**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

**Курc:** 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**

**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………………………………………………………………………3**

1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ………...………………………………………5**
   1. **Web-программирование…………………………………………….……...6**
   2. **HTML ………………………………………………...…………..7**
   3. **Dynamic HTML……………………………………………………………..9**

**1.4.1 Каскадные таблицы стилей……………………………………...9**

**2.ПРАКТИЧЕСКАЯ ЧАСТЬ…………………………………………...……12**

* 1. **Описание предметной области…………………………………………...12**
  2. **Создание окружения для проекта………………………………………..13**
  3. **Верстка HTML страницы…………………………………………………13**
  4. **Использование стилей CSS………………………………………………..22**

**3.РЕЗУЛЬТАТ…………………………………………………………………..274ЛИТЕРАТУРА………………………………………………………………..28**

**5.ПРИЛОЖЕНИЯ……………………………………………………………...29**

**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.

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:

1.1 Технологии серверной стороны (server-side).

1.2 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).

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**

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.</a>**

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.

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

**<HR>** - горизонтальная линия.

**<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.

*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.

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.

**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.

**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)

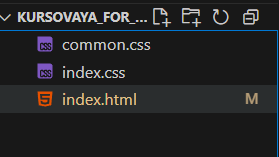


Рис 2.1 . Итоговый вид окружения

**2.3 Верстка HTML страницы**

Открываем ранее созданный “index.html” и пишем стандартный HTML5 код.

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](https://developer.mozilla.org/ru/docs/Glossary/Metadata)) о документе, например его [заголовок](https://developer.mozilla.org/ru/docs/Web/HTML/Element/title), [скрипты](https://developer.mozilla.org/ru/docs/Web/HTML/Element/script) и [страницы стилей](https://developer.mozilla.org/ru/docs/Web/HTML/Element/style).

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)

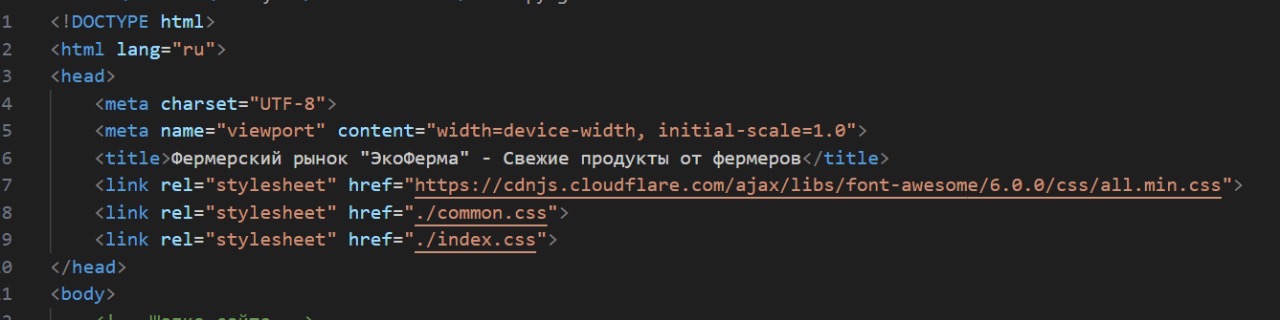


Рис. 2.2 – Оглавление html страницы

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>

                </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>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/ol)), неупорядоченном списке ([<ul>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/ul)), или меню ([<menu>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/menu)).

Затем создаем <div> и помещаем туда иконки:

            <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>

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

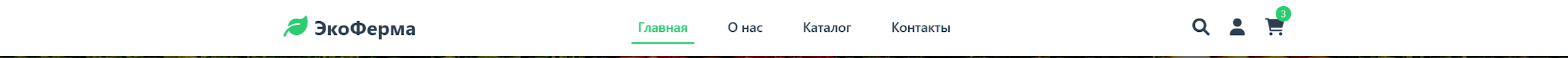


Рис. 2.3 – Внешний вид “Navbar”

<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.

