**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**Факультет: Информационные технологии и управление**

**Кафедра: Компьютерная инженерия**

**Предмет: «Веб-системы и технологии»**

**Курсовая работа**

**Тема** «Создание онлайновой информационной системы для городских телефонных служб»

**Группа:** 680.22

**Курc:** 3

**Специальность:** Информационные технологии

**Код специальности:** 050616

**Студент**: Усейнова Лейла

**Руководитель:** **асс. Халилов М. Э.**

**Зав.кафедрой: доц.Рагимова Н.А.**

**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 Усейнова Лейла**

**Зачетная книжка**

**Руководитель курсовой работы асс. Халилов М. Э.**

**Срок выдачи**

**Дата сдачи**

**Тема курсовой работы «** Создание онлайновой информационной

системы для городских телефонных служб **»**

**Отзыв руководителя курсовой:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подпись рук.курсовой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подпись зав.кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата защиты курсовой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Оценка\_\_\_\_\_\_\_\_\_\_**

**Пред.комиссии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**Члены комиссии: 1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**3.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ………………………..……………………………….…5**

1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ…..….………….…………………8**
   1. **Web-программирование……..…………………….……….……...8**
   2. **HTML ………………………….…...…………..9**
   3. **Dynamic HTML……………………………………………………..17**

**1.3.1 Каскадные таблицы стилей…………………………..………...18**

**1.3.2 JavaScript…………………………………………..…...21**

**1.4 Bootstrap……………………………………….…………………..23**

1. **ПРАКТИЧЕСКАЯ ЧАСТЬ……………………….…….…...……26**
   1. **Описание предметной области………………….….……………26**
   2. **Создание окружения для проекта………….…………………..27**
   3. **Верстка HTML страницы..………….………………………………30**
   4. **Использование стилей CSS…..……………….………………..…..34**

**3. РЕЗУЛЬТАТ……..…………..……………….……………..……..45**

**4. ЛИТЕРАТУРА…………….…………………….………….…………46**

**5.ПРИЛОЖЕНИЯ……………………………………………..………...47**

**ВВЕДЕНИЕ**

В условиях стремительного развития информационных технологий и цифровизации всех сфер жизни особенно актуальной становится автоматизация и упрощение доступа к справочной информации. Одной из таких сфер является работа городских телефонных служб, предоставляющих гражданам различные справки, консультации и экстренные контакты. Однако традиционные формы предоставления этой информации (по телефону или через бумажные справочники) становятся все менее удобными, особенно для молодого поколения, привыкшего к цифровым сервисам.

Целью данной работы является разработка онлайновой информационной системы, которая позволит пользователям быстро и удобно находить необходимые телефоны городских служб через интернет. Такой сайт обеспечит круглосуточный доступ к актуальной информации, улучшит взаимодействие граждан с различными структурами городской инфраструктуры и повысит общую эффективность их работы.

В рамках проекта будет спроектирована и реализована веб-страница с понятным интерфейсом, разделами по категориям служб (экстренные, медицинские, коммунальные, справочные и др.), а также функцией обратной связи. В разработке будут использованы современные технологии HTML, CSS и JavaScript, что обеспечит кроссбраузерную совместимость и адаптивность сайта для разных устройств.

Таким образом, создание подобной информационной системы является актуальной задачей, направленной на улучшение качества городской жизни и повышение цифровой грамотности населения.

**1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1** **Web-программирование**

Web-программирование — это область информационных технологий, охватывающая процесс создания веб-приложений и веб-сайтов, работающих через браузер и доступных через сеть Интернет. Оно играет ключевую роль в современной цифровой среде, позволяя организациям, государственным структурам и частным лицам публиковать и распространять информацию, оказывать услуги и обеспечивать взаимодействие с пользователями в онлайн-пространстве.

Основу web-программирования составляют три базовые технологии: HTML, CSS и JavaScript.

**HTML (HyperText Markup Language)** — это язык разметки гипертекста, определяющий структуру и содержимое веб-страниц. Он позволяет организовывать текст, добавлять изображения, видео, формировать таблицы, списки, формы ввода и другие элементы пользовательского интерфейса. HTML выступает скелетом любой веб-страницы и определяет её логическую структуру.

**CSS (Cascading Style Sheets)** — каскадные таблицы стилей, отвечающие за оформление веб-страницы. С помощью CSS разработчики могут задавать цветовую палитру, шрифты, размеры элементов, отступы, расположение блоков и анимационные эффекты. CSS отделяет визуальное оформление от структуры HTML, что упрощает поддержку и изменение внешнего вида сайта без необходимости вмешательства в его содержимое.

**JavaScript** — это язык программирования, применяемый для реализации интерактивного поведения на страницах. Он позволяет реагировать на действия пользователя (например, нажатие кнопок, ввод текста, прокрутку), динамически изменять содержимое, управлять формами, валидировать вводимые данные, а также взаимодействовать с сервером без перезагрузки страницы посредством технологии AJAX. Таким образом, JavaScript обеспечивает "живость" и интерактивность веб-интерфейса.

Современное web-программирование делится на две основные части:

* **Клиентская (frontend) разработка**, которая охватывает всё, что происходит в браузере пользователя. Она включает работу с HTML, CSS, JavaScript и различными фреймворками и библиотеками, такими как Bootstrap, React, Vue.js и др.
* **Серверная (backend) разработка**, которая происходит на стороне сервера и обеспечивает обработку запросов, взаимодействие с базами данных и бизнес-логику приложения. В серверной части могут использоваться языки PHP, Python, Ruby, Java, C#, Node.js и другие.

Кроме того, в процессе web-разработки важное значение имеют следующие аспекты:

* **Адаптивность** — корректное отображение сайта на различных устройствах (ПК, планшеты, смартфоны).
* **Кроссбраузерность** — совместимость сайта с различными браузерами (Chrome, Firefox, Safari, Edge и др.).
* **Безопасность** — защита пользовательских данных и предотвращение уязвимостей (например, XSS, CSRF, SQL-инъекций).
* **UX/UI-дизайн** — обеспечение удобства и логичности взаимодействия пользователя с интерфейсом.

Также разработка веб-сайтов часто включает в себя использование систем управления версиями (например, Git), различных редакторов кода (VS Code, Sublime Text), инструментов сборки (Webpack, Gulp) и среды тестирования.

В рамках данной курсовой работы предполагается использование классического стека клиентских технологий (HTML, CSS и JavaScript) для создания адаптивного и функционального информационного сайта, предоставляющего пользователю доступ к справочной информации о городских телефонных службах. Такой сайт не только упростит поиск нужных телефонов и контактов, но и повысит цифровую доступность сервисов городской инфраструктуры.

Таким образом, web-программирование представляет собой фундаментальную область современной разработки, позволяющую создавать удобные и эффективные средства взаимодействия между системой и пользователем в онлайн-среде.

**1.2****HTML**

HTML (HyperText Markup Language) является основным языком разметки для создания структуры веб-страниц в системе онлайн-информационной системы для городских телефонных служб. Этот стандартизированный язык служит каркасом для веб-страницы, который позже наполняется стилями с помощью CSS и функциональностью через JavaScript.

Каждая HTML-страница имеет расширение .html и начинается с обязательного указания типа документа через тег <!DOCTYPE html>. Внутри тега <html> размещается вся структура страницы, которая делится на два ключевых раздела: <head> и <body>.

Раздел <head> включает метаинформацию о странице:  
• Указание кодировки символов с помощью тега <meta charset="UTF-8">  
• Тег viewport для обеспечения адаптивности на мобильных устройствах  
• Заголовок страницы <title>, который отображается во вкладке браузера  
• Подключение внешних ресурсов, таких как CSS-стили и шрифты

Основное содержимое страницы, которое видит пользователь, находится в теле документа, между тегами <body>. Структура страницы обычно включает:

1. Шапку сайта (<header>), где размещаются логотип, навигационное меню (<nav>) и важные ссылки
2. Главный контент (<main>), разделённый на тематические секции с использованием тега <section>
3. Подвал (<footer>), содержащий контактную информацию и дополнительные ссылки

Для форм, например, для запроса информации по телефонным услугам, используются специальные элементы:  
• Контейнер формы <form>, где указывается метод передачи данных (GET или POST)  
• Поля ввода <input> различных типов, таких как text, date, number  
• Выпадающие списки <select>, с вариантами выбора через теги <option>  
• Чекбоксы <input type="checkbox"> для дополнительных услуг  
• Кнопки для отправки формы <button type="submit">

Семантическая разметка с использованием тегов <article>, <section>, <aside> способствует улучшению доступности контента и помогает поисковым системам правильно интерпретировать структуру страницы. Результаты поиска могут быть представлены в виде карточек, которые оформляются с использованием тегов <div> или <article>, с чёткой иерархией заголовков (<h2>-<h5>).

Современный HTML5 предоставляет множество функций для создания удобного интерфейса информационной системы для городских телефонных служб, включая встроенную валидацию форм, мультимедийные элементы и возможности интеграции с другими веб-технологиями через API.

**1.3 Dynamic HTML**

Dynamic HTML (DHTML) — это концепция, которая объединяет HTML, каскадные таблицы стилей (CSS), скриптовые языки и объектную модель документа (DOM). Этот подход позволяет создавать динамичные и интерактивные веб-страницы. Для работы с DHTML обычно используется JavaScript или VisualBasic, однако JavaScript стал наиболее популярным и широко применяемым языком для реализации таких решений.

