапка сайта более гибкой. Навигационное меню скрывается по умолчанию и появляется только при активации, чтобы сэкономить место на маленьких экранах.

Следующая файл ‘index.css’ для основной страницы то что отличается от других страниц:

Герой секция (Hero Section)

```
.hero {
    position: relative;
    height: 600px;
    background-image: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),
url('https://images.unsplash.com/photo-1542838132-92c53300491e?ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fi
t=crop&w=1920&q=80');
    background-size: cover;
    background-position: center;
    color: white;
    display: flex;
    align-items: center;
    text-align: center;
}
```

Объяснение: Эта секция представляет собой фоновое изображение с градиентом, который затемняет фон, чтобы текст на нем был читаемым. Используется flex для центрирования текста и кнопок внутри секции.

2. Категории

```
css
КопироватьРедактировать
.category-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
    gap: 2rem;
}

.category-card {
    position: relative;
    height: 200px;
    border-radius: var(--radius);
    overflow: hidden;
    box-shadow: var(--shadow);
    transition: all 0.3s ease;
}

.category-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
}
```

Объяснение: Категории представлены в виде карточек, которые отображаются с использованием grid. Для карточек добавлены эффекты при наведении — они поднимаются и получают усиленную тень, что делает их интерактивными.

3. Особенности (Features)

```
.feature-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 2rem;
}

.feature-card {
    background-color: white;
    padding: 2rem;
    border-radius: var(--radius);
    box-shadow: var(--shadow);
```

```
    text-align: center;
    transition: all 0.3s ease;
}

.feature-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
}
```

Объяснение: Эти карточки отображают особенности продукта или услуги. Они используют сетку grid и имеют эффект подъема при наведении, что делает их более заметными и интерактивными.

4**. Популярные продукты (Product Section)**

```
.product-filter {
    display: flex;
    justify-content: center;
    margin-bottom: 2rem;
    flex-wrap: wrap;
    gap: 0.5rem;
}

.filter-btn {
    background-color: transparent;
    border: none;
    padding: 0.5rem 1rem;
    font-weight: 600;
    cursor: pointer;
    position: relative;
    color: var(--dark);
    transition: all 0.3s ease;
}
```

Объяснение: Здесь настроены фильтры для популярных продуктов. Кнопки фильтров реагируют на события hover с изменением цвета и добавлением подчеркивания, чтобы показать активное состояние.

5. Отзывы (Testimonials)

```
.testimonial-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(350px, 1fr));
    gap: 2rem;
}

.testimonial-card {
    background-color: white;
    padding: 2rem;
    border-radius: var(--radius);
    box-shadow: var(--shadow);
    position: relative;
}

.testimonial-card::before {
    content: '\201C';
    font-size: 5rem;
    color: #f1f1f1;
    position: absolute;
    top: 10px;
    left: 20px;
    font-family: Georgia, serif;
}
```

Объяснение: Отзывы представлены в виде карточек, в которых используется символ кавычек (с помощью ::before), чтобы выделить текст отзыва. Это добавляет визуальный стиль и помогает пользователю лучше воспринимать контент.

6. Подписка на рассылку

```
.newsletter {
    background-color: var(--primary);
    padding: 5rem 0;
    color: white;
}

.newsletter-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    text-align: center;
}

.newsletter-form {
    display: flex;
    max-width: 500px;
    width: 100%;
}
```

Объяснение: Секция подписки имеет фоновый цвет, соответствующий основному бренду. Форма для ввода данных подписки использует flex для вертикального выравнивания элементов.

