

ОТЧЕТ ПО ПРАКТИКЕ

слушателя

Абдул Карим Элина Найфовна

(фамилия, имя, отчество)

по дополнительной профессиональной программе –
программе профессиональной переподготовки
«РАЗРАБОТКА WEB И МУЛЬТИМЕДИЙНЫХ ПРИЛОЖЕНИЙ»

Слушатель

13.09.2024



(подпись, дата)

Э. Н. Абдул
Карим

(инициалы, фамилия)

ЗАКЛЮЧЕНИЕ РУКОВОДИТЕЛЯ О ПРОХОЖДЕНИИ ПРАКТИКИ

Программа практики выполнена в полном объеме. Во время прохождения практики слушатель продемонстрировал сформированность профессиональных компетенций, соответствующих структуре формируемой профессиональной квалификации.

Результат промежуточной аттестации

Дата	Вид и форма промежуточной аттестации	Оценка (зачтено / не зачтено)
13.09.2024	Зачет в форме защиты отчета по практике	

Руководитель практики,
доцент, канд. экон. наук



13.09.2024

(подпись, дата)

М. А. Медведев

(инициалы, фамилия)

Екатеринбург
2024

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на практику слушателя

Абдул Карим Элина Найфовна

(фамилия, имя, отчество)

по дополнительной профессиональной программе –
программе профессиональной переподготовки
«Разработка web и мультимедийных приложений»

(наименование программы)

1. Задание на практику:

Приобретение практического опыта в области технической поддержки процессов создания (модификации) и сопровождения информационных ресурсов

2. Срок практики: с 26.08.2024 по 13.09.2024

Срок сдачи отчета по практике: 13.09.2024

3. Место прохождения практики:

Школа профессионального и академического образования

Рабочий график (план) проведения практики

Этапы практики	Наименование работ слушателя	Срок	Примечание
организационный	Изучение методических рекомендаций, освоение правил внутреннего трудового распорядка; правил по охране труда и технике безопасности, санитарно-эпидемиологических правил, режима конфиденциальности. Обсуждение с руководителем практики темы индивидуального задания на практическую подготовку в организации, уточнение ее формулировки, составление конкретного последовательного перечня работ, необходимых для выполнения задания.	с 26.08.2024 по 28.08.2024	выполнено
основной	Выполнение индивидуального задания по следующему перечню работ: мероприятия по сбору материала, наблюдение и анализ информации, подготовка процесса тестирования верстки страниц IP, кодирование на языках web-программирования, тестирование IP, работы по резервному копированию IP, установка прав пользователей IP.	с 29.08.2024 по 10.09.2024	выполнено
заключительный	Составление отчета по выполнению индивидуального задания на практику, подготовка к защите отчета. Согласование отчета с руководителем практики.	с 11.09.2024 по 13.09.2024	выполнено

Руководитель от УрФУ


подпись

М. А. Медведев
расшифровка подписи

Задание принял к исполнению (слушатель)


подпись

Э. Н. Абдул Карим

расшифровка подписи

СОДЕРЖАНИЕ

Введение.....	2
1 Анализ предметной области	4
3 Инструментальные средства разработки.....	6
4 Разработка прототипов и дизайн-шаблонов.....	8
5 Разработка клиентской части приложения.....	12
Заключение	31

ВВЕДЕНИЕ

В условиях стремительного развития интернет-технологий и повсеместного онлайн-присутствия бизнесов, веб-сайт становится важным инструментом для увеличения видимости и привлечения клиентов. Создание интерактивного и привлекательного веб-сайта для предприятия, такого как кофейня, является ключевым шагом к привлечению новых клиентов и укреплению связи с существующей аудиторией. Сайт для кофейни является эффективным способом презентации предложений и услуг заведения, а также обеспечения удобного онлайн заказа.

Цель текущего проекта заключается в создании привлекательного и функционального одностраничного веб-сайта для кофейни, который будет представлять заведение в интернете, обеспечивать удобный заказ кофейных напитков, а также собирать обратную связь от посетителей.

В данной работе было принято решение выполнить следующие задачи для реализации всего проекта:

1. Проанализировать область применения создаваемого продукта
2. Спроектировать прототип сайта, выбрать инструменты для его создания
3. Разработать дизайн-шаблоны сайта кофейни, визуально четко воспринимаемыми, а также настроить корректную валидацию и адаптацию внутри.

Отчет представляет собой вводную часть с последующим анализом предметной области. Следующий этап включает в себя создание примитивной архитектуры сайта, чтобы приступить к дальнейшей, более детальной разработке веб-сайта. Далее идет описание используемых технологий с последующим их применением, реализацией в виде

разработки сайта, пошаговой разработкой кода, его пояснением. В итоге получаем конечный результат в виде созданного веб-сайта.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Данный проект создается для небольшой кофейни, являющегося начинающим бизнесом. Такой концепт позволяет ознакомиться с товарами к приобретению, также заказать напиток онлайн. Есть возможность оценить обслуживание для улучшения качества сервиса и предоставляемых услуг по изготовлению напитков.

Предметом работы является сайт для кофейни, представляющий виртуальное пространство для кофейни, содержащий отдельные блоки, как объекты, которые включают в себя информативную часть или простой функционал.

Объектами в данной работе являются:

1. Тексты и изображения – основные элементы контента в каждом блоке, которые помогают передавать информацию посетителям.
2. Форма для заказа кофе – объект, который позволяет пользователям отправить заказ на кофе через сайт.
3. Оценки обслуживания – объект, собирающий обратную связь от посетителей относительно качества обслуживания в кофейне.

Для работы системы происходят процессы отображения информации и взаимодействие с посетителями в виде интерактива заполнения форм. В данном проекте серверная часть не разрабатывается, но если бы она была, то администратор кофейни/бариста смогли бы взаимодействовать с данными, полученными из форм заказа и оценки обслуживания, для обработки заказов, улучшения сервиса и связи с клиентами.

2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ



Рисунок 1 – Архитектура проекта

При проектировании веб-сайта можно выделить две основные части: клиентская и серверная. Клиентская часть представляет собой непосредственно сам одностраничный сайт, с которым взаимодействует пользователь, все компоненты, созданные в процессе работы.