С помощью DHTML можно создавать веб-страницы, которые реагируют на действия пользователя в реальном времени. Он позволяет гибко и эффективно обрабатывать данные запросов и формировать динамический ответ на основе изменений на странице, не требующих перезагрузки. Важной особенностью DHTML является возможность реализации таких интерфейсных решений, как **Drag'n'Drop** — перетаскивание объектов по экрану. Это позволяет создавать сложные интерактивные элементы, игры и другие веб-сервисы, которые требуют активного взаимодействия с пользователем.

Примечательно, что для работы с DHTML достаточно лишь стандартного веб-браузера, который будет интерпретировать и отображать страницу. В отличие от традиционных подходов, здесь нет необходимости в постоянных обращениях к серверу или базе данных для обновления содержимого страницы, что значительно улучшает производительность и делает приложение более отзывчивым.

***1.3.1******Каскадные таблицы стилей***

Каскадные таблицы стилей (CSS — Cascading Style Sheets) — это инструмент для определения внешнего вида HTML-документов. С помощью CSS можно настраивать оформление элементов веб-страницы, включая шрифты, цвета, размеры, отступы, расположение блоков и их поведение при взаимодействии с пользователем. Использование CSS позволяет отделить структуру документа (HTML) от его визуального представления, что упрощает процесс разработки и делает его более гибким и масштабируемым.

В веб-приложениях, таких как информационная система для городских телефонных служб, CSS играет важную роль в создании удобного и эстетически привлекательного интерфейса. Он помогает оформить страницы сайта в едином стиле, например, страницы с контактами, услугами или запросами, улучшая восприятие системы пользователями.

Ключевые возможности CSS:

* **Оформление текста**: выбор шрифтов, цветов, размера текста, выравнивание и межстрочный интервал.
* **Оформление блоков**: стили для границ, фонов, теней, отступов и рамок.
* **Размещение элементов**: использование различных моделей, таких как блочная, строчная, flexbox и grid.
* **Визуальные эффекты**: создание анимаций, плавных переходов и псевдоклассов (например, :hover и :focus).
* **Адаптивность**: медиазапросы (@media) позволяют изменять стиль страницы для разных размеров экранов, от мобильных устройств до больших мониторов.

CSS использует принцип каскадности, который позволяет нескольким стилям применяться к одному элементу. В случае конфликта приоритет определяется следующими критериями:

1. **Специфичность селектора**: селекторы id имеют больший приоритет, чем классы, а те, в свою очередь, — выше, чем селекторы по тегу.
2. **Порядок подключения**: если два правила имеют одинаковую специфичность, применяется последнее по порядку.
3. **Правило !important** — оно отменяет все другие стили, за исключением других правил !important с более высокой специфичностью.

CSS можно подключать различными способами:

1. **Встроенные стили** — задаются прямо в атрибуте style элемента HTML.
2. **Внутренние стили** — размещаются в теге <style> внутри самого HTML-документа.
3. **Внешние стили** — подключаются через тег <link>, что позволяет повторно использовать стили и упрощает их поддержку. Пример подключения внешнего файла:

html

КопироватьРедактировать

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

**Роль CSS в системе онлайн-информационной системы для городских телефонных служб**:

* CSS помогает создать удобный и приятный интерфейс для пользователей.
* Он выделяет важные элементы интерфейса, такие как кнопки и формы для отправки запросов.
* С помощью CSS можно упорядочить элементы страницы, например, контактную информацию и навигационные меню.
* Он позволяет обеспечивать адаптивную верстку, чтобы сайт правильно отображался на всех устройствах, от мобильных телефонов до десктопов.
* CSS улучшает внешний вид сайта, делая его более современным и профессиональным, что способствует увеличению доверия со стороны пользователей.

Таким образом, применение CSS в разработке информационной системы для городских телефонных служб помогает не только улучшить визуальное оформление сайта, но и повысить удобство его использования для конечных пользователей.

**1.3.2 JavaScript**

JavaScript (JS) — это язык программирования, который играет ключевую роль в разработке динамичных и интерактивных веб-приложений. Для онлайновых информационных систем, таких как системы городских телефонных служб, JS позволяет создавать интерфейсы, которые мгновенно реагируют на действия пользователя, обеспечивая удобство и скорость работы. В этом контексте JavaScript выполняет множество важнейших функций, от обработки данных и проверки информации до обновления контента в реальном времени.

**1. Интерактивность и обработка событий**

Одной из главных задач JavaScript в информационной системе является создание интерактивных элементов, которые реагируют на действия пользователей. Например, при вводе запроса или выборе опций из списка система должна немедленно показывать результаты без необходимости перезагружать страницу. Это важное преимущество для пользователей, так как они могут быстрее получить нужную информацию о номерах телефонов, времени работы служб или статусе звонков.

JavaScript позволяет отслеживать такие действия, как клики, ввод текста, прокрутка страницы и другие. Когда пользователь взаимодействует с элементами интерфейса, система автоматически обновляет информацию, показывая нужные результаты или уведомления. Это существенно улучшает пользовательский опыт и делает использование системы более удобным.

**2. Валидация данных**

Важным аспектом работы любой информационной системы является точность вводимых данных. В случае с системой для городских телефонных служб, это может касаться таких данных, как номера телефонов, адреса, контактные данные. JavaScript используется для предварительной проверки данных, введённых пользователем, до их отправки на сервер.

Например, если пользователь вводит номер телефона, JavaScript может проверить его формат и убедиться, что он соответствует установленным правилам, прежде чем данные будут переданы в базу данных. Если ввод неверный, система сразу уведомит пользователя о необходимости исправить ошибку, что сокращает количество ошибок и повышает удобство работы с системой.

**3. Асинхронное обновление данных**

Онлайновая система для телефонных служб должна предоставлять пользователям актуальную информацию в реальном времени, например, о статусе линии, времени ожидания или наличии свободных операторов. JavaScript позволяет обновлять данные на странице без необходимости её перезагрузки, что делает интерфейс более динамичным и современным.

Для этого используются асинхронные запросы, которые отправляют данные на сервер и получают ответ без обновления всей страницы. Это позволяет показывать актуальную информацию, такую как свободные линии или время ожидания ответа, в реальном времени. Такие обновления происходят автоматически и не требуют действий со стороны пользователя, что делает систему более удобной и эффективной.

**4. Удобство интерфейса и адаптивность**

Кроме динамической обработки данных, JavaScript также важен для создания удобных и адаптивных интерфейсов. Это особенно важно для городских телефонных служб, которые обслуживают различных пользователей с разными устройствами. С помощью JavaScript можно сделать интерфейс таким образом, чтобы он автоматически адаптировался под различные размеры экранов, от мобильных телефонов до больших мониторов, обеспечивая тем самым комфортное использование системы на любых устройствах.

Кроме того, JavaScript помогает организовывать и управлять элементами на странице, такими как выпадающие меню, формы для ввода данных, кнопки для отправки запросов. Это значительно улучшает взаимодействие с системой и делает процесс поиска информации или связи с телефонной службой быстрым и простым.

**5. Интеграция с другими сервисами**

JavaScript также может быть использован для интеграции с другими веб-сервисами и API. Например, для работы с внешними базами данных, для проверки доступности телефонных линий через сторонние сервисы или для интеграции с системами, которые позволяют отслеживать статус звонков в реальном времени. Это расширяет функциональность системы, позволяя пользователю получать ещё более точную информацию и улучшать взаимодействие с различными элементами телефонной службы.

**2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

**2.1 Описание предметной области**

Сайт CityTel представляет собой онлайн-платформу для предоставления услуг в сфере связи. Основной задачей сайта является обеспечение пользователей необходимой информацией о компании, услугах и контактных данных, а также возможность связи с компанией через форму обратной связи.

Основными разделами сайта являются:

* **Главная страница:** представление общей информации о компании, а также кнопки для навигации по основным разделам.
* **О нас:** подробная информация о компании, её истории и миссии.
* **Контакты:** форма обратной связи для связи с компанией, а также предоставление контактной информации.
* **FAQ:** раздел с часто задаваемыми вопросами, что позволяет пользователю быстро найти ответы на стандартные запросы.

Сайт должен быть удобным и функциональным, обеспечивая комфортный доступ как с мобильных устройств, так и с настольных ПК. Главными требованиями к веб-страницам являются:

1. **Юзабилити:** Структура и оформление сайта должны обеспечивать удобство навигации и использования.
2. **Адаптивность:** Сайт должен корректно отображаться на различных устройствах (мобильных телефонах, планшетах, ПК).
3. **Доступность:** Информация должна быть доступна пользователям с различными возможностями, включая поддержку экранных читалок и другие технологии.

В данной курсовой работе рассматривается создание и верстка сайта CityTel с учётом всех этих аспектов, а также использование современных стандартов HTML5, CSS и JavaScript для реализации функционала и дизайна.

**2.2 Создание окружения для проекта**

Перед началом разработки сайта CityTel было необходимо создать и настроить рабочее окружение. Этот процесс включает установку всех необходимых инструментов, программ и создание структуры проекта. Ниже описаны основные шаги и инструменты, использованные при подготовке среды разработки.

**Выбор редактора кода:**

Для верстки и написания скриптов использовался редактор Visual Studio Code (VS Code) благодаря его широким возможностям, поддержке расширений и удобному интерфейсу.

**Пояснение:** VS Code позволяет быстро создавать файлы, подключать плагины (например, Live Server для предпросмотра HTML-страниц), подсвечивает синтаксис и облегчает работу с HTML, CSS и JavaScript.

