**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**Факультет: Информационные технологии и управление**

**Кафедра: Компьютерная инженерия**

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

**Курсовая работа**

**Тема** «Разработка системы онлайн-бронирования отелей»

**Группа:** 680.22

**Курc:** 3

**Специальность:** Информационные технологии

**Код специальности:** 050616

**Студент**: Мургузов Турал

**Руководитель:** **преп. 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 Мургузов Турал**

**Зачетная книжка**

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

**Срок выдачи**

**Дата сдачи**

**Тема курсовой работы «Разработка системы онлайн-**

**бронирования отелей»**

**Отзыв руководителя курсовой:**

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

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

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

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

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

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

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

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

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

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

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

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ 4**

1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6**
   1. **Web-программирование 6**
   2. **HTML и XHTML 6**
   3. **Dynamic HTML 7**

**1.3.1 Каскадные таблицы стилей 8**

**1.3.2 JavaScript 10**

**1.4 Bootstrap 10**

1. **ПРАКТИЧЕСКАЯ ЧАСТЬ 12**
   1. **Описание предметной области 12**
   2. **Создание окружения для проекта 12**
   3. **Верстка HTML страницы 12**
   4. **Использование стилей CSS 29**
   5. **Использование JS 32**

**3. РЕЗУЛЬТАТ 37**

**4. ЛИТЕРАТУРА 38**

**5.ПРИЛОЖЕНИЯ 39**

**ВВЕДЕНИЕ**

Современные технологии веб-разработки открывают широкие возможности для создания удобных и функциональных систем онлайн-бронирования. Такие системы стали неотъемлемой частью гостиничного бизнеса, позволяя клиентам легко находить и бронировать номера, а владельцам отелей - эффективно управлять своими ресурсами.

Основу системы составляют два ключевых компонента: клиентская часть (frontend) и серверная часть (backend). Для frontend-разработки используются HTML5, CSS3 и JavaScript с фреймворком React.js, что обеспечивает создание интерактивного и адаптивного интерфейса. Серверная часть реализована на Node.js с Express.js, что гарантирует быструю обработку запросов, а для хранения данных применяется надежная СУБД PostgreSQL.

Система предлагает пользователям широкий функционал. Модуль поиска позволяет фильтровать отели по датам, местоположению, категории и стоимости. Процесс бронирования включает выбор типа номера, указание дополнительных услуг и подтверждение резерва. Для зарегистрированных пользователей доступен личный кабинет с историей бронирований и возможностью отмены. Административная часть системы предоставляет инструменты для управления номерами, обработки заявок и анализа загрузки отеля.

Разработка велась поэтапно, начиная с глубокого анализа требований и проектирования интерфейсов, через реализацию всех компонентов системы, до тщательного тестирования функциональности и безопасности. Особое внимание уделялось простоте использования, скорости работы поискового механизма и защите перданных пользователей.

В результате создана современная, надежная и масштабируемая система бронирования, которая не только упрощает процесс поиска и бронирования номеров для клиентов, но и предоставляет владельцам отелей мощный инструмент для управления бизнес-процессами и повышения эффективности работы. Интеграция с внешними сервисами (платежными системами, картами) делает систему еще более удобной и функциональной для всех участников процесса.

**1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1** **Web-программирование**

Современные технологии веб-разработки позволяют создавать удобные и функциональные системы онлайн-бронирования. Такие системы стали неотъемлемой частью многих сфер бизнеса, включая гостиничный бизнес, рестораны, авиаперевозки и другие услуги. Актуальность данной темы обусловлена растущим спросом на автоматизацию процессов бронирования и необходимостью предоставления пользователям удобного интерфейса для взаимодействия с сервисами.

**Основные технологии разработки**

Для создания системы онлайн-бронирования используются как клиентские, так и серверные технологии. На стороне клиента применяются HTML, CSS и JavaScript, которые обеспечивают создание интерактивного и адаптивного интерфейса. Популярные фреймворки, такие как React или Vue.js, значительно ускоряют разработку и улучшают пользовательский опыт.

Серверная часть системы строится на таких технологиях, как Node.js, PHP или Python, которые обрабатывают запросы пользователей и взаимодействуют с базой данных. Для хранения информации используются реляционные (MySQL, PostgreSQL) или NoSQL (MongoDB) базы данных, в зависимости от требований проекта.

**1.2****HTML**

HTML (HyperText Markup Language) является основным языком разметки для создания структуры веб-страниц в системе онлайн-бронирования. Этот стандартизированный язык обеспечивает базовый каркас, который затем наполняется стилями CSS и интерактивными элементами JavaScript.

Основой любой HTML-страницы служит документ с расширением .html, начинающийся с обязательного объявления типа документа !DOCTYPE html. Внутри тегов <html> располагается вся структура страницы, разделённая на два основных раздела: <head> и <body>.

В разделе <head> размещается метаинформация:

* Указание кодировки символов через <meta charset="UTF-8">
* Теги viewport для адаптивности под мобильные устройства
* Заголовок страницы <title>, отображаемый во вкладке браузера
* Подключение внешних ресурсов (CSS-стилей, шрифтов)

Основное содержимое системы бронирования располагается в теле документа - между тегами <body>. Типичная структура включает:

1. Шапку сайта (<header>), содержащую логотип, навигационное меню (<nav>) и основные ссылки
2. Главный контент (<main>), разделённый на смысловые секции (<section>)
3. Подвал (<footer>) с контактной информацией и дополнительными ссылками

Для форм бронирования используются специализированные элементы:

* Контейнер формы <form> с указанием метода передачи данных (GET/POST)
* Поля ввода <input> различных типов (text, date, number)
* Выпадающие списки <select> с вариантами выбора <option>
* Чекбоксы <input type="checkbox"> для дополнительных услуг
* Кнопки отправки данных <button type="submit">

Семантическая разметка с использованием тегов <article>, <section>, <aside> улучшает доступность системы и помогает поисковым системам лучше понимать структуру контента. Для отображения результатов поиска применяются карточки, организованные через <div> или <article> с чёткой иерархией заголовков (<h2>-<h5>).

Современный HTML5 предоставляет множество возможностей для создания интуитивного интерфейса системы бронирования, включая встроенную валидацию форм, мультимедийные элементы и API для интеграции с другими веб-технологиями.

**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***

Каскадные таблицы стилей (CSS — *Cascading Style Sheets*) — это язык стилей, предназначенный для описания внешнего вида HTML-документов. CSS позволяет задавать оформление элементов веб-страницы, включая шрифты, цвета, отступы, размеры, расположение и поведение блоков при взаимодействии с пользователем. Использование CSS позволяет разработчику отделить логику структуры страницы (HTML) от её визуального представления, тем самым обеспечивая более гибкую и масштабируемую разработку.