В качестве серверной части был выбран инструмент Local Storage – встроенный механизм хранения данных веб-браузера, который позволяет сохранять информацию на стороне клиента в виде пар ключ-значение. LocalStorage прост в использовании благодаря своим методам, таким как «setItem()» для сохранения данных и «getItem()» для их извлечения. Как раз такие методы и используются в данной работе. Он предоставляет возможность сохранять данные на длительное время, чтобы они были доступны даже после перезапуска браузера. Объем хранилища LocalStorage обычно ограничен (порядка 5–10 МБ на домен), и данные доступны только для скриптов на том же домене, обеспечивая безопасность информации. Для обращения к такому механизму

3 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ

В разработке данного проекта были выбраны классические средства разработки веб-приложения: HTML, CSS, JavaScript.

HTML (HyperText Markup Language) – это язык разметки, позволяющий создавать структуру и содержимое веб-страниц. HTML необходим для определения элементов на странице, таких как заголовки, параграфы, изображения, ссылки и другие. Основная задача HTML - определить структуру документа, чтобы браузер мог правильно отобразить его для пользователя. Особенность HTML заключается в том, что это простой и понятный язык, доступный для большинства веб-разработчиков. HTML используется в разработке сайтов для создания информационной основы, которая впоследствии оформляется с помощью CSS. Стоит отметить, что HTML является основой веб-разработки, без которого невозможно создание функциональных и пользовательских сайтов.

CSS (Cascading Style Sheets) является языком стилей, который используется для стилизации внешнего вида веб-страниц, чтобы придать странице привлекательный внешний вид, управлять расположением элементов, цветами, шрифтами и другими стилями. Особенность CSS заключается в том, что он позволяет разработчикам отделить структуру HTML от ее визуального оформления, делая код более чистым и удобным для поддержки и изменений. С помощью CSS можно создавать адаптивный дизайн, который корректно отображается на различных устройствах и экранах.

JavaScript является интерпретируемым языком программирования, который используется для создания интерактивных элементов на веб-страницах. JavaScript нужен в разработке сайта для добавления динамического поведения, как анимации, обработки событий, проверки

данных, загрузки контента без перезагрузки страницы и других функций. Особенность JavaScript заключается в том, что он выполняется на стороне клиента (в браузере пользователя) и позволяет создавать динамичные и интерактивные пользовательские интерфейсы. В создании нашего проекта данный инструмент использовался только для реализации качественной валидации и для передачи данных в Local Storage.

4 РАЗРАБОТКА ПРОТОТИПОВ И ДИЗАЙН-ШАБЛОНОВ

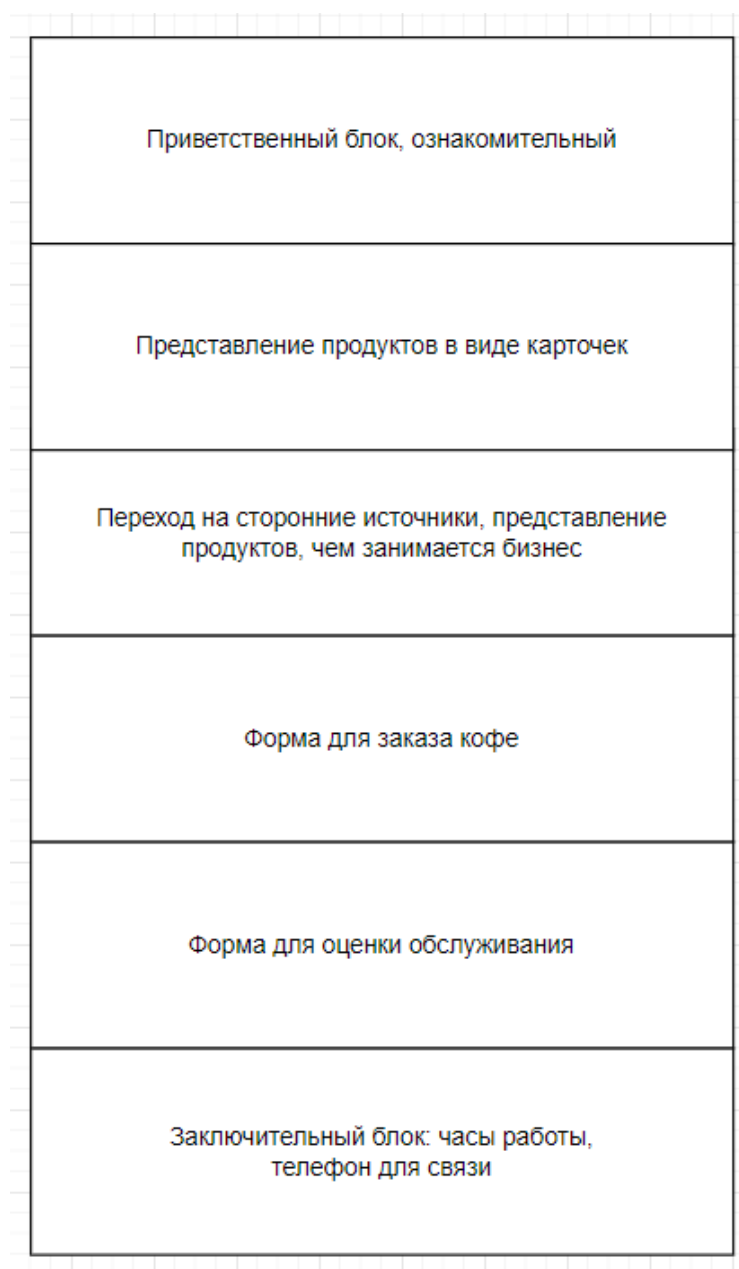


Рисунок 2 – Прототип сайта

Здесь была выбрана общая разметка одностраничного сайта с 6 блоками, расположение которых логически обоснованно. Далее представлены дизайн-шаблоны всего сайта по отдельности.