**Структура проекта:**

CityTel/

├── index.html

├── about.html

├── contact.html

├── style.css

└── script.js

**Пояснение:** Структура проекта включает три HTML-страницы: главную, страницу "О нас" и "Контакты". Внешний файл стилей style.css содержит все CSS-правила для сайта. В script.js размещён JavaScript-код для динамических элементов.

**Подключение CSS и JavaScript:**

Во всех HTML-документах CSS-файл подключается через тег link, а JavaScript — через script:

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

...

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

**Пояснение:** Эти строки подключают внешний файл стилей и скриптов, обеспечивая единый стиль и поведение на всех страницах сайта.

**Используемые технологии:**

* **HTML5** — для структурирования контента
* **CSS3** — для оформления внешнего вида страниц
* **JavaScript** — для реализации динамических функций (мобильное меню, аккордеон, валидация формы)

**Инструменты и расширения:**

* Расширение Live Server — для локального предпросмотра HTML-страниц
* Расширение Prettier — для автоматического форматирования кода
* Инструмент DevTools в браузере — для отладки и анализа элементов страницы

**Заключение:**

Создание и настройка среды разработки является важным этапом в работе над любым веб-проектом. В случае сайта CityTel была организована простая и эффективная структура, использованы современные инструменты и обеспечена возможность дальнейшего масштабирования и поддержки проекта.

**2.3 Верстка HTML страницы**

HTML-код сайта CityTel отвечает за структуру и содержимое всех страниц: главной, страницы "О нас", "Контакты" и других. Он реализован с использованием стандартной семантической разметки HTML5, что делает страницы понятными как для пользователей, так и для поисковых систем. Далее приведён подробный построчный разбор ключевых элементов с примерами кода и пояснениями.

****

****

Код представляет собой шапку сайта телекоммуникационной компании, выполненную по стандартам HTML5. Основные элементы:

1. Базовая структура

Декларация HTML5

Язык контента - английский

UTF-8 кодировка

1. Адаптивный дизайн

Viewport для мобильных устройств

Гамбургер-меню (3 линии)

Контейнер для управления макетом

1. Навигация

Логотип с выделением "Tel"

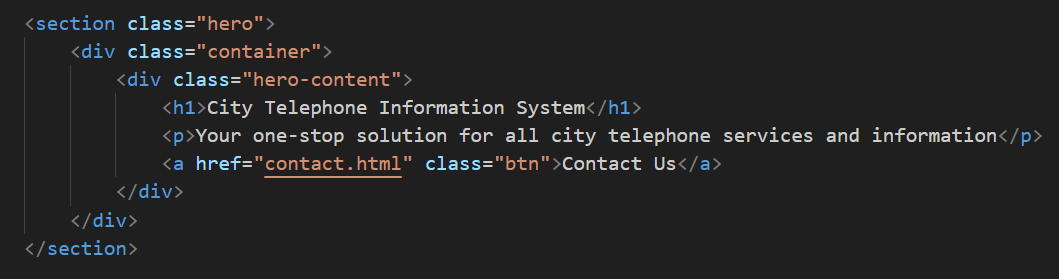
Главное меню (Главная, О нас, Контакты)

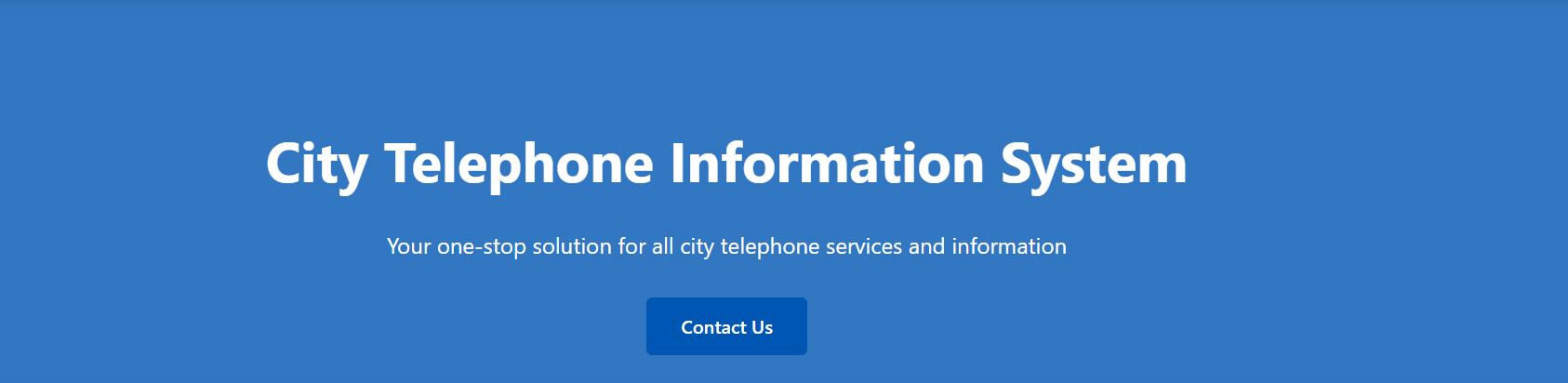
Активный пункт выделен

1. Ресурсы

Внешний CSS-файл

Иконки Font Awesome





Данный фрагмент кода представляет собой главный баннер (hero section) сайта телефонной компании. Разберём его структуру и функциональные элементы:

1. **Структурные элементы**

Секция помечена классом "hero" для стилизации

Используется контейнер (div class="container") для контроля ширины содержимого

Внутренний блок "hero-content" содержит основной контент

1. **Контентная часть**

Заголовок h1 с названием системы

Описательный текст под заголовком

Кнопка-ссылка "Contact Us" с классом "btn" для стилизации

1. **Функциональные особенности**

Ссылка ведёт на страницу контактов (contact.html)

Кнопка имеет отдельный класс для особого оформления

Простая и понятная структура призыва к действию

1. **Технические аспекты**

Используются семантические HTML-теги

Чёткая иерархия заголовков (h1)

Логичное расположение элементов

Подготовка для дальнейшей стилизации через CSS

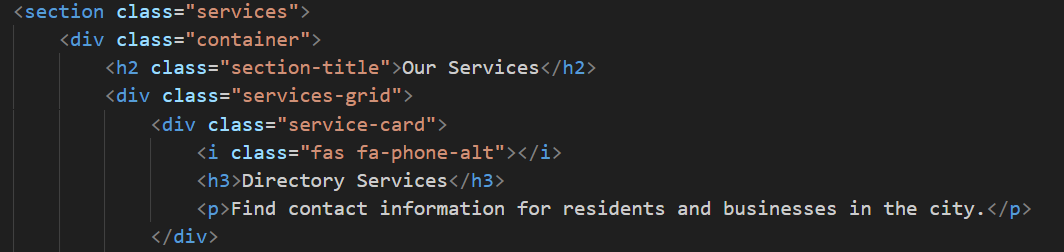
Данная реализация соответствует современным стандартам веб- дизайна и обеспечивает:

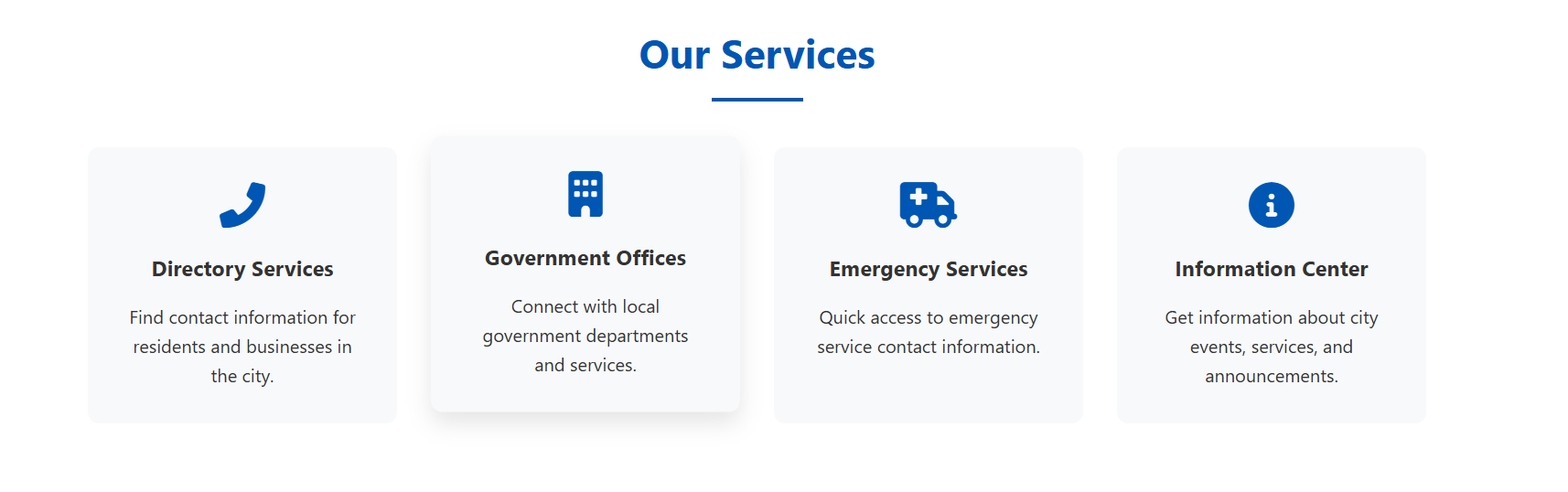
Быстрое понимание назначения сайта

Ясный призыв к действию

Адаптивность (благодаря контейнеру)