В контексте веб-приложений, таких как **система онлайн-бронирования отелей**, CSS является неотъемлемой частью пользовательского интерфейса. Он обеспечивает визуальную привлекательность сайта, помогает создать единый стиль оформления всех страниц (главной, страницы отелей, бронирования, контактов и т. д.) и значительно улучшает пользовательский опыт.

**Основные возможности CSS:**

* **Оформление текста:** шрифты, размеры, цвета, выравнивание, межстрочный интервал.
* **Оформление блоков:** границы, фоны, тени, отступы, рамки.
* **Размещение элементов на странице:** блочная и строчная модель, flexbox, grid.
* **Визуальные эффекты:** анимации, плавные переходы, псевдоклассы (:hover, :focus).
* **Адаптивность:** медиазапросы (@media) позволяют изменять стиль сайта под разные разрешения экранов — от смартфонов до больших мониторов.

**Принцип каскадности**

CSS использует механизм каскадности, при котором несколько стилей могут применяться к одному элементу. В случае конфликта приоритет определяется по следующим критериям:

1. **Специфичность селектора:** селекторы по id имеют больший приоритет, чем по классу, а те — выше, чем по тегу.
2. **Порядок подключения:** если два одинаковых по специфичности правила конфликтуют, применяется последнее по порядку.
3. **Важно-правила (!important)** — переопределяют все другие, за исключением других !important с более высокой специфичностью.

**Типы подключения CSS:**

1. **Встроенные стили** — задаются в атрибуте style конкретного HTML-элемента.
2. **Внутренние стили** — размещаются в <style> внутри HTML-документа.
3. **Внешние таблицы стилей** — подключаются с помощью тега <link> и используются наиболее часто, так как обеспечивают переиспользуемость и удобство поддержки.

**Пример подключения внешнего файла:**

html

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

**Роль CSS в системе онлайн-бронирования**

В системе онлайн-бронирования отелей CSS позволяет:

* создать приятный интерфейс для пользователей;
* выделить важные элементы (например, кнопки «Забронировать» или формы поиска);
* организовать структуру элементов страницы — карточки отелей, формы, навигационное меню;
* обеспечить адаптивную верстку, чтобы сайт корректно отображался на всех устройствах;
* сделать сайт более профессиональным и современным, что повышает доверие пользователей.

***1.3.2 JavaScript***

JavaScript (JS) — это язык программирования, предназначенный для создания интерактивных и динамических элементов веб-страниц. В отличие от HTML и CSS, которые отвечают за структуру и стиль, JavaScript обеспечивает поведение и взаимодействие пользователя с веб-интерфейсом.

В системе онлайн-бронирования отелей JavaScript играет важную роль, так как позволяет реализовать функции, улучшающие удобство и эффективность взаимодействия с сайтом. С его помощью можно обрабатывать действия пользователя в реальном времени, без необходимости перезагрузки страницы. Это особенно важно для бронирования, где требуется высокая скорость отклика и точность ввода данных.

Одна из ключевых задач JavaScript в данном проекте — валидация форм. Он позволяет проверить правильность заполнения полей до их отправки на сервер, что снижает количество ошибок и повышает надёжность системы. Кроме того, JavaScript отвечает за отображение уведомлений, подсказок и предупреждений, что делает интерфейс сайта более информативным и удобным.

Также JavaScript используется для управления элементами интерфейса, такими как мобильное меню, выпадающие списки, всплывающие окна и аккордеоны с часто задаваемыми вопросами. Это повышает интерактивность сайта и делает навигацию более интуитивной.

Дополнительной важной функцией JavaScript является возможность асинхронной работы с сервером. Благодаря этому сайт может отправлять и получать данные без перезагрузки страницы. Например, можно динамически загружать список доступных номеров, фильтровать отели по параметрам или обновлять стоимость бронирования в зависимости от выбора пользователя.

Таким образом, JavaScript обеспечивает современный уровень функциональности веб-приложения, делает систему бронирования более динамичной, удобной и ориентированной на пользователя.

**1.4 Bootstrap**

Bootstrap — это популярный фреймворк для разработки адаптивных и кросс браузерных веб-интерфейсов. Он представляет собой набор готовых компонентов, стилей и JavaScript-скриптов, который значительно упрощает и ускоряет процесс верстки веб-страниц. Bootstrap позволяет быстро создавать современные и функциональные пользовательские интерфейсы, не прибегая к написанию большого количества собственного CSS и JavaScript-кода.

В системе онлайн-бронирования отелей Bootstrap используется для организации визуального оформления сайта, построения сетки, размещения контента и обеспечения адаптивности. Он помогает легко выстраивать макет страниц — от навигационных панелей и карточек отелей до форм бронирования и кнопок действий.

Одним из ключевых преимуществ Bootstrap является встроенная система сеток, которая позволяет размещать элементы на странице в несколько колонок, автоматически подстраивая их под ширину экрана. Это особенно важно для адаптивного дизайна, чтобы сайт корректно отображался как на компьютерах, так и на мобильных устройствах.

Кроме того, Bootstrap предоставляет обширную библиотеку компонентов, таких как модальные окна, навигационные меню, вкладки, алерты и карусели. Использование этих компонентов повышает интерактивность сайта и делает интерфейс более современным и удобным для пользователя.

Также важно отметить, что Bootstrap поддерживает единый стиль оформления. Благодаря этому все элементы интерфейса выглядят согласованно и профессионально, что положительно влияет на восприятие сайта потенциальными клиентами.

Применение Bootstrap в рамках разработки системы бронирования отелей позволяет существенно сократить время верстки, повысить качество интерфейса и обеспечить стабильную работу на всех устройствах. Это делает фреймворк эффективным инструментом для реализации веб-проекта с минимальными затратами и максимальным удобством.

**2. ПРАКТИЧЕСКАЯ ЧАСТЬ.**

**2.1 Описание предметной области**

В данной курсовой работе рассматривается создание веб-страницы для системы гостиничного бронирования, которая позволяет пользователям искать и бронировать номера в отелях через интернет. Основной целью проекта является создание интерфейса, который был бы прост в использовании и обеспечивал бы удобство взаимодействия с системой для пользователей. В рамках веб-страницы предусмотрены такие разделы, как домашняя страница, информация о компании, контактная информация и форма для отправки сообщений. Кроме того, разработан раздел с часто задаваемыми вопросами и карта, показывающая местоположение компании.