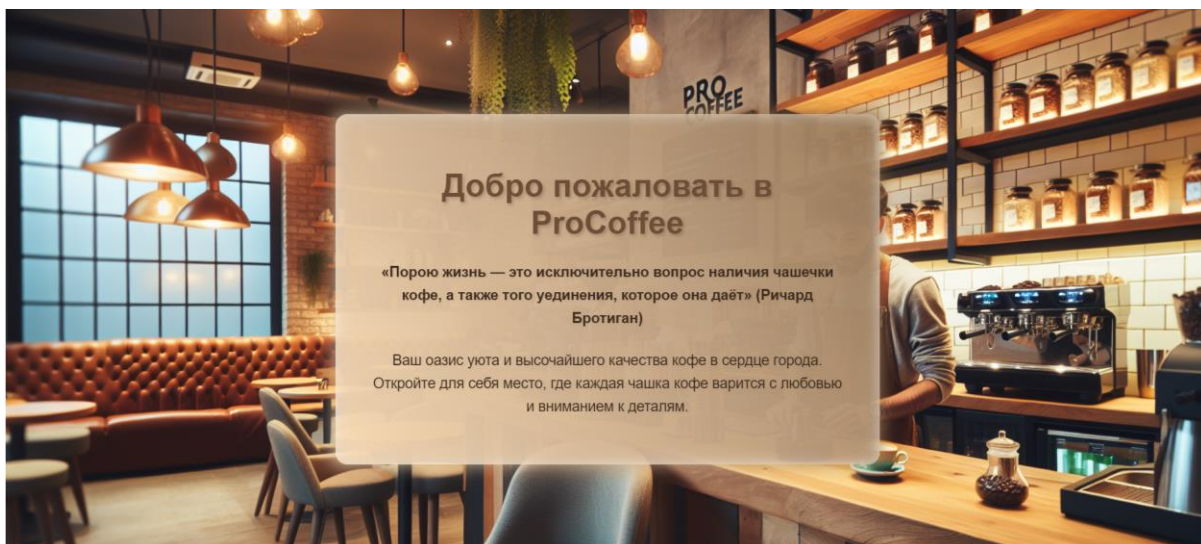


Рисунок 3 – Приветственный блок

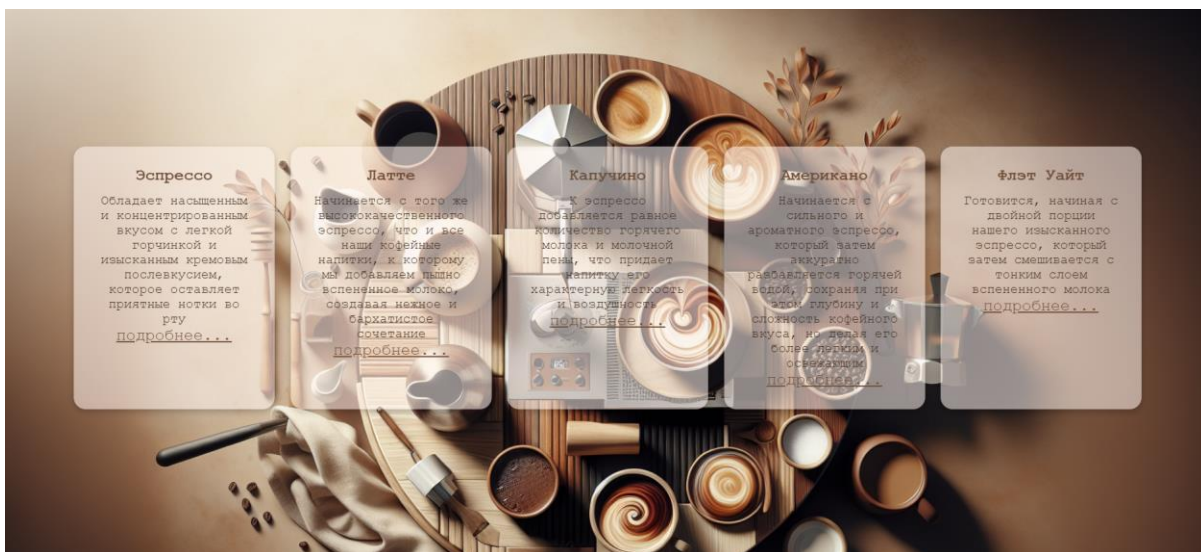


Рисунок 4 – Блок с представлением ассортимента

Что мы печем?

Мы производим выпечку и кондитерские изделия «как дома» – без добавления консервантов, красителей; формула максимально приближена к домашней выпечке.

ЧИТАТЬ ДАЛЕЕ

Где купить?

Наша пекарня все делает на совесть, поэтому для изготовления пирогов, тортов или определенных сортов печенья необходимо от нескольких часов до нескольких дней.


ЧИТАТЬ ДАЛЕЕ

Работать с нами?

Если вы заинтересованы в сотрудничестве с нами, вы можете оставить заявку на нашем сайте. Мы свяжемся и обсудим все детали.

ЧИТАТЬ ДАЛЕЕ

Рисунок 5 – Блок, показывающий дополнительную информацию



Для какого котика напиток?

Имя

Номер телефона

Почта для чека

Почта

Выбор напитка

Выберите ваш напиток

Латте

Капучино

Американо

Моккачино

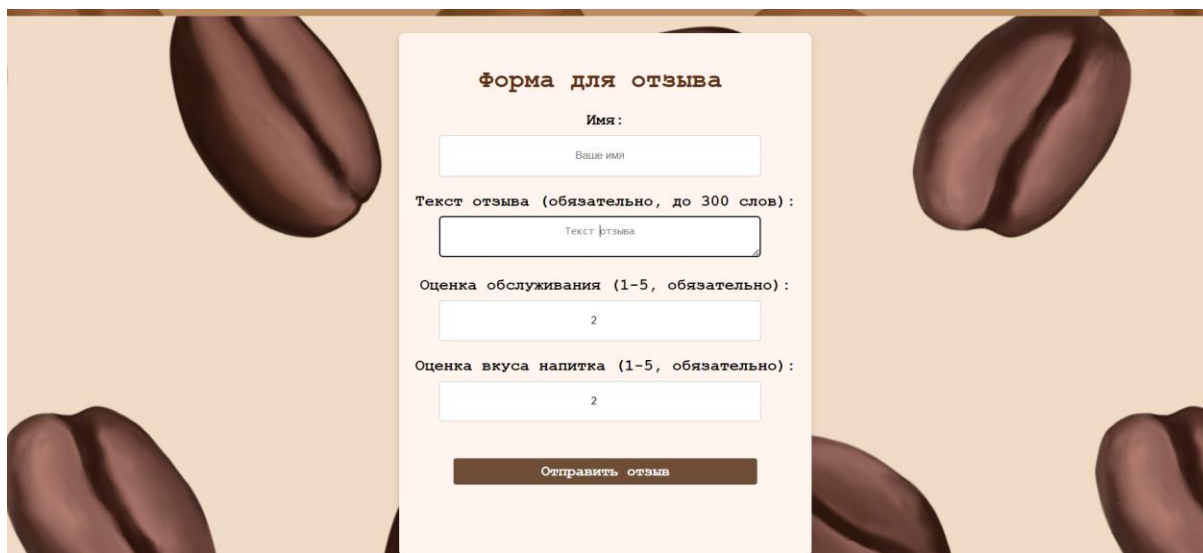
Сделайте вкус ярче!