В итоге мы получим следующее:

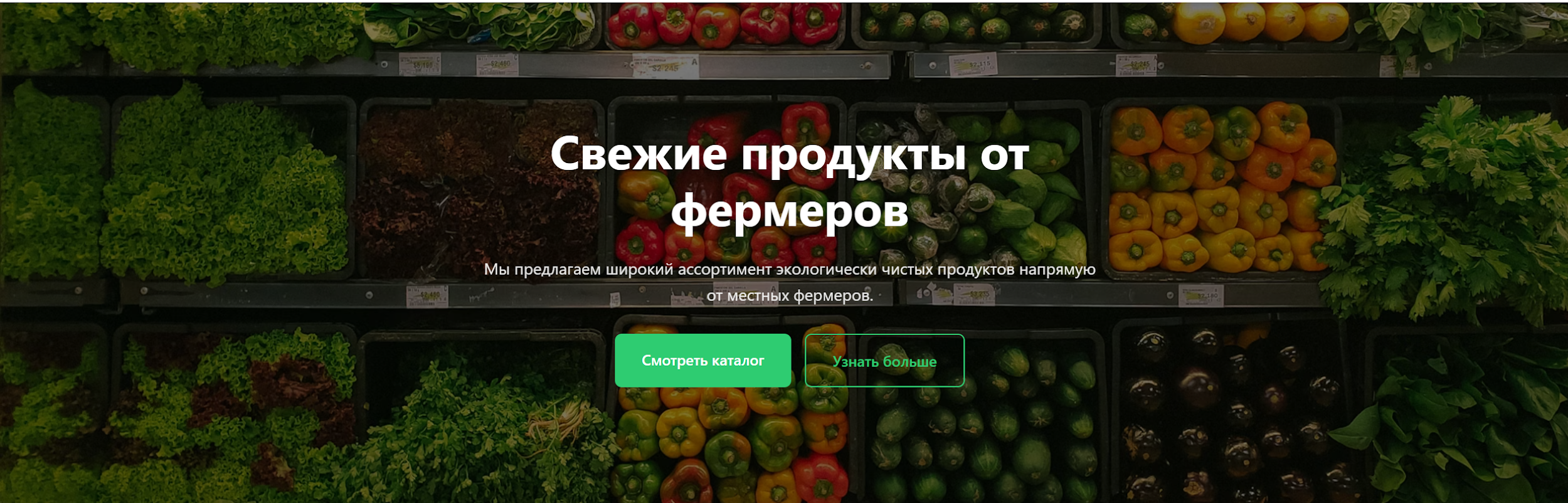


Рис. 2.4 – Внешний вид “Hero секции”