**2.2 Создание окружения для проекта**

Проект был разработан с использованием стандартных технологий веб-разработки: HTML, CSS и JavaScript. Для создания интерфейса страницы была использована библиотека Bootstrap, которая предоставляет готовые стили и компоненты, упрощающие процесс верстки. Также для отображения иконок были использованы шрифты Font Awesome. В проекте задействован файл стилей (styles.css) для кастомизации внешнего вида элементов и создания уникального дизайна страницы.

**2.3 Верстка HTML страницы**

HTML-код страницы представляет собой структуру веб-страницы, разделенную на несколько основных частей: заголовок страницы, главное меню навигации, форму обратной связи и информационные блоки. Код включает в себя следующие основные элементы: <nav> для создания навигационного меню, <section> для разделов контента<form> для формы обратной связи и различные контейнеры для группировки информации. Страница включает в себя также раздел с картой, что позволяет пользователям удобно найти расположение компании.

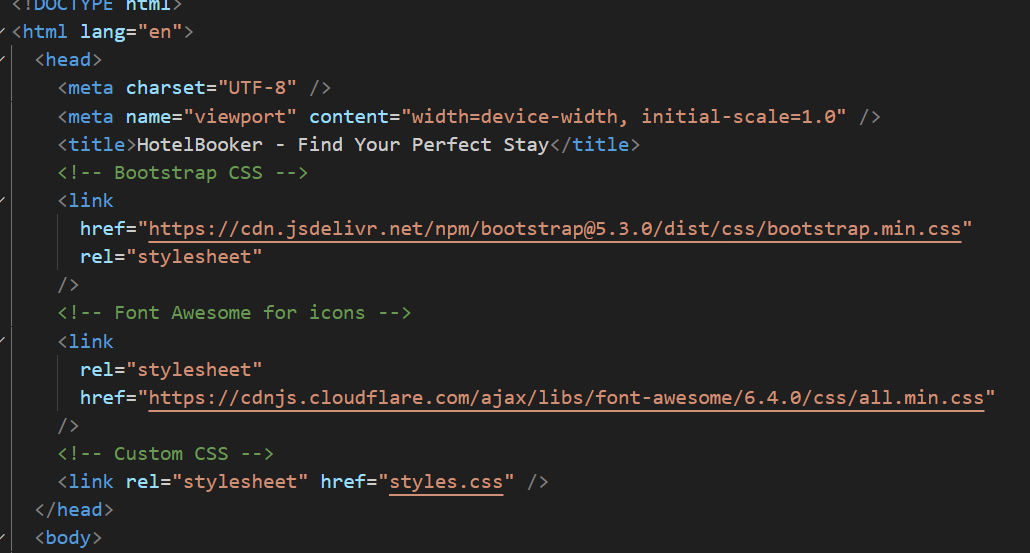


Рис. 2.1. Структура веб-страницы

В результате получается (рис. 2.2)

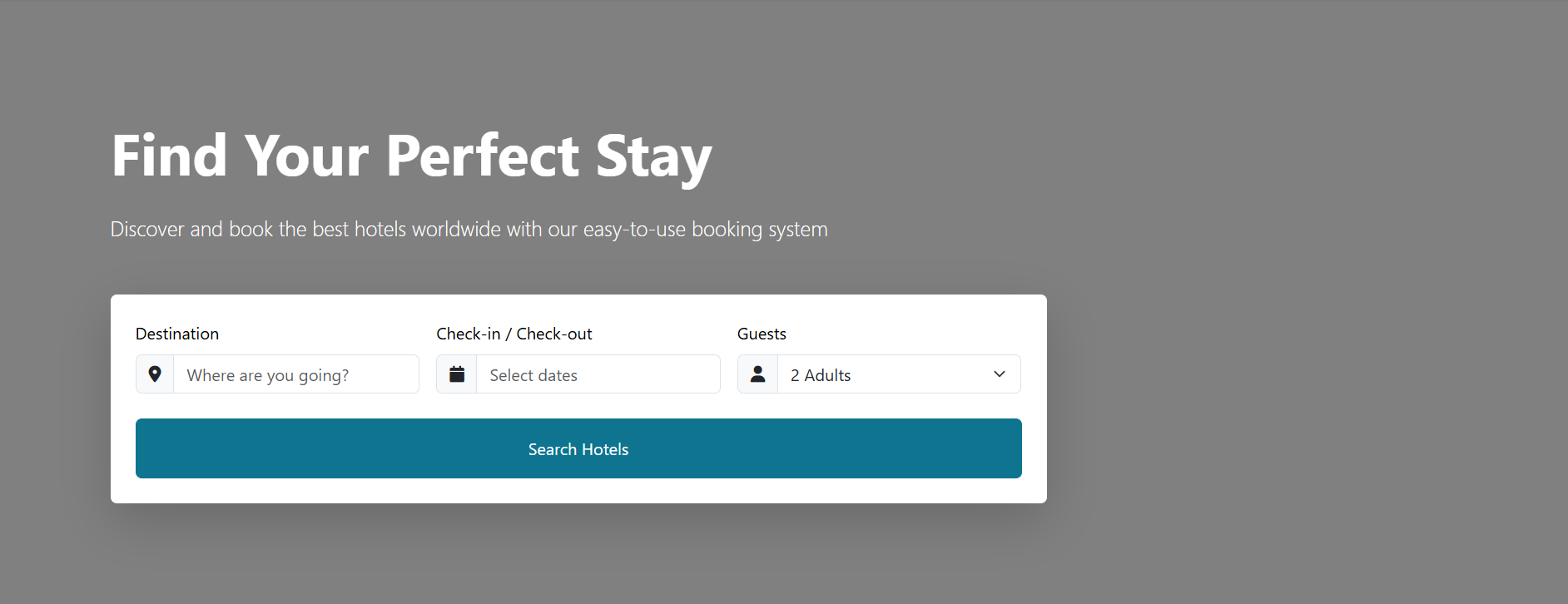


Рис. 2.2

Фрагмент кода HTML-документа для веб-приложения "HotelBooker"

Данный код (рис.2.1) представляет собой начальную часть HTML-разметки веб-приложения для поиска отелей. В нём реализованы следующие элементы:

1. Базовые мета-теги:

Указана кодировка UTF-8 для корректного отображения текста.

Использован viewport для адаптивности на мобильных устройствах.

1. Подключение внешних библиотек:

Bootstrap 5 (через CDN) — для быстрого создания адаптивного интерфейса.

Font Awesome (через CDN) — для добавления иконок.

Кастомный CSS-файл (styles.css) — для индивидуального оформления элементов.



Рис. 2.3



Рис. 2.4. Навигационная панель

Этот код (рис. 2.3) создаёт навигационную панель (Navbar) для веб-приложения "HotelBooker" с использованием Bootstrap 5. В результате получаем Рис. 2.4.