☐ Корица ☐ Сахар ☐ Натуральные взбитые сливки

Выберите объем и время выдачи

Объем: 200

Рисунок 6 – Форма для заказа



Форма для отзыва

Имя:

Текст отзыва (обязательно, до 300 слов):

Оценка обслуживания (1-5, обязательно):

Оценка вкуса напитка (1-5, обязательно):

Отправить отзыв

Рисунок 7 – Форма обратной связи

(480) 565-7878

Екатеринбург Мира, 32

Понедельник - Пятница с 08:00 до 19:00

Суббота с 09:00 до 18:00

Закрывает в понедельник

Рисунок 8 – Заключительный блок

5 РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ ПРИЛОЖЕНИЯ

Данный проект реализован в достаточно большом объеме. Далее представлен код в виде реализации лишь некоторых элементов и наиболее значимых функций.

Важными блоками в данной работе являются формы для взаимодействия с клиентом – форма для заказа кофе и сбора обратной связи от пользователя.

Первым выступает приветственный блок, назначение которого – поприветствовать пользователя, представить продукт, завлечь клиента визуально, чтобы у него был стимул приобрести товар.

HTML код первого блока.

```
<!-- Приветственный блок -->
<div class="titulnik">
  <title>ProCoffee - Уют и Качество</title>
  <div class="main-block">
    <div class="title-block">
      <h1>Добро пожаловать в ProCoffee</h1>
      <blockquote class="blockquote">
        <p>«Порою жизнь – это исключительно вопрос наличия чашечки кофе, а также того уединения, которое она даёт»</p>
        <cite>Ричард Бротиган</cite>
      </blockquote>
      <h3>Место, где можно насладиться ароматным кофе, отдохнуть от повседневной суеты и насладиться приятной атмосферой, попивая свежесваренный кофе. Здесь можно провести время с друзьями</h3>
    </div>
  </div>
</div>
```

В CSS стили реализация была следующая:

```
body{
  padding: 0px;
  margin: 0px;
  margin-left: -10px;
}
```

```

.titulnik, html {
    height: 100%;
    margin: 0;
    padding: 0;
    font-family: Courier, monospace;
}

.main-block {
    background-image: url('img/background.png');
    background-size: cover;
    background-position: center;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
}

.blockquote {
    font-family: Courier, monospace;
    position: relative;
    padding: 16px 24px;
    margin: 16px 46px;
    font-size: 16px;
    text-align: center;
    color: #3d2b1f;
}

.blockquote:before,
.blockquote:after {
    position: absolute;
    color: #b19372;
    font-size: 100px;
    font-family: Times, sans-serif;
    line-height: 100px;
}

.blockquote:before {
    content: '"';
    left: -30px;
    top: -10px;
}

.blockquote:after {
    content: '"';
    right: -30px;
    bottom: -10px;
}

.blockquote cite {
    color: #916c4c;
}

```



```

    font-size: 15px;
    padding-top: 10px;
}

.title-block {
    background-color: rgba(255, 255, 255, 0.8);
    padding: 40px;
    border-radius: 15px;
    max-width: 600px;
    position: relative;
    animation: fadeIn 5s infinite alternate;
}

.title-block::before {
    content: '';
    position: absolute;
    top: 0px; right: 0px; bottom: 0px; left: 0px;
    border-radius: 15px;
    background: linear-gradient(60deg, rgba(247, 215, 171, 0.8), rgba(191,
155, 116, 0.8), rgba(130, 93, 53, 0.8));
    z-index: -1;
    filter: blur(0px);
}

.title-block h1 {
    font-family: Andale Mono, monospace;
    color: #6f4e37;
    margin-bottom: 20px;
    font-size: 2.5em;
    font-weight: bold;
    text-shadow: 2px 2px 4px rgba(0,0,0,0.2);
}

.title-block p, h4, h3 {
    color: #3d2b1f;
    font-size: 18px;
    line-height: 1.6;
    text-shadow: 1px 1px 2px rgba(0,0,0,0.1);
}

@keyframes fadeIn {
    0% { opacity: 0.5; }
    100% { opacity: 1; }
}

```

``.titulnik, html`` устанавливает общие стили для элементов с классом ``.titulnik`` и для элемента ``html``. Устанавливает высоту всего документа на 100%, обнуляет внутренние отступы и поля, и задает шрифт ``Arial`` для всего документа.

``.main-block`` задает задний фон блока с классом ``.main-block`` с изображением, которое заполняет блок и позиционируется по центру. Размер блока равен высоте видимого экрана (``100vh``). Отцентрировывает содержимое блока по горизонтали и вертикали. Выравнивает текст по центру.

``.blockquote`` стилизует элементы `blockquote`, устанавливая семейство шрифтов на Courier или моноширинный шрифт, добавляя отступы и поля вокруг цитаты, устанавливая размер шрифта, цвет и выравнивание текста. Он также добавляет кавычки до и после цитаты и стилизует текст цитаты внутри `blockquote`.

``.title-block`` создает блок с фоном цвета белого с непрозрачностью 80%, с отступами внутри блока, закругленными углами и ограничением ширины до 600px. Располагает блок относительно других элементов. Применяет анимацию `fadeIn` с изменением непрозрачности блока.

``.title-block::before`` создает псевдоэлемент ``.::before``, который оформляет рамку вокруг блока с градиентом. Размещает псевдоэлемент под основным содержимым блока. Применяет эффект размытия с помощью ``blur(10px)``.