**Следующая часть секция категория:**

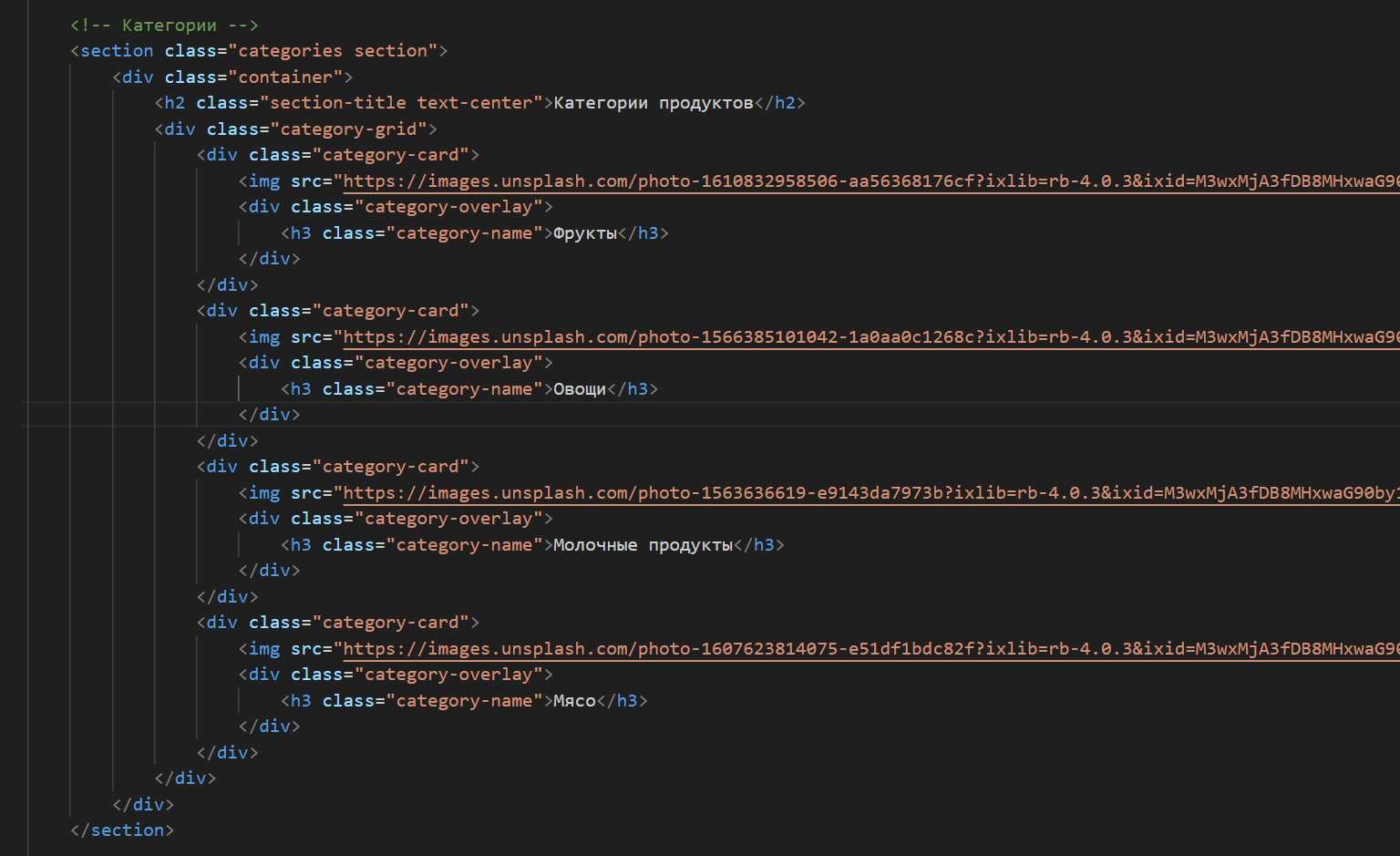


Рис. 2.5 – Объяснение кода “Hero секции”

**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**.

В итоге мы получаем следующее:

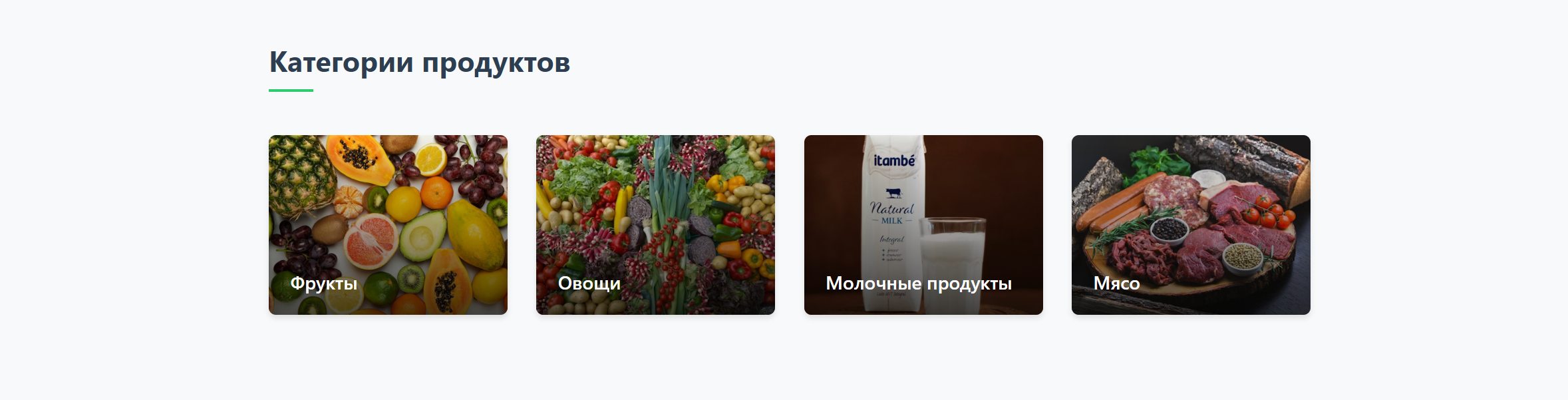


Рис. 2.6 – Внешний вид секции “Категория продуктов”

Следующая секция **Популярные продукты:**

**Напишем самое основное:**

**Сетка с продуктами**:

<div class="product-grid">

<div class="product-card" data-category="fruits">

<!-- Продукт "Яблоки" -->

</div>

.........

</div>

Каждая карточка продукта обернута в <div> 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>

**Рейтинг продукта**:

* Отображаются звездочки рейтинга и количество отзывов.

<div class="product-rating">

<div class="rating-stars">

<i class="fas fa-star"></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>

Получаем следующее :