Простоту дальнейшего развития





Данный фрагмент кода представляет собой раздел с перечнем услуг телефонной компании. Рассмотрим его ключевые особенности:

1. **Структура секции**

* Секция помечена классом "services" для стилизации
* Используется контейнер (container) для контроля макета
* Заголовок уровня h2 с классом "section-title"
* Сетка услуг (services-grid) для расположения карточек

1. **Карточка услуги**

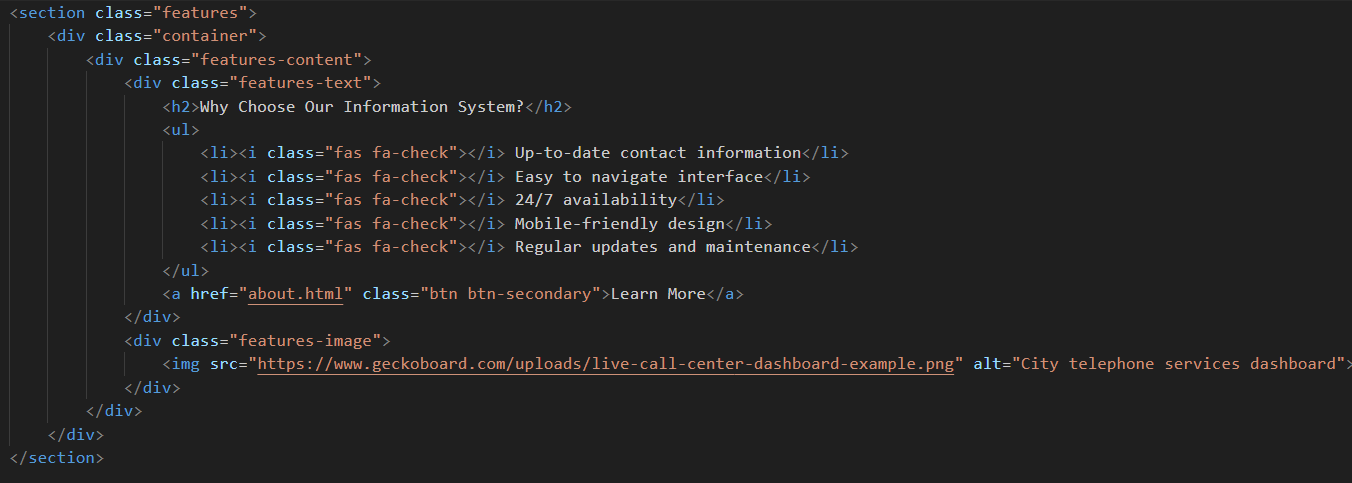
* Отдельный блок service-card для каждой услуги
* Использование иконки из Font Awesome (fa-phone-alt)
* Заголовок уровня h3 с названием услуги
* Описательный абзац (p) с краткой информацией

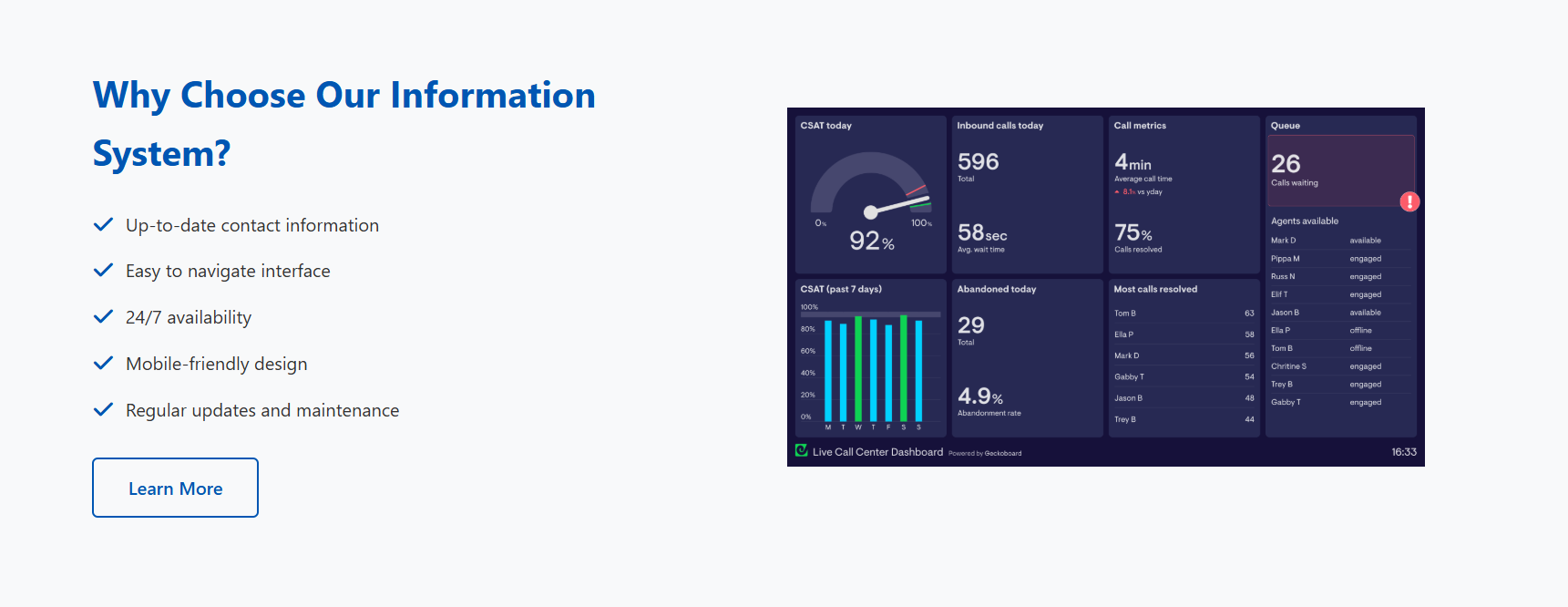
1. **Технические особенности**

* Семантическая разметка (section, h2-h3)
* Использование классов для стилизации
* Иконки подключены через Font Awesome
* Гибкая сетка для расположения карточек

1. **Функциональные преимущества**

* Чёткая визуальная иерархия
* Простое добавление новых карточек услуг
* Поддержка адаптивного дизайна
* Удобство восприятия информации





Данный фрагмент кода представляет собой раздел, демонстрирующий преимущества телефонной информационной системы. Рассмотрим его структуру и функциональные элементы:

1. **Общая структура**

* Секция с классом "features" для стилизации
* Основной контейнер (container) для контроля ширины
* Двухколоночный макет (features-content):
  + Текстовая часть (features-text)
  + Графическая часть (features-image)

1. **Текстовая колонка**

* Заголовок h2 с вопросом-призывом
* Список преимуществ (ul) с иконками-галочками (Font Awesome)
* Кнопка "Learn More" с дополнительным классом btn-secondary
* Ссылка ведет на страницу "О нас" (about.html)

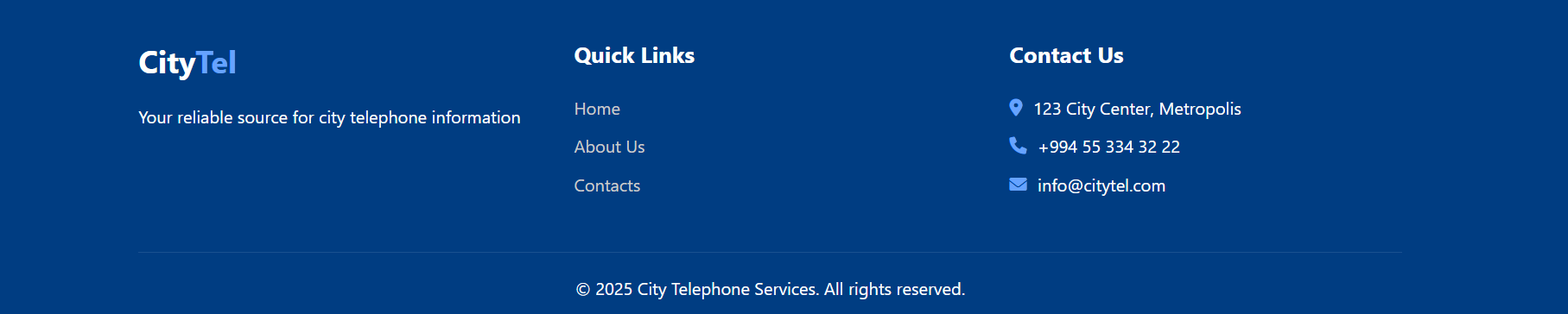
1. **Графическая колонка**

* Изображение дашборда (внешняя ссылка)
* Подпись в атрибуте alt для доступности

1. **Ключевые особенности**

* Четкая визуальная иерархия информации
* Использование иконок для улучшения восприятия
* Двухколоночная адаптивная структура





Данный фрагмент кода представляет собой подвал сайта телефонной компании. Рассмотрим его структуру и функциональные элементы:

1. **Общая структура**

* Основной контейнер footer с классом container
* Трехколоночная структура (footer-content):
  + Логотип и описание (footer-logo)
  + Навигационные ссылки (footer-links)
  + Контактная информация (footer-contact)
* Нижняя часть с копирайтом (footer-bottom)

1. **Блок логотипа**

* Повторение логотипа CityTel с акцентом на "Tel"
* Краткий слоган компании
* Сохранение стилистического единства с шапкой сайта