``.title-block h1`` устанавливает цвет заголовка в светло-коричневом цвете, отступ снизу, размер шрифта 2.5em, полужирное начертание и тень текста для придания объемности.

``.title-block p, h4`` устанавливает цвет текста абзацев и подзаголовков, размер шрифта 18px, межстрочный интервал 1.6 и тень текста для легкой тени.

``.@keyframes fadeIn`` задает анимацию `fadeIn`, которая плавно увеличивает непрозрачность элемента с 50% до 100%.

Далее часть кода адаптивности первого блока:

```
/* Для экранов шириной 1200px и больше */
@media only screen and (min-width: 1200px) {
  .title-block {
    Text-align: center;
    max-width: 600px;
    width: auto;
    height: auto;
```

```

        padding: 60px;
    }
    .title-block::before{
        width: auto;
        height: auto;
    }

    .title-block h1 {
        font-size: 2.2em;
        width: auto;
        height: auto;
    }

    .title-block p, .title-block h4, .blockquote {
        font-size: 16px;
        width: auto;
        height: auto;
    }
}

/* Для экранов шириной от 992px до 1199px */
@media only screen and (min-width: 992px) and (max-width: 1199px) {
    .title-block {
        max-width: 700px;
        Text-align: center;
        padding: 50px;
        width: auto;
        height: auto;
    }

    .title-block::before{
        width: auto;
        height: auto;
    }

    .title-block h1 {
        font-size: 2em;
        width: auto;
        height: auto;
    }
    .title-block p, .title-block h4, .title-block h3, .blockquote {

        width: auto;
        height: auto;
    }
}

```

.main-block, как и классы .title-block p, .title-block h4, .title-block h3, устанавливает height: auto; и width: auto;, чтобы контейнер мог изменять размеры в зависимости от ширины экрана. Далее можно заметить, что почти во всех классах установлена данная характеристика, как раз для того, чтобы все блоки адаптировались по ширине экрана.

blockquote: устанавливает width: auto; и height: auto; для адаптивности текста цитаты.

max-width: устанавливает максимально возможную ширину для каждого масштаба, это делается для того, чтобы ограничить расширение блока.

Text-align: center; регулирует положение элементов блока по центру при изменении масштаба.

font-size используется для установки размера шрифта текста. В данном случае мы меняем размер шрифта в соответствии масштабу.

Далее идет второй блок — где можно ознакомиться с основной продукцией, анимация позволяет при наведении мышкой подсветить каждый блок, сделать читаемым текст на каждой карточке. Такой интерактив позволяет пользователю более детально вовлечься в процесс ознакомления с продуктом.

```
<div class="menu">
  <div class="coffee-menu">
    <div class="drink">
      <h3>Эспрессо</h3>
      <p>Обладает насыщенным и концентрированным вкусом с легкой
горчинкой и изысканным кремовым послевкусием, которое оставляет приятные нотки
во рту</p>
      <a
href="https://roastomania.ru/blog/prigotovlenie/vse_o_espresso/">подробнее...<
/a>
    </div>
    <div class="drink">
      <h3>Латте</h3>
      <p>Начинается с того же высококачественного эспрессо, что и
все наши кофейные напитки, к которому мы добавляем пышно вспененное молоко,
создавая нежное и бархатистое сочетание</p>
      <a
href="https://manufactorycoffee.ru/chtotakoelatte">подробнее...</a>
    </div>
    <div class="drink">
      <h3>Капучино</h3>
      <p>К эспрессо добавляется равное количество горячего молока и
молочной пены, что придает напитку его характерную легкость и воздушность</p>
```

```

        <a
href="https://www.nestleprofessional.ru/article/o_kofe/zagadochnaya-istoriya-
kapuchino">подробнее...</a>
    </div>
    <div class="drink">
        <h3>Американо</h3>
        <p>Начинается с сильного и ароматного эспresso, который затем
аккуратно разбавляется горячей водой, сохраняя при этом глубину и сложность
кофейного вкуса, но делая его более легким и освежающим</p>
        <a
href="https://shop.tastycoffee.ru/blog/amerikano">подробнее...</a>
    </div>
    <div class="drink">
        <h3>Флэт Уайт</h3>
        <p>Готовится, начиная с двойной порции нашего изысканного
эспresso, который затем смешивается с тонким слоем вспененного молока</p>
        <a
href="https://russianbarista.ru/blogs/coffee/flatwhite">подробнее...</a>
    </div>
</div>

```

Здесь также можно перейти по ссылкам на сторонние источники, чтобы получить дополнительную информацию о продуктах.

Часть кода адаптивности второго блока:

```

@media only screen and (min-width: 1200px) {
    .coffee-menu{
        Text-align: center;
        width: auto;
        height: auto;
    }
    .menu{
        width: auto;
    }
}
@media screen and (max-width: 992px) {
    .coffee-menu{
        height: auto;
        Text-align: center;
    }
    .menu{
        height: auto;
    }
}
@media only screen and (min-width: 992px) and (max-width: 1199px) {
    .coffee-menu{
        width: auto;
    }
}

```

```

    Text-align: center;
    justify-content: center;
  }
  .menu{
    height: auto;
    width: auto;
  }
}

```

При адаптации блоков coffee-menu, menu под каждый масштаб были использованы характеристики height: auto; и width: auto для изменения их размеров в зависимости от ширины экрана.

text-align: center и justify-content: используются для выравнивания текста по центру. Это делается для центрирования текста и другого контента на различных устройствах, чтобы обеспечить более приятное и удобное отображение.

Подобные характеристики были использованы и для таких размеров, как 768px, 576px.