Основные элементы:

1. Логотип и название (HotelBooker) — ссылка на главную страницу.
2. Адаптивное меню (раскрывается на мобильных устройствах).
3. Ссылки на разделы:
   * Home (главная),
   * About (о проекте),
   * Contact (контакты).
4. Кнопки авторизации (Sign In и Sign Up).

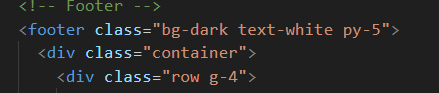


Рис. 2.5. Фрагмент кода футера веб-приложения “HotelBooker”

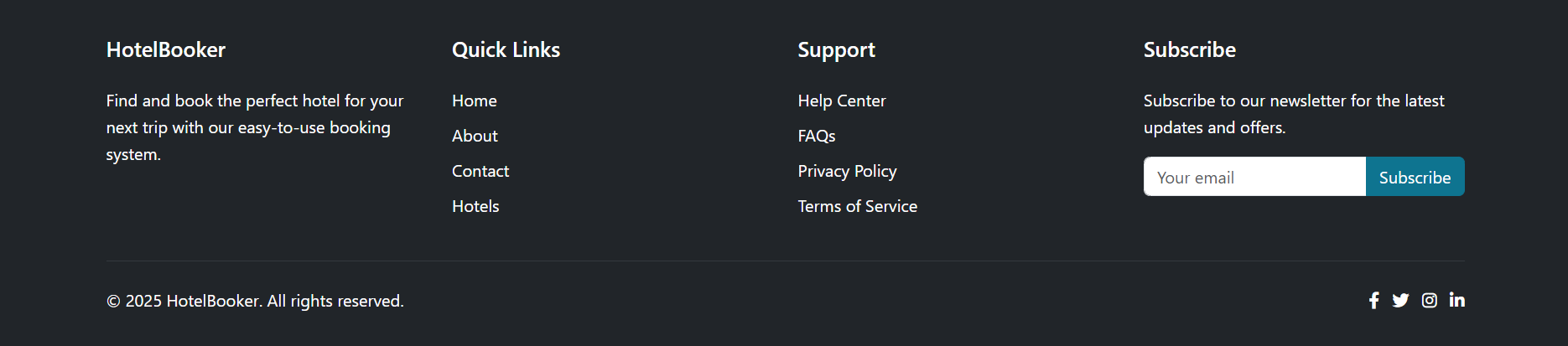


Рис. 2.6. “HotelBooker”

Данный код (рис. 2.5) представляет собой начальную часть реализации подвала (футера) сайта с использованием фреймворка Bootstrap 5. В результате получаем Рис. 2.6. В коде применяются следующие основные технологии и подходы:

1. Семантическая верстка:

Использован тег <footer>, что соответствует современным стандартам семантической HTML-разметки

Применены классы Bootstrap для быстрой стилизации

1. Стилевое оформление:

Фон темного цвета (класс bg-dark)

Текст белого цвета (класс text-white)

Значительные вертикальные отступы (класс py-5 добавляет padding сверху и снизу в 3rem)

1. Адаптивная сетка:

Использован контейнер (container) для ограничения ширины контента

Применена система строк (row) и колонок Bootstrap

Указан промежуток между колонками (g-4 - gutter размером 1.5rem)

1. Типичное содержимое футера:

Контактная информация

Навигационные ссылки

Социальные сети

Форма подписки

Копирайт и юридическая информация

Данная реализация демонстрирует:

Следование принципам адаптивного дизайна

Использование возможностей фреймворка Bootstrap

Применение современных подходов к верстке

Внимание к визуальному оформлению интерфейса

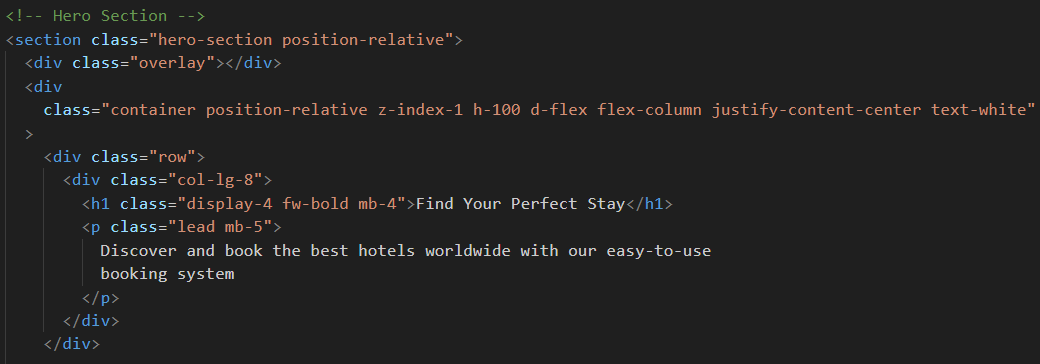


Рис. 2.7. Герой-секция для HotelBooker

**Код (рис. 2.7) создает главный баннер (Hero Section) с:**

Заголовком (Find Your Perfect Stay)

Подзаголовком с описанием сервиса

Адаптивным дизайном (Bootstrap):

Центрирование контента по вертикали

Темный оверлей для улучшения читаемости

Отзывчивая сетка (col-lg-8)

**Функционал:**

Привлекает внимание пользователя

Четко передает цель сервиса

Готов к добавлению кнопок CTA

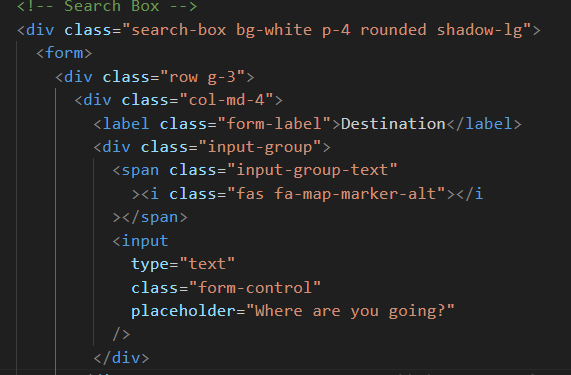


Рис. 2.8. Код поисковой формы

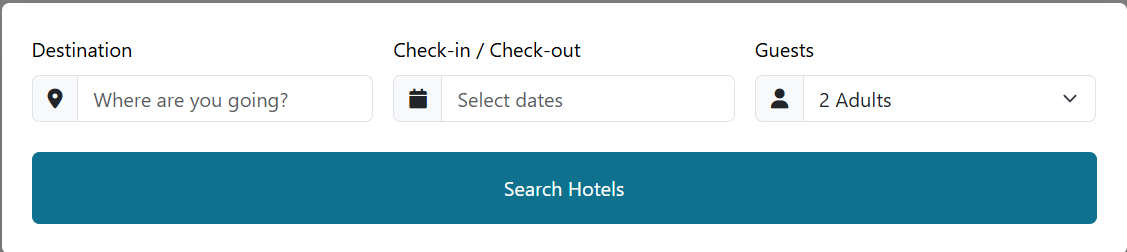


Рис. 2.9

Поисковая форма для HotelBooker (рис. 2.9)