1. **Навигационные ссылки**

* Заголовок "Quick Links"
* Повторение основных разделов сайта:
  + Главная страница
  + О компании
  + Контакты

1. **Контактная информация**

* Физический адрес с иконкой места
* Телефонный номер с иконкой телефона
* Email с иконкой конверта
* Использование Font Awesome для иконок

1. **Нижняя часть**

* Информация о копирайте
* Год основания (2025)
* Полное название компании

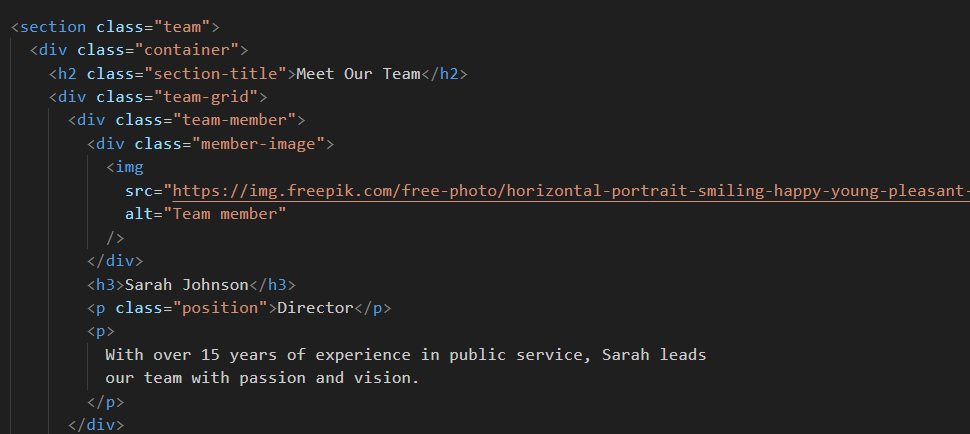
**Технические особенности:**

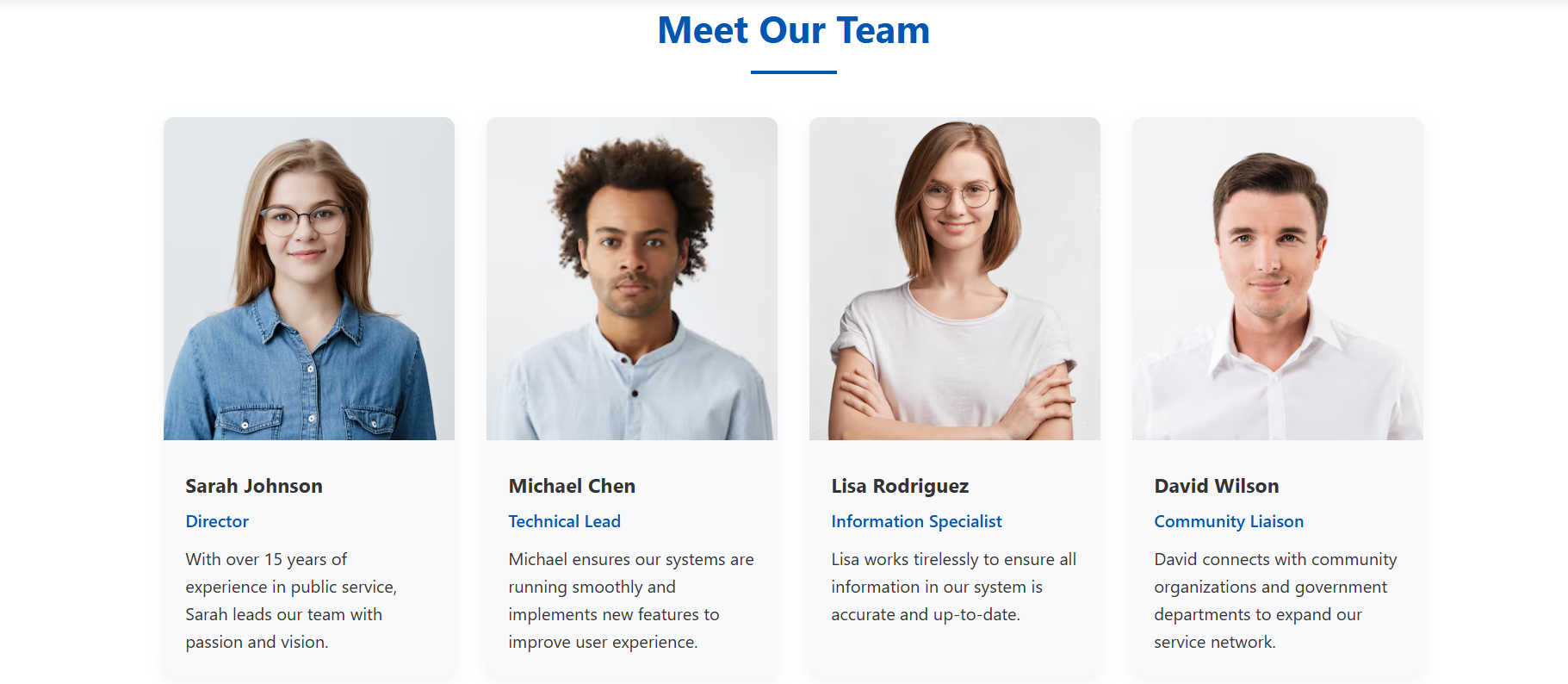
* Семантический тег <footer>
* Логичная группировка информации
* Использование иконок для улучшения UX
* Поддержка принципов доступности
* Четкое визуальное разделение блоков

**Функциональные преимущества:**

1. Единообразие дизайна с основной частью сайта
2. Удобная навигация по ключевым разделам
3. Быстрый доступ к контактным данным
4. Поддержка корпоративного стиля
5. Адаптивность для мобильных устройств

Футер выполняет важные функции по предоставлению пользователям контактной информации и быстрой навигации, завершая страницу в едином стиле с остальными элементами сайта.





Данный фрагмент кода представляет раздел с информацией о сотрудниках компании. Рассмотрим его структуру и особенности:

1. **Общая структура**

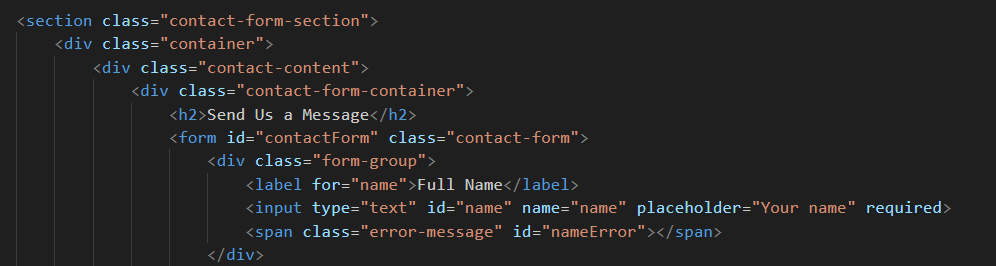
* Секция с классом "team" для стилизации
* Основной контейнер (container) для контроля ширины
* Заголовок h2 с классом "section-title" для единообразия стилей
* Сетка для расположения карточек сотрудников (team-grid)

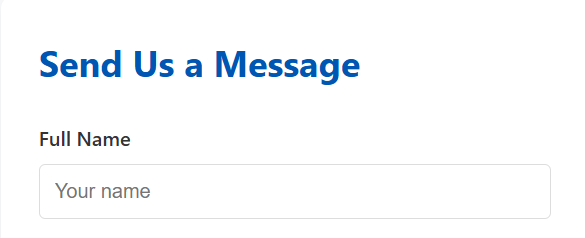
1. **Карточка сотрудника**

* Отдельный блок team-member для каждого участника команды
* Фотография сотрудника (member-image) с внешнего источника
* Заголовок h3 с именем сотрудника
* Параграф с указанием должности (класс position)
* Описание опыта и квалификации

1. **Технические особенности**

* Семантическая разметка (section, h2-h3)
* Использование классов для стилизации
* Внешняя ссылка на изображение (требует проверки доступности)
* Поддержка адаптивной сетки





Данный фрагмент кода представляет собой раздел с контактной формой для отправки сообщений. Разберем его структуру и функциональные элементы:

1. **Общая структура**

* Секция с классом "contact-form-section" для стилизации
* Основной контейнер (container) для контроля ширины
* Блок с контентом (contact-content)
* Контейнер формы (contact-form-container)

1. **Элементы формы**

* Заголовок h2 с призывом к действию
* Форма с ID "contactForm" и классом "contact-form"
* Группа полей ввода (form-group):
  + Поле для имени (name) с:
    - Лейблом (label)
    - Текстовым input
    - Плейсхолдером
    - Обязательным атрибутом required
    - Контейнером для ошибки (error-message)

1. **Технические особенности**

* Семантическая HTML5-разметка
* Использование атрибутов:
  + name - для идентификации данных формы
  + id - для связи label с input
  + required - валидация на стороне клиента
* Подготовка для JavaScript-валидации (error-message)