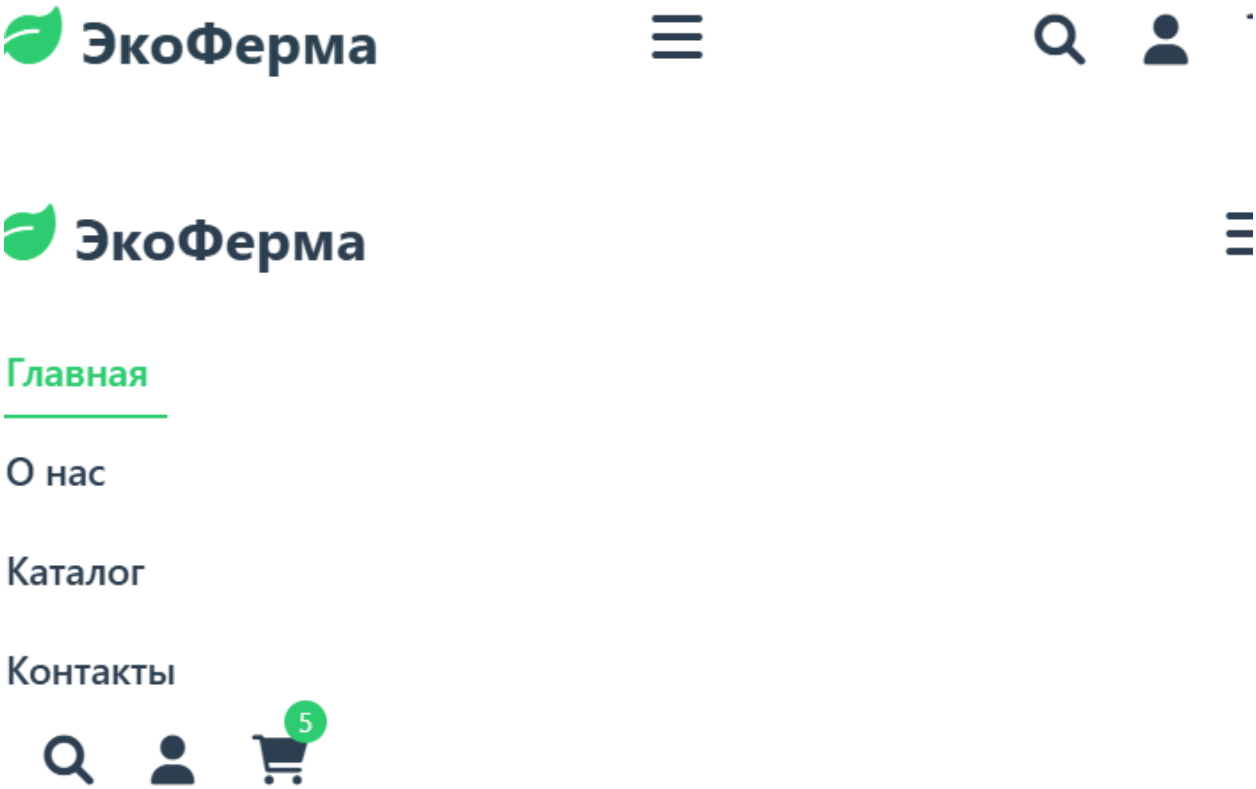
7. Адаптивность

```
@media (max-width: 992px) {
    .hero h1 {
        font-size: 2.8rem;
    }
}

@media (max-width: 768px) {
    .hero {
        height: 500px;
    }

    .hero h1 {
        font-size: 2.2rem;
    }

    .hero p {
        font-size: 1rem;
    }
}

@media (max-width: 576px) {
    .hero {
        height: 400px;
    }

    .hero h1 {
        font-size: 1.8rem;
    }

    .testimonial-grid {
        grid-template-columns: 1fr;
    }
```