Код (рис. 2.8) реализует интерактивный блок поиска отелей с:

1. Визуальными элементами:

Белый фон с закругленными углами (bg-white rounded)

Тень для глубины (shadow-lg)

Иконка локации (Font Awesome)

2. Функциональными полями:

Поле "Destination" с:

Лейблом

Иконкой-маркером

Инпут-полем (текстовый тип)

Плейсхолдером-подсказкой

3. Техническими особенностями:

Адаптивная сетка Bootstrap (row g-3, col-md-4)

Группа ввода (input-group)

Отступы (p-4 для padding, g-3 для промежутков)

Напишем код для секции "Featured Hotels" (рис. 2.10)

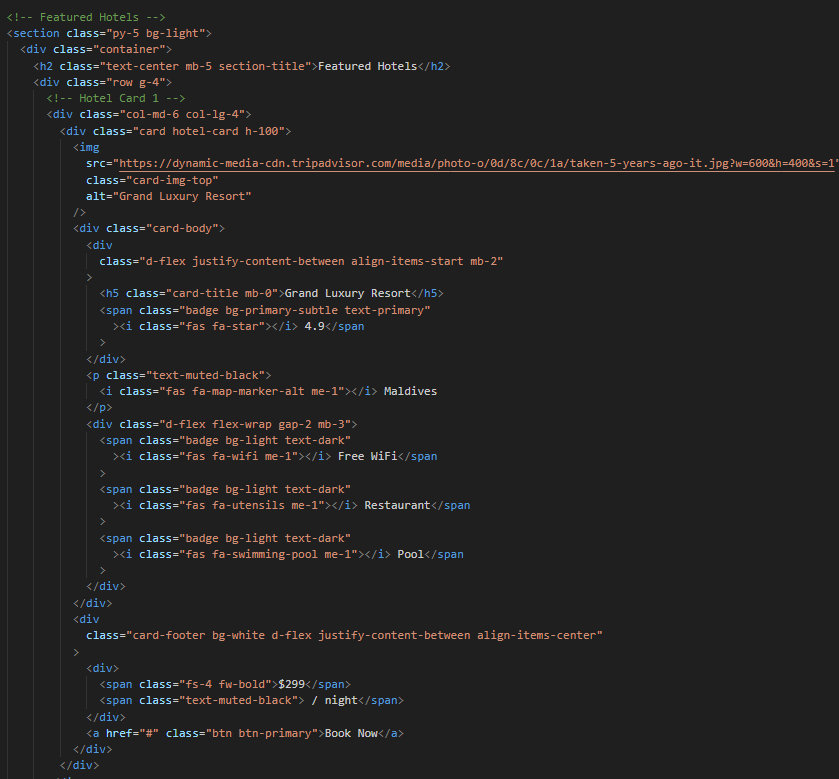


Рис. 2.10. Секция "Featured Hotels"

И в результате получим (рис. 2.11):

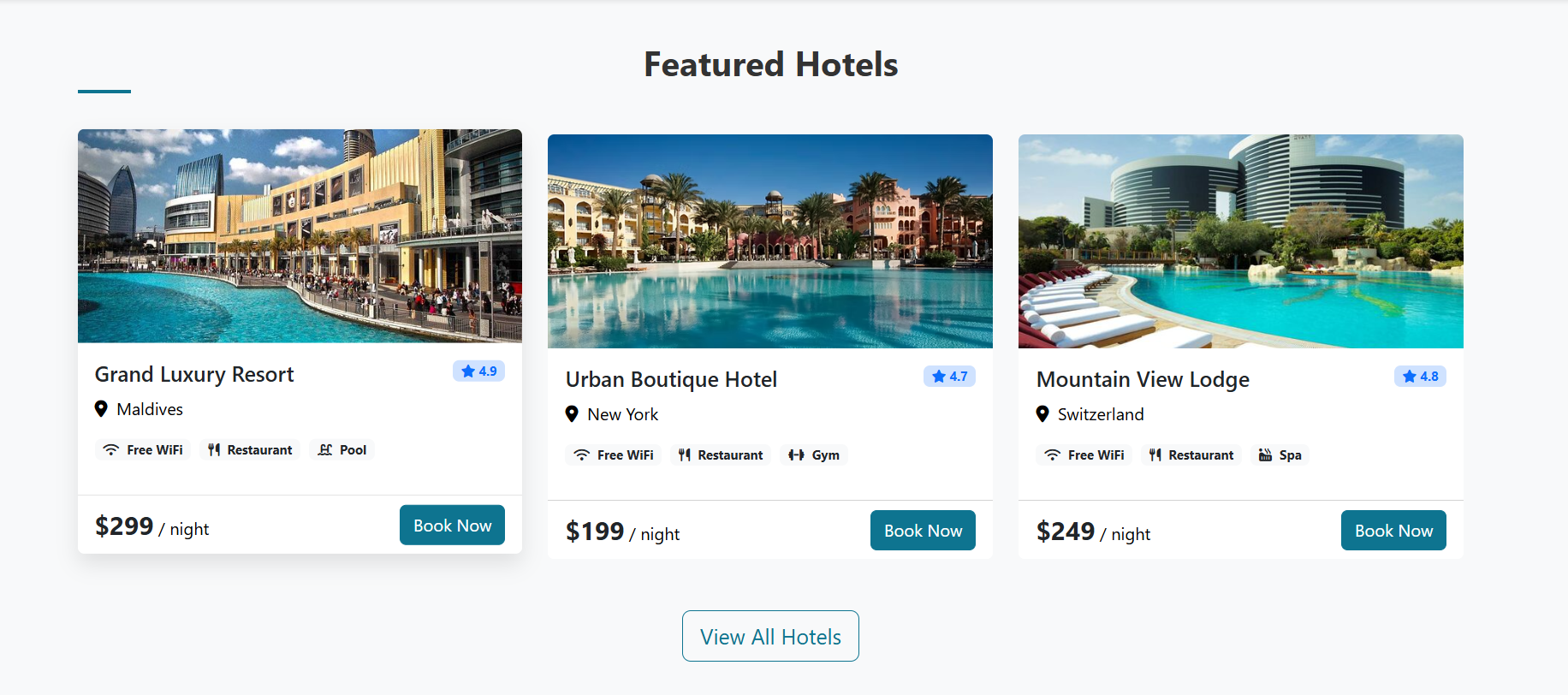


Рис. 2.11.