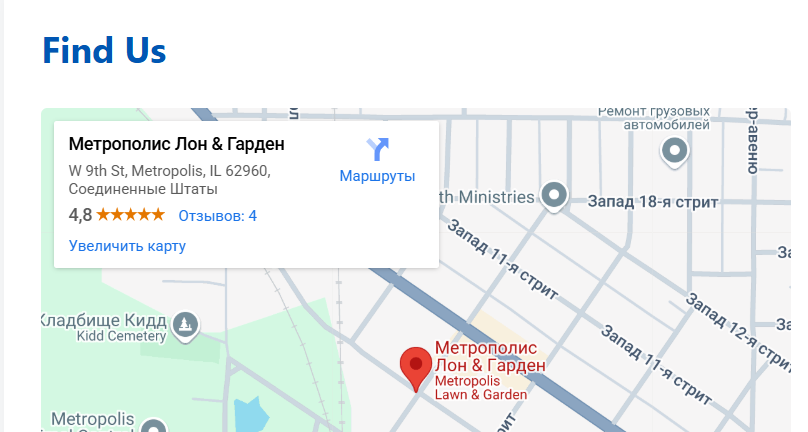




**Особенности реализации:**

* Четкое разделение функционала (отправка) и обратной связи (сообщение)
* Подготовка для JavaScript-обработки:
  + Отправка данных формы
  + Отображение статуса операции





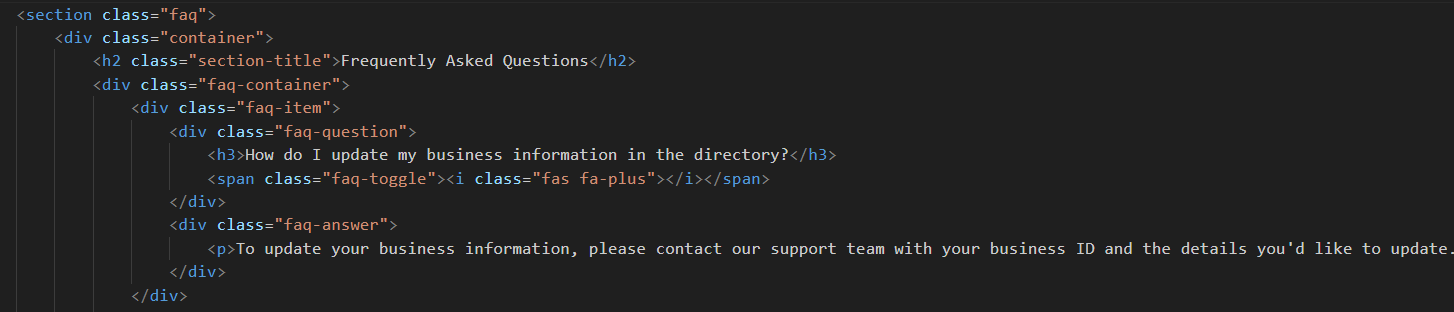
Данный фрагмент кода представляет раздел с картой местоположения компании. Рассмотрим его структуру и функциональные возможности:

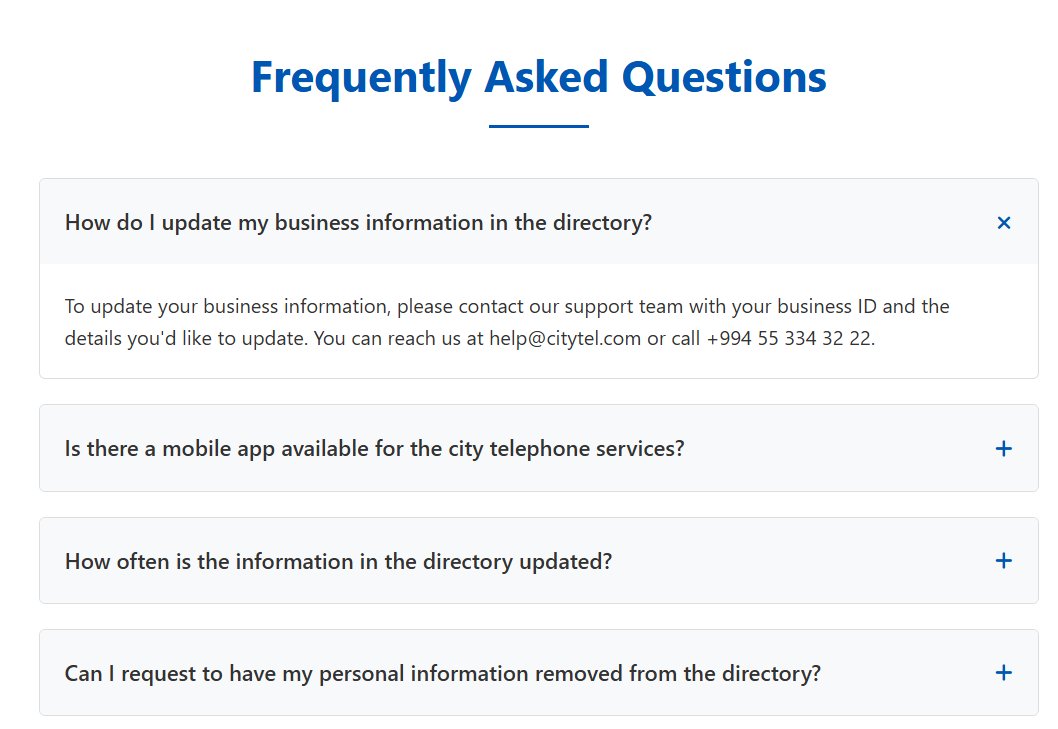
1. **Общая структура**

* Блок contact-map - основной контейнер секции
* Заголовок h2 с текстом "Find Us" (найди нас)
* Контейнер map-container для встраивания карты

1. **Технические особенности**

* Семантическая разметка с четким разделением:
  + Заголовочной части
  + Контейнера для контента





Данный фрагмент кода представляет раздел с часто задаваемыми вопросами. Разберем его структуру и особенности:

1. **Общая структура**

* Секция с классом faq для стилизации
* Контейнер с классом container для контроля ширины
* Заголовок h2 с классом section-title для единообразия стилей
* Основной блок faq-container для всех вопросов

1. **Элемент вопроса-ответа**

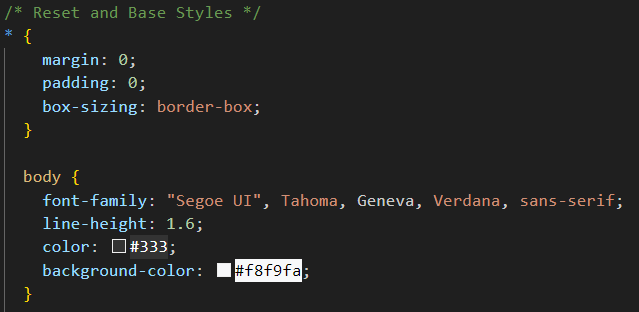
* Отдельный блок faq-item для каждого вопроса
* Часть с вопросом (faq-question):
  + Заголовок h3 с текстом вопроса
  + Иконка "+" из Font Awesome
* Часть с ответом (faq-answer):
  + Текст ответа в параграфе

**Заключение:**

HTML-верстка сайта CityTel построена по современным стандартам. Использование семантической разметки, разбивка на логические блоки и адаптивные элементы обеспечивают удобство восприятия, поддержку мобильных устройств и простоту масштабирования проекта. Такая структура делает сайт понятным для пользователя и легко поддерживаемым в будущем.

**2.4 Использование стилей CSS**

CSS-файл сайта CityTel определяет внешний вид всех страниц: главной, страницы "О нас", "Контакты" и других. Он содержит как общие стили, так и стили, специфичные для различных компонентов пользовательского интерфейса. Далее приводится подробный построчный анализ с пояснениями и вставками соответствующих фрагментов кода.



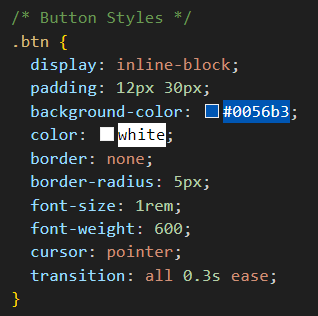
Данный фрагмент содержит начальные стили для веб-страницы. Разберем его назначение и особенности:

1. **Reset стилей**

* Правило \* сбрасывает:
  + Все внешние и внутренние отступы (margin: 0; padding: 0;)
  + Устанавливает box-sizing: border-box для всех элементов (размер включает padding и border)

1. **Базовые стили body**

* Шрифтовая система:
* Межстрочный интервал line-height: 1.6 для удобочитаемости
* Цвет текста #333 (темно-серый)
* Фоновый цвет #f8f9fa (светло-серый)



Данный CSS-код задает базовые стили для кнопок интерфейса. Разберем его свойства и назначение:

1. **Основные параметры**

* display: inline-block - позволяет задавать размеры, сохраняя строчное поведение
* Поля padding: 12px 30px (вертикальные/горизонтальные)
* Цвет фона синий (некорректно указан как [#0056b3])
* Цвет текста белый (некорректно указан как [white])

1. **Детали оформления**

* border: none - убирает стандартную рамку
* border-radius: 5px - скругление углов
* Типографика:
  + Размер шрифта 1rem (≈16px)
  + Полужирное начертание (600)

1. **Интерактивные свойства**

* cursor: pointer - меняет курсор при наведении
* transition: all 0.3s ease - плавные анимации всех свойств



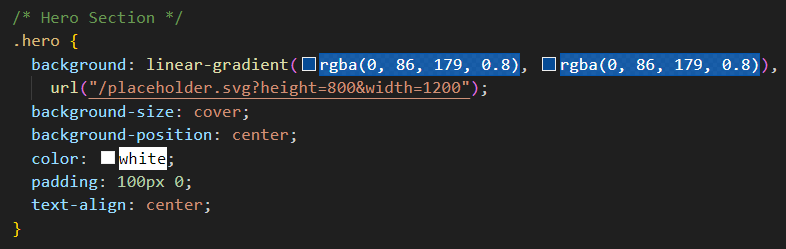
Данный CSS-код задает оформление для верхней навигационной панели сайта. Разберем его ключевые особенности:

1. **Базовые стили header**

* background-color: white - белый фон (некорректный символ ■ вместо #)
* box-shadow - легкая тень снизу (некорректный символ □)
  + Параметры: 2px по вертикали, 10px размытие
  + Цвет: полупрозрачный черный (rgba)
* position: sticky - липкое позиционирование при скролле
* top: 0 - прилипание к верху окна
* z-index: 1000 - гарантированное нахождение поверх других элементов

1. **Стили контейнера**

* Flex-раскладка (display: flex)
* Распределение пространства (justify-content: space-between)
* Вертикальное выравнивание по центру (align-items: center)
* Внутренние отступы 15px со всех сторон



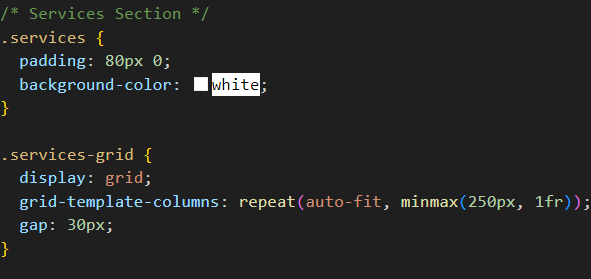
Данный CSS-код задает оформление главной промо-секции сайта. Разберем его особенности:

1. **Фоновое оформление**

* Градиентная подложка с прозрачностью
  + Цвет: синий (#0056b3) с прозрачностью 80%
* Фоновое изображение (заглушка placeholder.svg)
  + Размеры: 1200x800px
* background-size: cover - заполнение всего пространства
* background-position: center - центрирование изображения

1. **Стили содержимого**

* Цвет текста белый
* Отступы: 100px сверху/снизу
* Текст по центру (text-align: center)



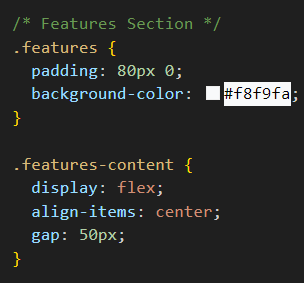
Данный CSS-код задает оформление раздела с услугами компании. Вот его ключевые характеристики:

1. **Общие стили секции**

* Вертикальные отступы 80px (padding: 80px 0)
* Фоновый цвет
* Сеточная система для карточек услуг

1. **Грид-раскладка**

* display: grid - использование CSS Grid
* Адаптивные колонки:
  + repeat(auto-fit, minmax(250px, 1fr))
  + Минимальная ширина колонки 250px
  + Автоматическое количество колонок
* Отступ между элементами 30px (gap: 30px)



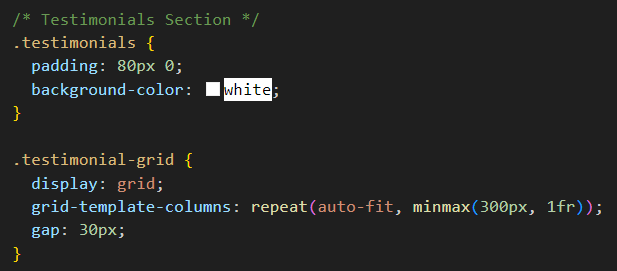
Данный CSS-код задает оформление раздела с ключевыми преимуществами компании. Разберем его структуру и функциональность:

1. **Базовые стили секции**

* Вертикальные отступы 80px (padding: 80px 0)
* Фоновый цвет светло-серый
* Flex-раскладка для содержимого

1. **Расположение контента**

* display: flex - гибкое расположение элементов
* align-items: center - вертикальное выравнивание по центру
* gap: 50px - расстояние между колонками



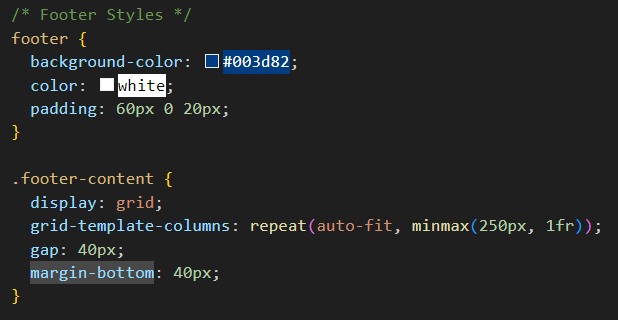
Данный CSS-код задает оформление раздела с отзывами клиентов. Вот его основные характеристики и рекомендации по улучшению:

1. **Базовые стили секции**

* Вертикальные отступы 80px (padding: 80px 0)
* Фоновый цвет белый
* Адаптивная сетка для отзывов

1. **Грид-раскладка**

* display: grid - использование CSS Grid Layout
* Автоматическое количество колонок:
  + Минимальная ширина 300px
  + Максимальная - 1fr (равномерное распределение)
* Отступ между элементами 30px (gap: 30px)



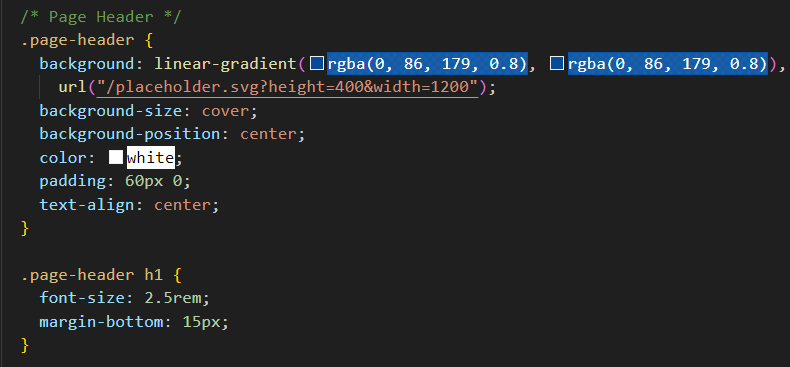
Данный CSS-код задает оформление нижней части веб-страницы. Разберем его особенности и предложим улучшения:

1. **Базовые стили подвала**

* Фоновый цвет: темно-синий (#003d82)
* Цвет текста: белый (white)
* Внутренние отступы: 60px сверху/снизу, 20px по бокам
* Современная сеточная система

1. **Грид-раскладка контента**

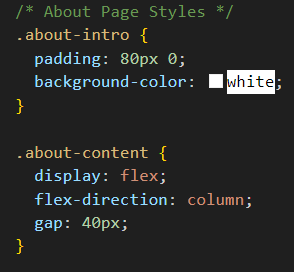
* display: grid - использование CSS Grid
* Адаптивные колонки:
  + Минимальная ширина 250px
  + Автоматическое количество колонок
* Большой отступ между элементами (gap: 40px)
* Отступ снизу 40px перед копирайтом



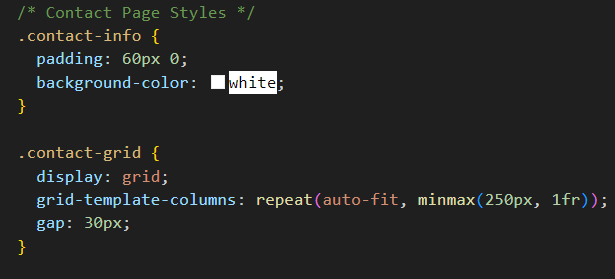
Класс .page-header задает стиль шапки страницы:

* **Фон**: Линейный градиент (linear-gradient) с полупрозрачным синим цветом (rgba(0, 86, 179, 0.8)) накладывается поверх фонового изображения (url), что улучшает читаемость текста.
* **Размер и позиционирование**: background-size: cover и background-position: center обеспечивают адаптивное отображение фона.
* **Текст**: Белый цвет (color: white), центрирование (text-align: center) и отступы (padding: 60px 0).

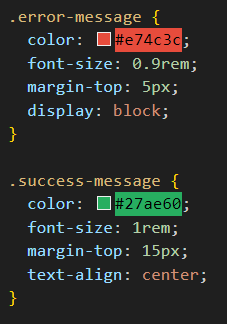
Заголовок (h1) имеет размер 2.5rem и нижний отступ 15px для визуального разделения с другими элементами.



1. .about-intro
   * **Отступы**: Внутренние отступы 80px сверху и снизу (padding: 80px 0).
   * **Фон**: Белый цвет (background-color: white), что обеспечивает нейтральный фон для контента.
2. .about-content
   * **Расположение элементов**: Используется Flexbox (display: flex) с вертикальным направлением (flex-direction: column).
   * **Расстояние между элементами**: Отступ 40px между дочерними элементами (gap: 40px), что улучшает читаемость и структуру.



1. **Секция**.contact-info
   * **Вертикальные отступы**: Внутренние отступы по 60px сверху и снизу (padding: 60px 0) для визуального отделения секции.
   * **Фон**: Чистый белый цвет (background-color: white), обеспечивающий нейтральное и читаемое пространство.
2. **Сетка**.contact-grid
   * **Адаптивная раскладка**: Используется CSS Grid (display: grid) с автоматическим подбором колонок (repeat(auto-fit, minmax(250px, 1fr))).
     + Колонки подстраиваются под ширину экрана, минимальная ширина — 250px, максимальная — 1fr (равномерное распределение).
   * **Промежутки**: Расстояние между элементами сетки — 30px (gap: 30px), что улучшает визуальную организацию.