Дальше идет блок для с представлением информации о сервисе, обслуживании клиентов, привлечении партнеров.

```

<!-- Блок 3 -->
<div class="container">
  <div class="section">
    <h2 class="section-title">Что мы печем?</h2>
    <p class="section-content">Мы производим выпечку и кондитерские изделия «как дома» – без добавления консервантов, красителей; формула максимально приближена к домашней выпечке.</p>
    <a href="https://www.vsedoma-family.ru/catalog/vypechka" class="cta-button cta-button-animated">ЧИТАТЬ ДАЛЕЕ</a>
    <div class="divider"></div>

  </div>
  <div class="section">
    <h2 class="section-title">Где купить?</h2>
    <p class="section-content">Наша пекарня все делает на совесть, поэтому для изготовления пирогов, пирогов, тортов или определенных сортов печенья необходимо от нескольких часов до нескольких дней.</p>
    <a href="https://xn--b1ademjnk7etb.xn--p1ai/" class="cta-button cta-button-animated">ЧИТАТЬ ДАЛЕЕ</a>
  </div>
  <div class="divider"></div>
  <div class="section">
    <h2 class="section-title">Работать с нами?</h2>

```

```

        <p class="section-content">Если вы заинтересованы в сотрудничестве с
нами, вы можете оставить заявку на нашем сайте. Мы свяжемся и обсудим все
детали.</p>
        <a href="https://fr.lifemart.ru/supply" class="cta-button cta-
button-animated">ЧИТАТЬ ДАЛЕЕ</a>
    </div>
</div>

```

Этот блок разбит на три раздела: «Что мы печем?», «Где купить?» и «Работать с нами?». В каждом разделе есть заголовок, описание и ссылка для получения дополнительной информации.

Рассмотрим CSS код адаптации блока.

```

/* Для экранов шириной 1200px и больше */
@media only screen and (min-width: 1200px) {
    .section-title {
        font-size: 30px;
    }
    .section-content {
        font-size: 20px;
    }
    .cta-button {
        padding: 15px 40px;
        border-radius: 6px;
    }
}

/* Для экранов шириной от 992px до 1199px */
@media only screen and (min-width: 992px) and (max-width: 1199px) {
    .section-title {
        font-size: 26px;
    }
    .section-content {
        font-size: 18px;
    }
    .cta-button {
        padding: 12px 30px;
        border-radius: 5px;
    }
}

/* Для экранов шириной от 768px до 991px */
@media only screen and (min-width: 768px) and (max-width: 991px) {
    .section-title {
        font-size: 22px;
    }
    .section-content {
        font-size: 16px;
    }
}

```



```

.cta-button {
  padding: 10px 20px;
  border-radius: 4px;
}
}

/* Для экранов шириной от 576px до 767px */
@media only screen and (min-width: 576px) and (max-width: 767px) {
  .section-title {
    font-size: 18px;
  }
  .section-content {
    font-size: 14px;
  }
  .cta-button {
    padding: 8px 16px;
    border-radius: 3px;
  }
  .section-content, .cta-button {
    padding: 10px 15px;
  }
}

```

Для экранов с минимальной шириной 1200px: .section-title имеет размер шрифта 30px, .section-content имеет размер шрифта 20px., .cta-button имеет отступы в 15px сверху и снизу, 40px слева и справа, а также скругление углов радиусом 6px.

Для экранов с шириной от 992px до 1199px меняем размер шрифта и отступы в соответствии масштабу: .section-title имеет размер шрифта 26px, .section-content имеет размер шрифта 18px, а .cta-button имеет отступы в 12px сверху и снизу, 30px слева и справа, а также скругление углов радиусом 5px. Далее можно заметить то, что чем меньше масштаб, тем меньше мы берём показатели.

После чего идет форма, созданная для заказа кофе.

Код представляет собой HTML форму заказа кофе со множеством полей для заполнения пользователем. В форме присутствуют поля для указания имени (fio), номера телефона (tel), адреса электронной почты (mail) и выбора напитка из списка доступных вариантов. Пользователь также может добавить различные добавки к напитку, выбрав соответствующие чекбоксы. Кроме того, предоставляется возможность выбора объема кофе и времени выдачи заказа. После заполнения всех необходимых полей пользователь может отправить заявку на заказ, нажав на кнопку "Отправить заявку на заказ". Цель данной формы - упростить процесс заказа кофе для пользователя, предоставив удобный и информативный интерфейс, позволяющий выбрать предпочитаемый

напиток, дополнения и параметры заказа. Код обеспечивает пользователей возможность легко и удобно сделать заказ кофе онлайн с указанием всех необходимых деталей и предпочтений.

Данные с формы собираются с помощью инструмента Local Storage. Далее реализована серверная часть для сбора данных с формы.

```
document.addEventListener('DOMContentLoaded', function() {
  const form = document.querySelector('.form');

  form.addEventListener('submit', function(event) {
    event.preventDefault();

    const formData = new FormData(form);
    const data = {};

    // Обработка данных формы
    formData.forEach(function(value, key) {
      if (key === 'fio' || key === 'tel' || key === 'mail' || key ===
'coffee' || key === 'price' || key === 'time') {
        data[key] = value;
      }
      if (key === 'dop') {
        if (!data[key]) {
          data[key] = [];
        }
        data[key].push(value);
      }
    });

    localStorage.setItem('formData', JSON.stringify(data));

    alert('Ваш напиток будет ждать вас к назначенному времени!');
  });

  // Заполнение полей формы значениями из localStorage
  const savedData = JSON.parse(localStorage.getItem('formData'));
  if (savedData) {
    document.getElementById('fio').value = savedData['fio'] || '';
    document.getElementById('phone').value = savedData['tel'] || '';
    document.getElementById('mail').value = savedData['mail'] || '';
    document.querySelector('select[name="coffee"]').value =
savedData['coffee'] || '';
    document.getElementById('price').value = savedData['price'] || '';
    document.getElementById('time').value = savedData['time'] || '';

    // Заполнение значений чекбоксов добавок
    if (savedData['dop']) {
```

```

        savedData['dop'].forEach(function(value) {
            document.querySelector('input[type="checkbox"][value="' +
value + '"]').checked = true;
        });
    }
}
});

```

Код извлекает данные и заполняет соответствующие поля формы с информацией о пользователе и заказе. Если данные присутствуют в `localStorage`, то они извлекаются и заполняют поля формы с именем ('fio'), номером телефона ('tel'), адресом электронной почты ('mail'), выбранным кофе ('coffee'), ценой ('price') и временем заказа ('time'). Также код заполняет значения чекбоксов с добавками, если они были выбраны ранее. Основное назначение кода - сохранить и автоматически заполнить информацию пользователя и заказа при последующих посещениях страницы. Это улучшает пользовательский опыт и экономит время на повторное ввод данных.

```

@media screen and (max-width: 1500px) {
    .imag {
        width: 500px;
        display: block;
        margin-left: auto;
        margin-right: auto;
    }
    .form{
        width: 90%;
    }
}
@media screen and (max-width: 1300px) {
    .container1{
        max-width: 1150px;
    }
    .main_block{
        justify-content:center;
        display: flex;
        flex-direction: row;
        align-items: center;
    }
    .imag {
        width: 500px;
        display: block;
        margin-left: auto;
        margin-right: auto;
    }
}