Технический разбор секции "Featured Hotels"

1. Сетка и адаптивность:

Используется система сетки Bootstrap (row + col-md-6 col-lg-4)

Карточки занимают:

100% ширины на мобильных (по умолчанию)

50% на планшетах (col-md-6)

33% на десктопах (col-lg-4)

* + Гуттеры g-4 создают фиксированные промежутки между карточками

1. Карточка отеля:

Класс h-100 делает высоту карточек одинаковой в строке

Состоит из:

Верхнего изображения (card-img-top)

Основного контента (card-body)

Футера (card-footer)

1. Элементы интерфейса:

Рейтинг реализован через badge bg-primary-subtle

Удобства отображаются как badges с иконками

Гибкое позиционирование через Flexbox:

d-flex justify-content-between - выравнивание по краям.

align-items-center - вертикальное центрирование.

1. Типографика:
2. Заголовок карточки (card-title) с mb-0 убирает нижний отступ

Цена выделена крупным шрифтом (fs-4) и жирным начертанием (fw-bold)

1. Особенности верстки:

Иконки Font Awesome встраиваются через <i> тег

Псевдоклассы (bg-light, text-dark) для стилизации элементов

Система отступов Bootstrap (mb-2, mb-3 и др.)

Сейчас приступим к созданию «Преимущества» (рис. 2.12)

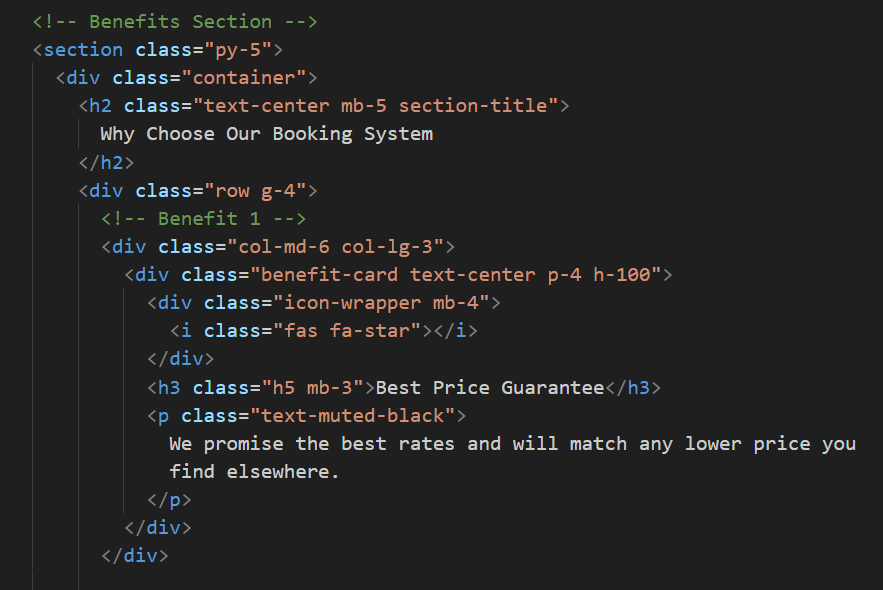


Рис. 2.12

Итог представлен на рис. 2.13

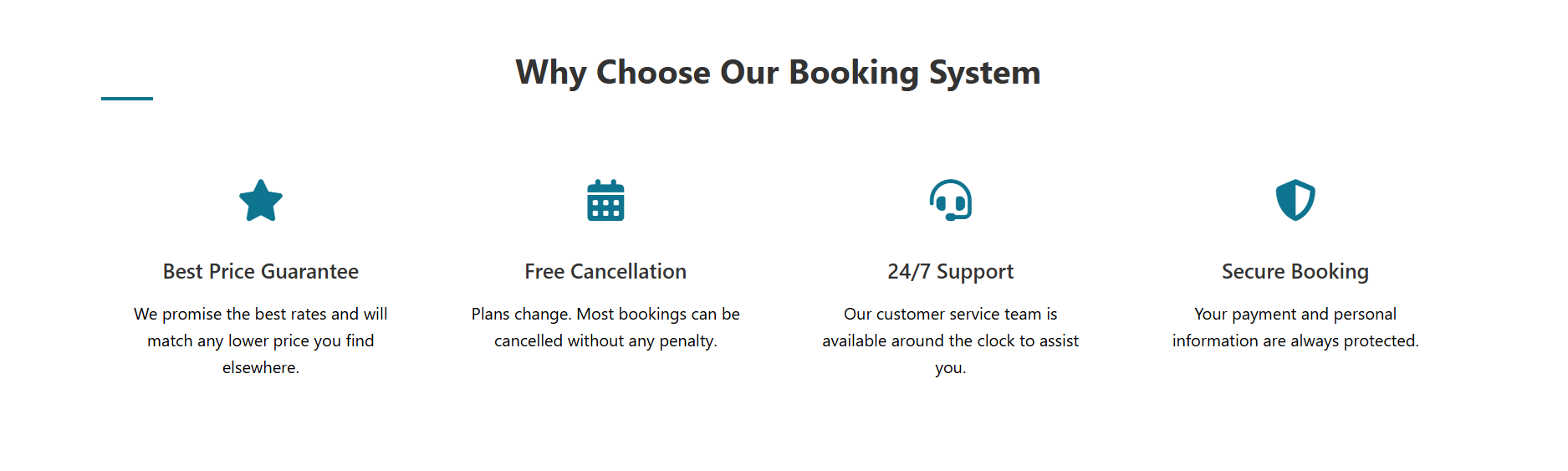


Рис. 2.13

Технический анализ секции "Преимущества"

1. Структура и разметка:

Секция использует семантический HTML5 тег <section>

Контейнер Bootstrap (container) для центрирования контента

Карточки преимуществ используют 3-колоночную сетку на десктопах (col-lg-3)

1. Стилизация:

Вертикальные отступы секции (py-5)

Центрированный заголовок (text-center)

Карточки с:

Фиксированной высотой (h-100)

Внутренними отступами (p-4)

Иконками Font Awesome

Типографикой Bootstrap (h5, text-muted-black)