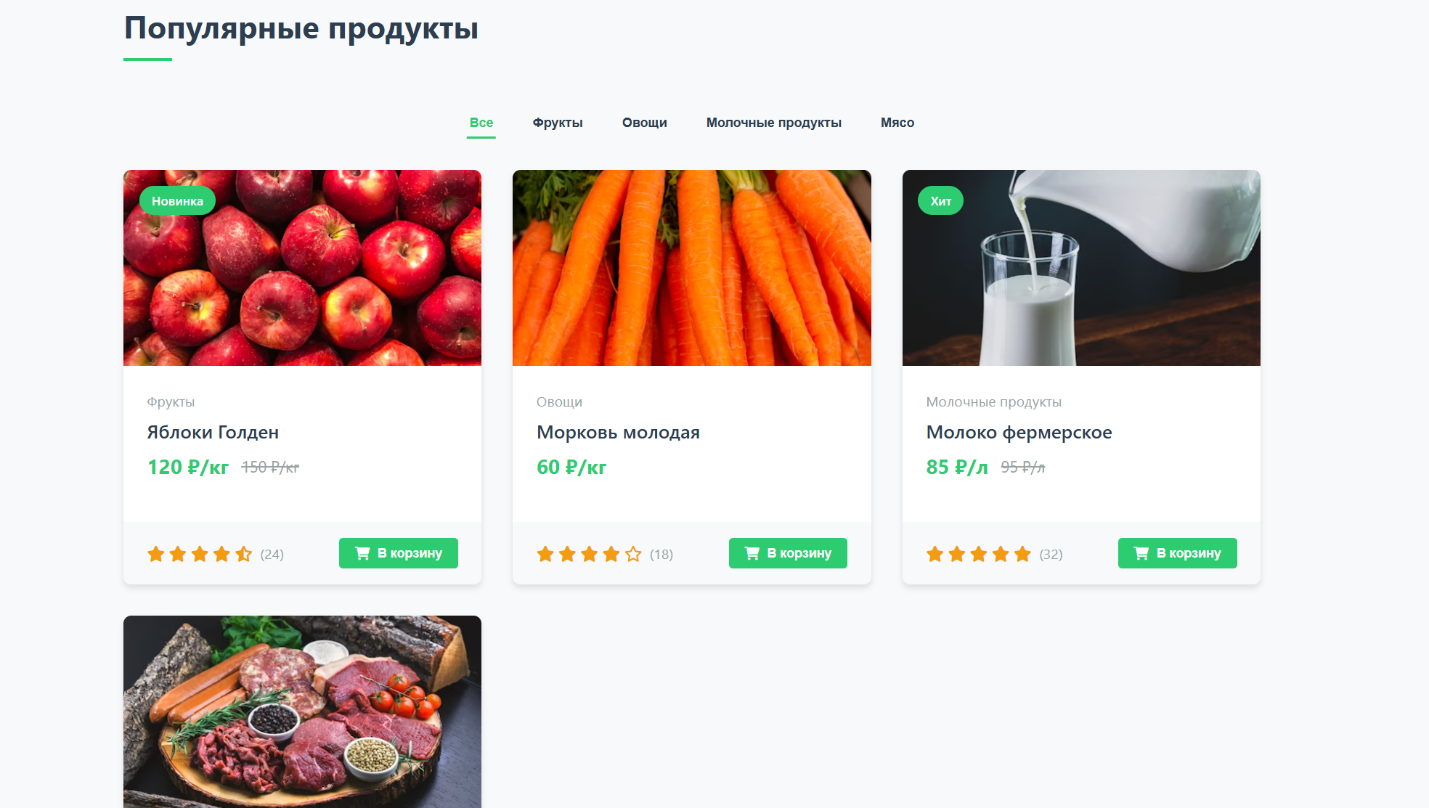


Рис. 2.7 – Внешний вид секции “Популярные продукты”

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.

<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:

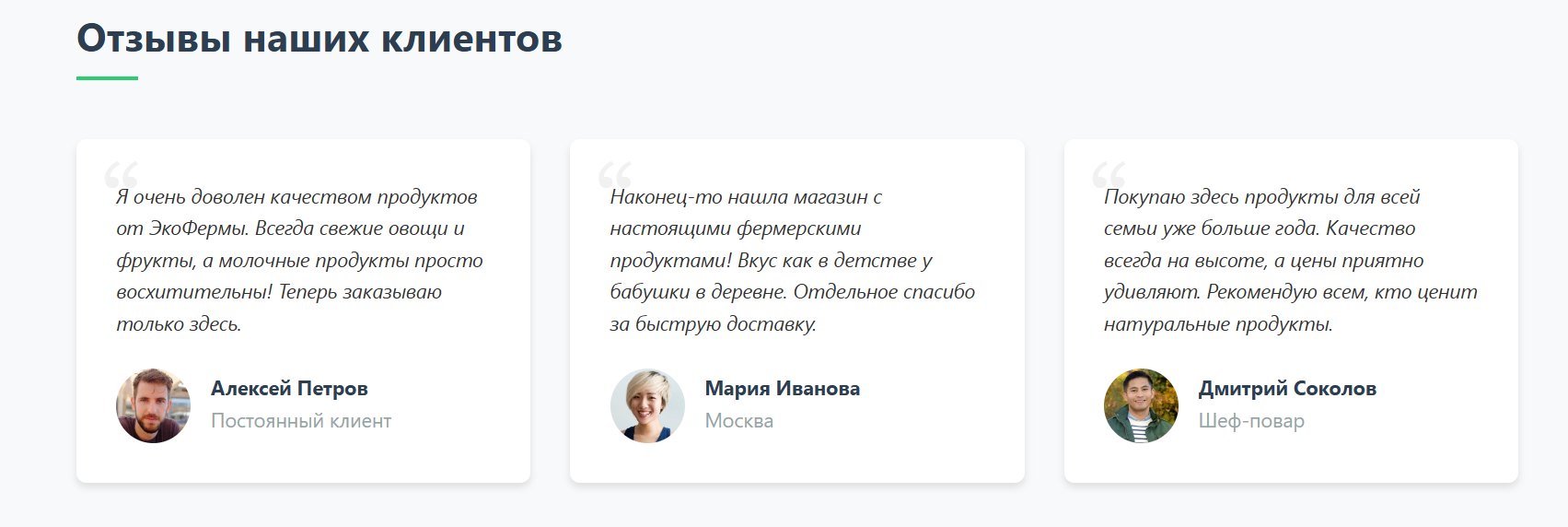


Рис. 2.8 – Внешний вид секции “Отзывы наших клиентов”