```

```

    }
    .form{
        width: 80%;
        margin-left: -10px;
        margin-bottom: 10px;
    }
    .tit{
        margin-left: 50px;
    }
}
@media screen and (max-width: 1200px) {
    .container1{
        max-width: 1100px;
    }
    .imag {
        width: 500px;
        margin-top: 10px;
    }

    .form{
        text-align: center;
        width: 70%;
        margin-bottom: 20px;
    }

    .tit{
        margin-left: 0px;
    }
}

```

Для экранов с максимальной шириной 1500px: изображение (imag) имеет фиксированную ширину 500px и центрируется, а форма (form) занимает 90% ширины контейнера.

Для экранов с максимальной шириной 1300px: ограничивается ширину .container1 до 1150px, центрируется и настраивается макет для .main_block, и настраиваются размеры и отступы .imag и .form для улучшенного отображения.

Для экранов с максимальной шириной 1200px: код дополнительно настраивает макет, уменьшая ширину .container1, центрирует и устанавливает размеры .imag и .form, и переустанавливает .tit.

Для экранов с максимальной шириной 1100px: код настраивает размеры и отступы .form и .imag, центрирует элементы с помощью .main_form, и переустанавливает .tit.

Для экранов с максимальной шириной 992px: код настраивает .container1 и дочерние элементы для лучшей адаптивности, центрирует и устанавливает размеры .imag, и настраивает типографику для более мелких экранов.

Следующий блок содержит форму для отзыва и оценки обслуживания нашей кофейни.

```
<!-- Форма для оценки обслуживания -->
<div class="main_form">
  <div class="container1">
    <div class="main_block">
      <div class="blocks">
        
      </div>
      <div class="blocks">
        <h2 class="tit1">Форма для отзыва</h2>
        <form id="feedbackForm" class="form-coffee">
          <div class="input-group">
            <label for="name" class="tex">Имя:</label>
            <input type="text" id="name" name="name" class="input"
placeholder="Ваше имя">
          </div>
          <div class="input-group">
            <label for="review" class="tex">Текст отзыва (обязательно, до
300 слов):</label>
            <textarea id="review" name="review" class="input"
onblur="checkFeedback();" required placeholder="Текст отзыва"></textarea>
            <span class="error-message" id="error-message"></span>
          </div>
          <div class="input-group">
            <label for="service-rating" class="tex">Оценка обслуживания
(1-5, обязательно):</label>
            <input type="number" id="service-rating" name="service_rating"
class="input" onblur="checkserviceRating()" required min="1" max="5"
placeholder="1-5">
            <span id="service-rating-error" class="service-rating-
error"></span>
          </div>
          <div class="input-group">
            <label for="drink-taste-rating" class="tex">Оценка вкуса
напитка (1-5, обязательно):</label>
            <input type="number" id="drink-taste-rating"
name="drink_taste_rating" class="input" onblur="checkdrinkTasteRating();"
required min="1" max="5" placeholder="1-5">
            <span id="drink-taste-rating-error" class="drink-taste-rating-
error"></span>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
```

```

        <div class="input-group">
            <button type="submit" class="but" >Отправить отзыв</button>
        </div>

    </form>
</div>
</div>
</div>

```

В форме содержатся поля для ввода имени, текста отзыва, оценки обслуживания и оценки вкуса напитка. Также присутствуют элементы для вывода ошибок при вводе данных. При отправке формы вызываются функции `checkFeedback()`, `checkserviceRating()`, и `checkdrinkTasteRating()`, которые выполняют проверку введенных данных. Цель данной формы - собрать обратную связь от клиентов о качестве обслуживания и вкусе напитков, а также улучшить общее впечатление о предоставляемых услугах.

Функции, вызываемые в коде описаны в js файле:

```

/*Форма отзыва*/

function checkFeedback() {
    let Feedback = document.getElementById("review").value;
    let regex = /^[a-zA-Z\s]{2,300}$/;

    if (regex.test(Feedback)) {
        document.getElementById("review").style.color = "green";
        document.getElementById("review").innerHTML = "Данные введены
корректно";
        return true;
    }
    else {
        document.getElementById("review").style.color = "red";
        document.getElementById("error-message").innerHTML = "Введите от 2 до
300 символов";
        return false;
    }
}

function checkserviceRating() {
    let serviceRating = document.getElementById("service-rating").value;
    let regex = /^[1-5]$/;

    if (regex.test( serviceRating)) {
        document.getElementById("service-rating").style.color = "green";
    }
}

```

```

        document.getElementById("service-rating").innerHTML = "Данные введены
корректно";
        return true;
    }
    else {
        document.getElementById("service-rating").style.color = "red";
        document.getElementById("service-rating-error").innerHTML = "Оценка
обслуживания должна быть от 1 до 5";
        return false;
    }
}

function checkdrinkTasteRating() {
    let drinkTasteRating = document.getElementById("drink-taste-
rating").value;
    let regex = /^[1-5]$/;

    if (regex.test(drinkTasteRating)) {
        document.getElementById("drink-taste-rating").style.color = "green";
        document.getElementById("drink-taste-rating").innerHTML = "Данные
введены корректно";
        return true;
    }
    else {
        document.getElementById("drink-taste-rating").style.color = "red";
        document.getElementById("drink-taste-rating-error").innerHTML =
"Оценка вкуса напитка должна быть от 1 до 5";
        return false;
    }
}

```

Функция `checkFeedback()` проверяет введенный текст отзыва на соответствие заданному регулярному выражению, которое требует от 2 до 300 символов в верхнем или нижнем регистре и пробелы. При успешной проверке изменяется цвет текстового поля на зеленый и выводится сообщение об успешном вводе данных. В случае несоответствия условиям проверки, текстовое поле помечается красным цветом, и выводится сообщение об ошибке, требующее ввода от 2 до 300 символов. Функция возвращает логическое значение (true/false) в зависимости от результата проверки.