1. **Стиль ошибок (**.error-message**)**
   * **Цвет текста**: Красный (#e74c3d) для визуального выделения ошибки.
   * **Размер шрифта**: 0.9rem (меньше основного текста для ненавязчивого отображения).
   * **Расположение**: Блочный элемент (display: block) с небольшим отступом сверху (margin-top: 5px).
2. **Стиль успешных сообщений (**.success-message**)**
   * **Цвет текста**: Зеленый (#27ae60) для индикации успешного действия.
   * **Размер шрифта**: 1rem (стандартный, чуть крупнее ошибок).
   * **Расположение**: Центрирование (text-align: center) с увеличенным отступом сверху (margin-top: 15px).



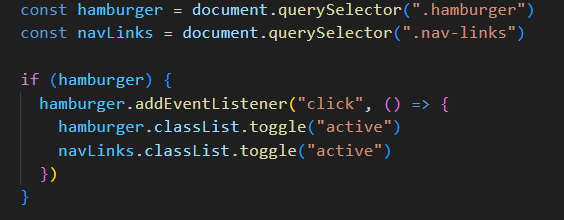
1. **Медиазапрос для экранов ≤ 992px**
   * .features-content: Переключает направление flex-контейнера на вертикальное (flex-direction: column), что обеспечивает линейное расположение элементов на узких экранах.
   * .contact-content: Изменяет сетку на одноколоночную (grid-template-columns: 1fr), упрощая восприятие контента на мобильных устройствах.
2. **Медиазапрос для экранов ≤ 768px**
   * .nav-links:
     + **Скрытое меню**: Позиционируется фиксированно (position: fixed) за пределами экрана (left: -100%).
     + **Анимация**: Плавное появление/скрытие через transition: left 0.3s ease.
     + **Внешний вид**: Занимает всю доступную высоту (height: calc(100vh - 70px)) и ширину (width: 100%), белый фон (background-color: white), вертикальное центрированное расположение (flex-direction: column, align-items/justify-content: center).

**Заключение:**

CSS-оформление сайта CityTel демонстрирует аккуратный, современный и отзывчивый интерфейс, построенный на основе flex-разметки, плавных анимаций и адаптивной верстки. Использование единых цветовых решений и стилистической гармонии делает интерфейс удобным и визуально приятным.

Таким образом, стили CSS обеспечивают не только эстетическую привлекательность, но и функциональность интерфейса сайта, делая его доступным на различных устройствах и понятным для пользователей.

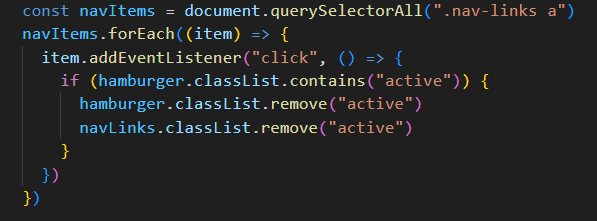
* 1. **Использование JS**



Скрипт реализует функционал "бургер"-меню для адаптивной навигации, позволяя открывать/закрывать меню при клике на иконку.

Ключевые элементы:

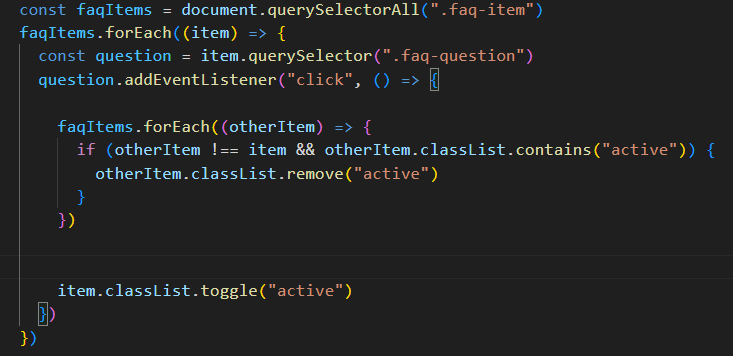
1. Выбор элементов DOM:
   * hamburger — иконка меню (элемент с классом .hamburger).
   * navlinks — блок навигационных ссылок (элемент с классом .nav-links).
2. Логика работы:
   * При клике на hamburger:
     + Добавляется/удаляется класс active для иконки (hamburger), что может менять её стиль (например, превращать в крестик).
     + Добавляется/удаляется класс active для меню (navlinks), что изменяет его видимость (например, через CSS-свойство left: 0).



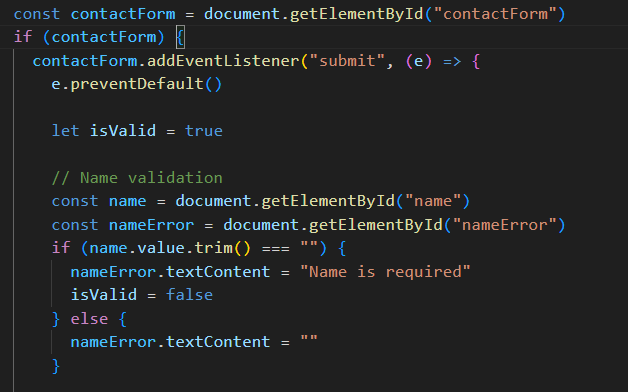
Скрипт автоматически закрывает мобильное меню при клике на любую из навигационных ссылок, улучшая пользовательский опыт на малых экранах.

**Ключевые элементы:**

1. **Выбор элементов:**
   * navItems — коллекция всех ссылок внутри .nav-links (используется querySelectorAll).
   * hamburger и navlinks — глобальные переменные (должны быть объявлены ранее, как в предыдущем скрипте).
2. **Логика работы:**
   * Для каждой ссылки (item) добавляется обработчик клика.
   * При клике проверяется, есть ли класс active у hamburger (т.е. открыто ли меню).
   * Если меню открыто, класс active удаляется у обоих элементов (hamburger и navlinks), что скрывает меню.



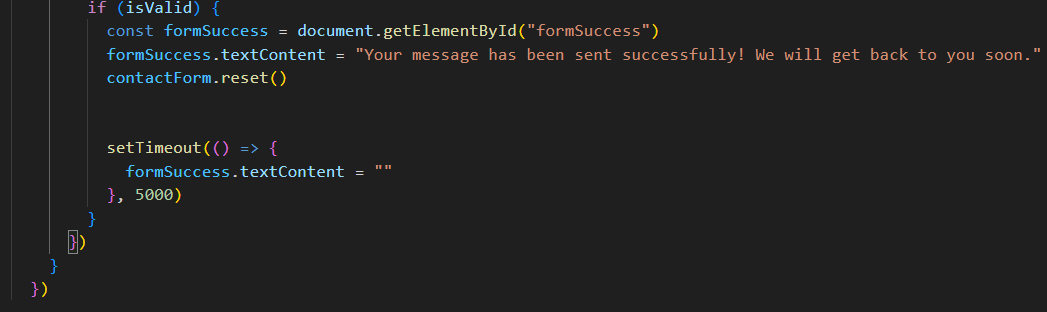
Скрипт реализует аккордеон для FAQ-секции, где при клике на вопрос автоматически закрываются все остальные открытые вопросы и переключается состояние текущего (открывается/закрывается). Для работы необходимо наличие соответствующих CSS-стилей, которые управляют отображением контента при добавлении/удалении класса active у элементов.

****

Скрипт выполняет валидацию формы контактов, проверяя заполнение обязательных полей перед отправкой.

**Логика работы:**

1. **Отмена стандартной отправки формы**
   * Перехватывает событие submit через e.preventDefault().
2. **Валидация поля имени**
   * Проверяет, не пустое ли значение в поле name (с удалением пробелов через trim()).
   * Если поле пустое:
     + Выводит сообщение об ошибке "Name is required".
     + Устанавливает флаг isValid = false.
   * Если поле заполнено — очищает ошибку.

****

Код обрабатывает успешную отправку контактной формы, выводя сообщение об успехе и сбрасывая поля формы.

**Логика работы:**

1. **Проверка валидности формы**
   * Блок выполняется только если isValid === true.
2. **Действия при успешной отправке:**
   * Выводит сообщение "Your message has been sent successfully!..." в элемент formSuccess.
   * Сбрасывает все поля формы через contactForm.reset().
   * Автоматически очищает сообщение через 5 секунд (5000 мс) с помощью setTimeout.

**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://kursovaya-for-leyla.vercel.app/>

**4****. ЛИТЕРАТУРА**

Книги:

1. «*HTML и CSS. Разработка и дизайн веб-сайтов*» — Джон Дакетт  
   *(Лучшая книга для начинающих, много визуальных примеров)*
2. «*JavaScript и jQuery. Интерактивная веб-разработка*» — Джон Дакетт  
   *(Продолжение первой книги, посвящено JS и jQuery)*
3. «*Вы не знаете JS*» (серия книг) — Кайл Симпсон  
   *(Глубокое погружение в JavaScript, бесплатно на GitHub)*
4. «*CSS. Карманный справочник*» — Эрик Мейер  
   *(Удобный справочник по свойствам CSS)*
5. «*JavaScript. Подробное руководство*» — Дэвид Флэнаган  
   *(Полное руководство по JS, подходит для продвинутых)*

**5.ПРИЛОЖЕНИЯ**

Репозиторий:

<https://github.com/shalbuzz/Kursovaya_for_Leyla.git>

Страница сайта на хостинге GitHub Pages:

<https://kursovaya-for-leyla.vercel.app/>