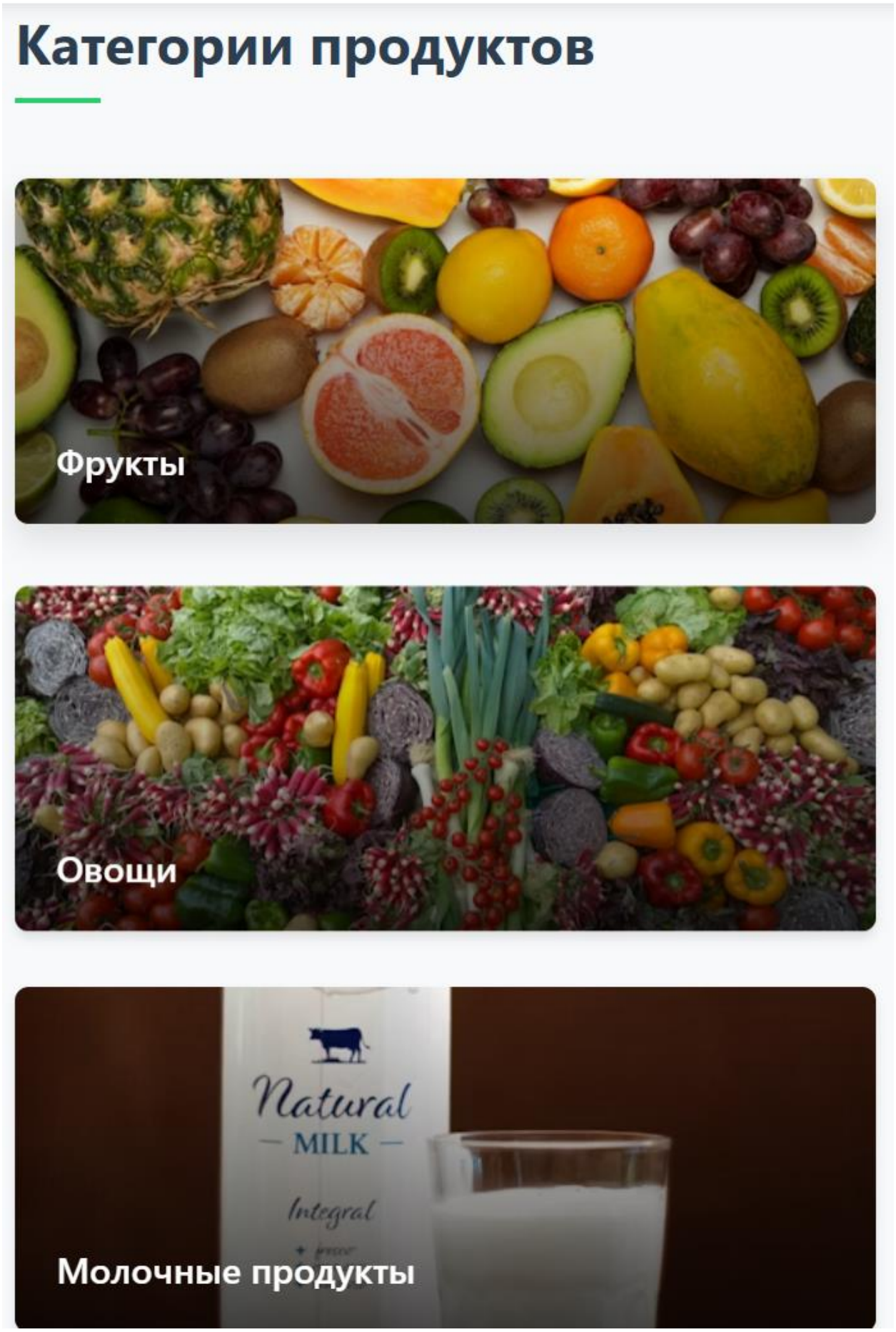
```
}
```

Объяснение: Медиа-запросы адаптируют страницы под различные размеры экранов. Например, уменьшение размеров шрифта на мобильных устройствах и изменение структуры сетки для отзывов и геро-секции.

Теперь перейдем к адаптивному дизайну, многое использовано с помощью ‘bootstrap’ поэтому другое можете ознакомиться на сайте Вот что он добавляет ‘бургер-меню’ который появляется на маленьких экранах при открывании происходит показ, а при обратном закрывается



Другие части кода также используют ‘bootstrap’.


Категории продуктов
Фрукты
Овощи
Natural
MILK
Integral
Молочные продукты

3. РЕЗУЛЬТАТ

За время выполнения курсовой работы были изучены принципы разработки web-сайта:

1. Мы создали веб-страницу используя HTML.

2. Стили выполнены с использованием CSS

3. Создана адаптивная веб-страница с использованием медиа-запросов.

В результате была создана современная, функциональная и эстетичная веб-страница, которая удовлетворяет требованиям заказчиков, удобна для пользователей и соответствует современным трендам в веб-дизайне.

Ссылка на веб-страницу: https://github.com/shalbuzz/Kursovaya_for_Nurlan.git

4. ЛИТЕРАТУРА

1. И. Н. Васильева, Д. Ю. Федоров - WEB-Технологии

2. https://webonto.ru/vvedenie-v-web-tehnologii/
3. https://blog.skillfactory.ru/glossary/bootstrap/
4. Итинсон Кристина Сергеевна - Научная статья “WEB 1.0, WEB 2.0, WEB 3.0: ЭТАПЫ РАЗВИТИЯ ВЕБ-ТЕХНОЛОГИЙ И ИХ ВЛИЯНИЕ НА ОБРАЗОВАНИЕ”
5. https://dic.academic.ru/dic.nsf/ruwiki/75382
6. https://itbukva.com/stati/16377-chto-takoe-web-tekhnologii-kak-rabotayut-sajty.html
7. Зеленко О.В., Валеева Л.Р., Климанов С.Г. - Обзор современных Web – технологий
8. https://aws.amazon.com/ru/what-is/javascript/
9. https://1cloud.ru/blog/www_history_finish
10. https://iit-web-lectures.readthedocs.io/ru/latest/www/css.html

5. ПРИЛОЖЕНИЯ

Репозиторий:

- https://github.com/shalbuzz/Kursovaya_for_Nurlan.git

Страница сайта на хостинге GitHub Pages:

- https://kursovaya-for-nurlan.vercel.app/