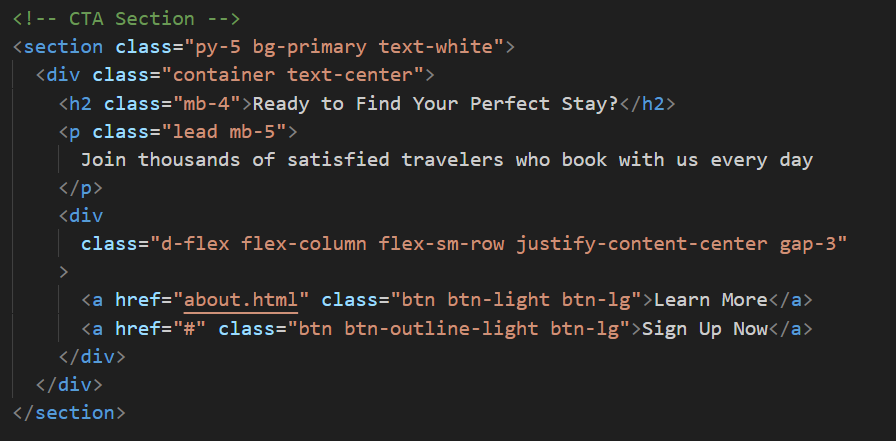


Рис. 2.14. CTA-секция

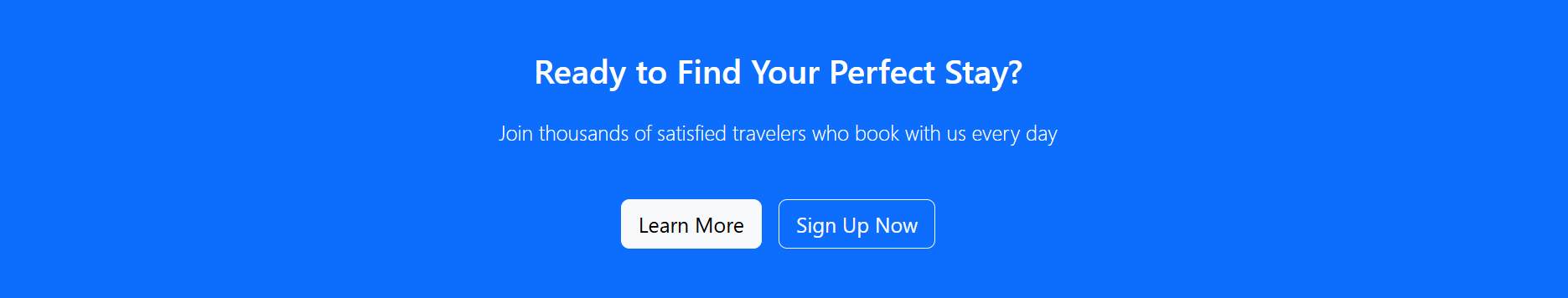


Рис. 2.15

Технический разбор CTA-секции (рис. 2.15)

1. Базовая структура:

Семантическая секция <section> с фоновым цветом (bg-primary)

Центрированный контейнер Bootstrap (container text-center)

Два уровня текста: заголовок (h2) и подзаголовок (lead)

1. Кнопки и их поведение:

Группа кнопок с flex-расположением:

Вертикальное расположение на мобильных (flex-column)

Горизонтальное на планшетах/десктопах (flex-sm-row)

Равные промежутки (gap-3)

Варианты кнопок:

Основная (btn-light btn-lg)

Контурная (btn-outline-light btn-lg)