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>

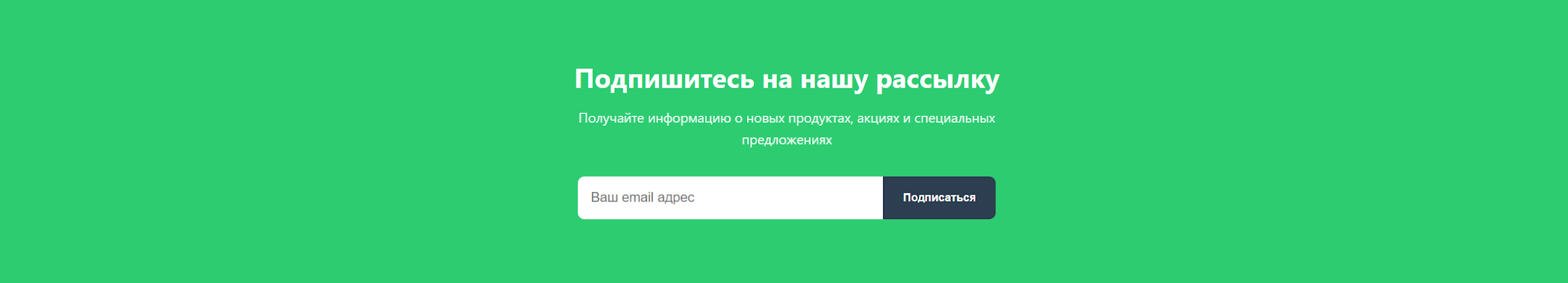


Рис. 2.9 – Внешний вид секции “Подписка на рассылку”

**Секция "Подписка на рассылку"**:

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

**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.

<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-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>

..........

</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.

Получаем:

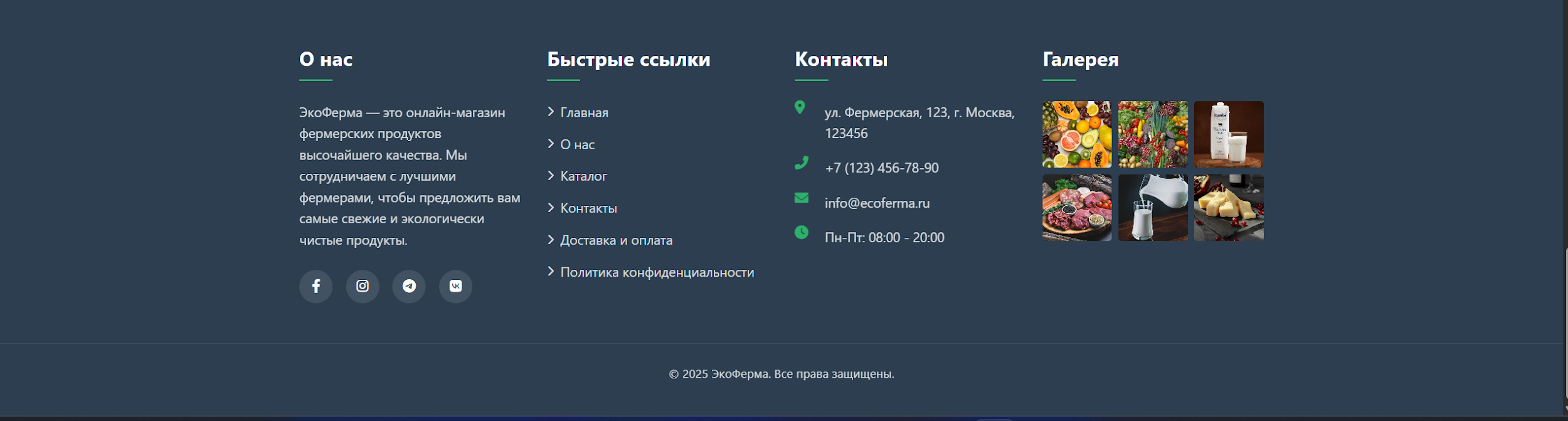


Рис. 2.10 – Внешний вид “Footer”

**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;

}

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.

1. **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.

**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&fit=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.

1. **Категории**

.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;

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;

}

**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.

**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/>