Функция `checkserviceRating()` проверяет введенную оценку обслуживания на соответствие заданному регулярному выражению, которое требует цифры от 1 до 5. При успешной проверке меняется цвет оценочного поля на зеленый и выводится сообщение об успешном вводе данных. В случае несоответствия условиям проверки, цвет поля изменяется

на красный, и выводится сообщение с предупреждением. Функция возвращает true, если данные корректны, и false в случае ошибки.

Функция `checkdrinkTasteRating()` аналогична предыдущим функциям, но проверяет ввод пользователя на соответствие оценке вкуса напитка от 1 до 5. Если условия проверки выполнены, цвет поля меняется на зеленый, и выводится сообщение об успешном вводе данных. В случае несоответствия цвет поля становится красным, и появляется сообщение с предупреждением. Функция возвращает true или false в зависимости от результатов проверки.

```
// Сохранение данных из формы отзыва в localStorage
const feedbackForm = document.getElementById('feedbackForm');

feedbackForm.addEventListener('submit', function(event) {
    event.preventDefault();

    const formValues = new FormData(feedbackForm);
    const data = {};

    formValues.forEach(function(value, key) {
        data[key] = value;
    });

    localStorage.setItem('feedbackFormData', JSON.stringify(data));

    alert('Данные отзыва успешно сохранены в localStorage!');
});

// Заполнение полей формы значениями из localStorage
const savedFeedbackFormData =
JSON.parse(localStorage.getItem('feedbackFormData'));
if (savedFeedbackFormData) {
    document.getElementById('name').value = savedFeedbackFormData['name'] ||
'';
    document.getElementById('review').value = savedFeedbackFormData['review']
|| '';
    document.getElementById('service-rating').value =
savedFeedbackFormData['service_rating'] || '';
    document.getElementById('drink-taste-rating').value =
savedFeedbackFormData['drink_taste_rating'] || '';
}
```

Код сохраняет данные из формы отзыва в `localStorage`, используя объект `FormData`, и выводит уведомление. При загрузке страницы код заполняет поля формы данными из `localStorage`, если они там есть. Он проверяет наличие данных по ключу 'feedbackFormData' в `localStorage` и заполняет поля формы значениями из объекта `savedFeedbackFormData`.

Основное предназначение кода - сохранить и восстановить данные отзыва пользователя, чтобы он мог видеть и редактировать свой предыдущий отзыв без повторного заполнения формы. Код облегчает пользовательский опыт и сохраняет введенные данные между посещениями страницы.

Для данного блока был написан следующий код адаптации.

```
@media screen and (max-width: 1200px) {
  .feedback-form-wrapper {
    padding: 15px;
  }
  .input-group textarea,
  .input-group input {
    margin-left: 20px;
  }
  .input-group label {
    margin-bottom: 3px;
  }
  .input-group input[type="text"],
  .input-group input[type="number"],
  .input-group textarea,
  .input-group button {
    width: 70%;
    padding: 8px;
  }
  .error-message,
  .service-rating-error,
  .drink-taste-rating-error {
    margin-left: 30px;
  }
}
@media screen and (max-width: 992px) {
  .tit1 {
    text-align: center;
    margin-left: 0;
  }
  .input-group textarea,
  .input-group input {
    margin-left: 0;
  }
}
@media screen and (max-width: 768px) {
  .feedback-form-wrapper {
    padding: 10px;
  }
  .input-group input[type="text"],
  .input-group input[type="number"],
```

```

.input-group textarea,
.input-group button {
  width: 60%;
  padding: 6px;
}
}

```

`.feedback-form-wrapper`: Устанавливает отступы для контейнера формы обратной связи.

`.input-group textarea, .input-group input`: Устанавливает отступ слева для полей ввода внутри контейнера `.input-group`.

`.input-group label`: Устанавливает отступ снизу для меток полей ввода внутри `.input-group`.

`.input-group input[type="text"], .input-group input[type="number"], .input-group textarea, .input-group button`: Устанавливает ширину и отступы для полей ввода и кнопок внутри `.input-group`.

`.error-message, .service-rating-error, .drink-taste-rating-error`: Устанавливает отступ слева для сообщений об ошибках.

`.tit1`: Выравнивает текст по центру и устанавливает отступ слева для элемента с классом `.tit1`

`span.error-message, span.service-rating-error, span.drink-taste-rating-error`: Настраивает размеры и расположение сообщений об ошибках.

Заключительный блок наименее объемный в реализации, он нужен только для представления контактов и обозначения часов работы.

```

<div class="container2">
  <div class="col">
    <h2 class="whb">(480)565-7878</h2>
    <p class="white1">Екатеринбург Мира, 32</p>
    <p class="white2">Понедельник – Пятница с 08:00 до 19:00 </p>
    <p class="white2">Суббота с 09:00 до 18:00 </p>
    <p class="white3">Закрит в понедельник</p>
  </div>
</div>

```

ЗАКЛЮЧЕНИЕ

В данной работе был создан веб-сайт кофейни, который сочетает в себе красивый дизайн, адаптивность для различных устройств, валидацию формы и использование локального хранилища для отправки данных. Размещение контента в блоках и элементах формы позволяет пользователям легко взаимодействовать с сайтом.

CSS код был использован для стилизации сайта, что включает в себя выбор цветовой палитры, шрифтов, отступов, анимаций и многих других декоративных элементов, делающих сайт привлекательным и функциональным. Для адаптации сайта к разным экранам и устройствам был написан специальный CSS-код, который реагирует на изменение размеров экрана и корректирует расположение и размеры элементов для оптимального отображения.

JavaScript код был использован для валидации формы, чтобы обеспечить корректный ввод данных пользователем и предотвратить ошибочные отправки. Это позволяет улучшить пользовательский опыт и управлять потоком данных на сайте. Кроме того, использование локального хранилища в JavaScript помогает сохранить данные пользователя на его устройстве без необходимости отправки на сервер, что улучшает скорость и безопасность обработки информации.

Данная работа объединила различные технологии и методики для создания функционального, красивого и адаптивного веб-сайта кофейни, который удовлетворяет требованиям современных пользователей.