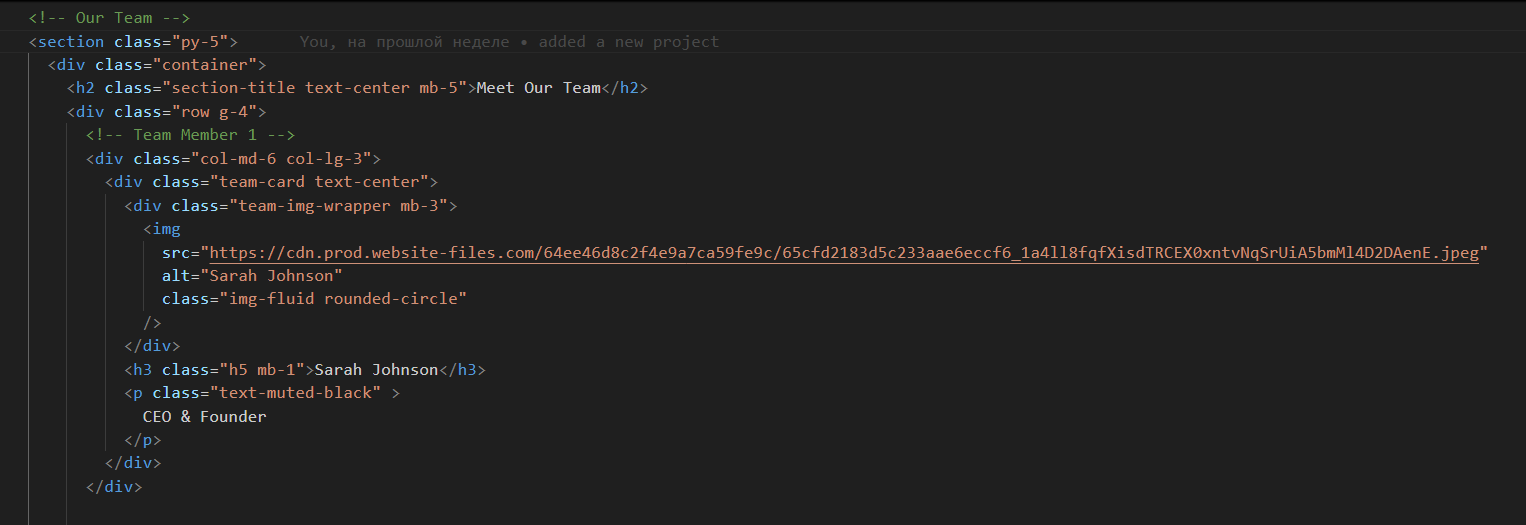


Рис. 2.16

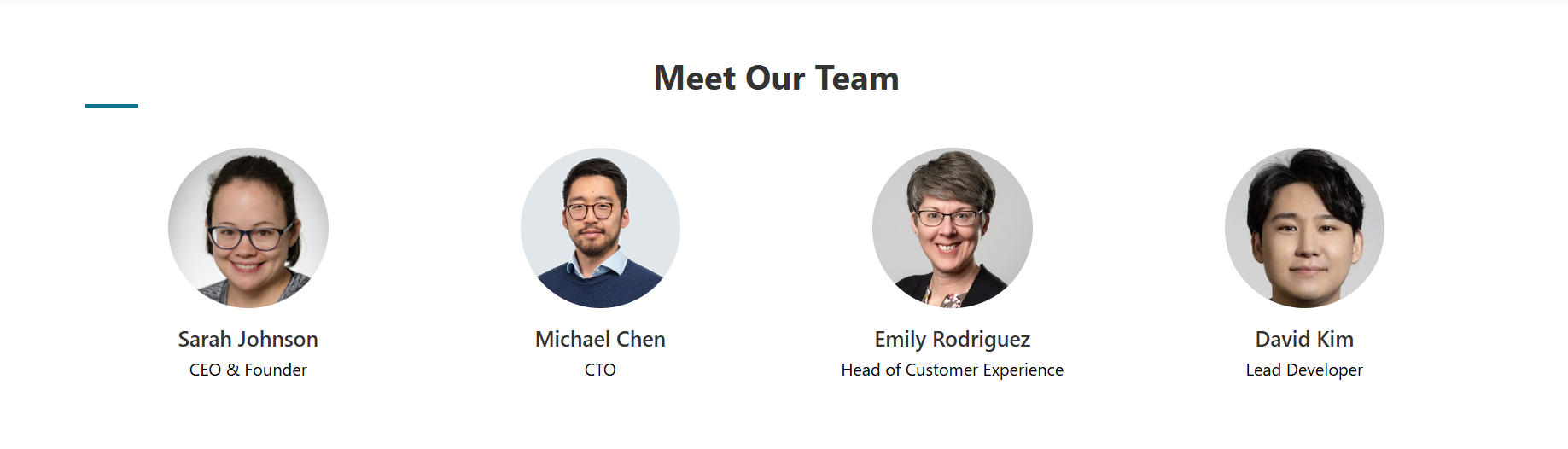


Рис. 2.17

Секция "Наша команда" (рис. 2.17)

Код создает адаптивную секцию для отображения членов команды:

Структура:

Заголовок "Our Team" с центрированием

Сетка карточек сотрудников (Bootstrap row g-4)

Карточка включает:

Круглое фото (rounded-circle)

Имя (заголовок h5)

Должность (text-muted)

Технические особенности:

Адаптивный дизайн:

2 колонки на планшетах (col-md-6)

4 колонки на десктопах (col-lg-3)

Вертикальные отступы (py-5 для секции, mb-3 для элементов)

Центрирование контента (text-center)

Элементы карточки:

1. Контейнер изображения (team-img-wrapper)
2. Оптимизированное фото (img-fluid)
3. Текстовая информация с иерархией

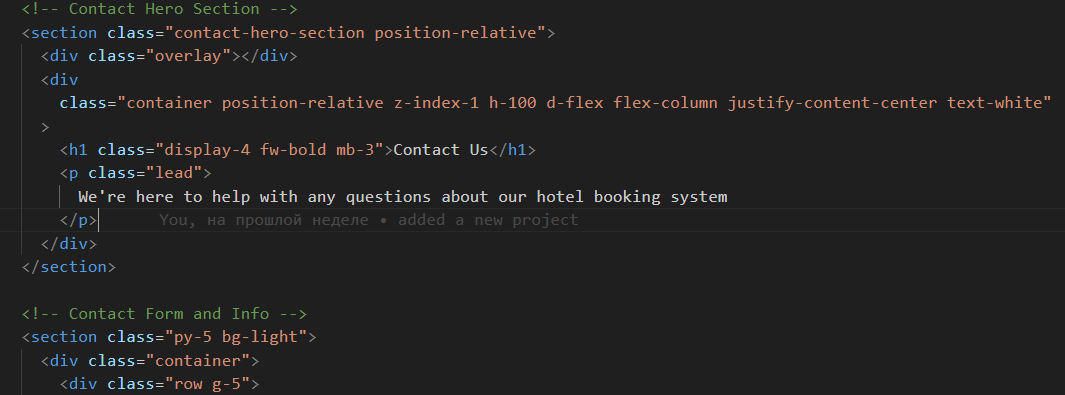


Рис. 2.18.

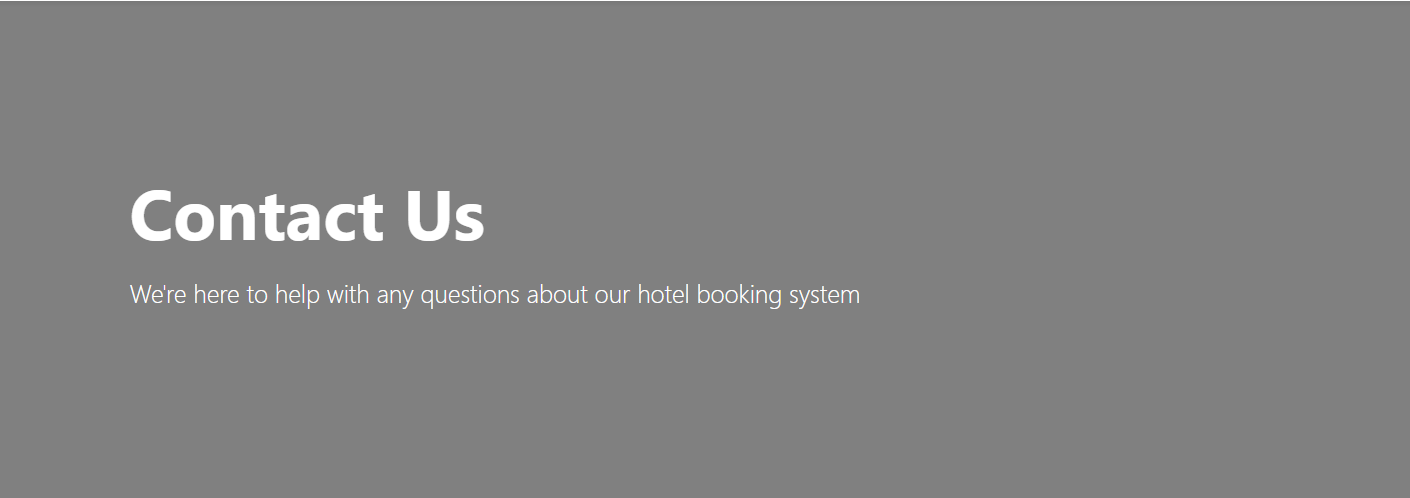


Рис. 2.19

Данный код (рис. 2.18) создает страницу контактов для сайта бронирования отелей, состоящую из двух ключевых секций:

1. Герой-секция (Contact Hero Section):

Визуально привлекательный "шапка" страницы

Содержит основной заголовок "Contact Us"

Включает краткое описание назначения контактов

Использует технику оверлея для улучшения читаемости текста на фоновом изображении

1. Секция с формой и информацией (начало реализации):

Предназначена для размещения:

Формы обратной связи

Контактных данных (телефоны, email, адрес)

Карты местоположения

Имеет светлый фон для визуального отделения от герой-секции



Рис. 2.20

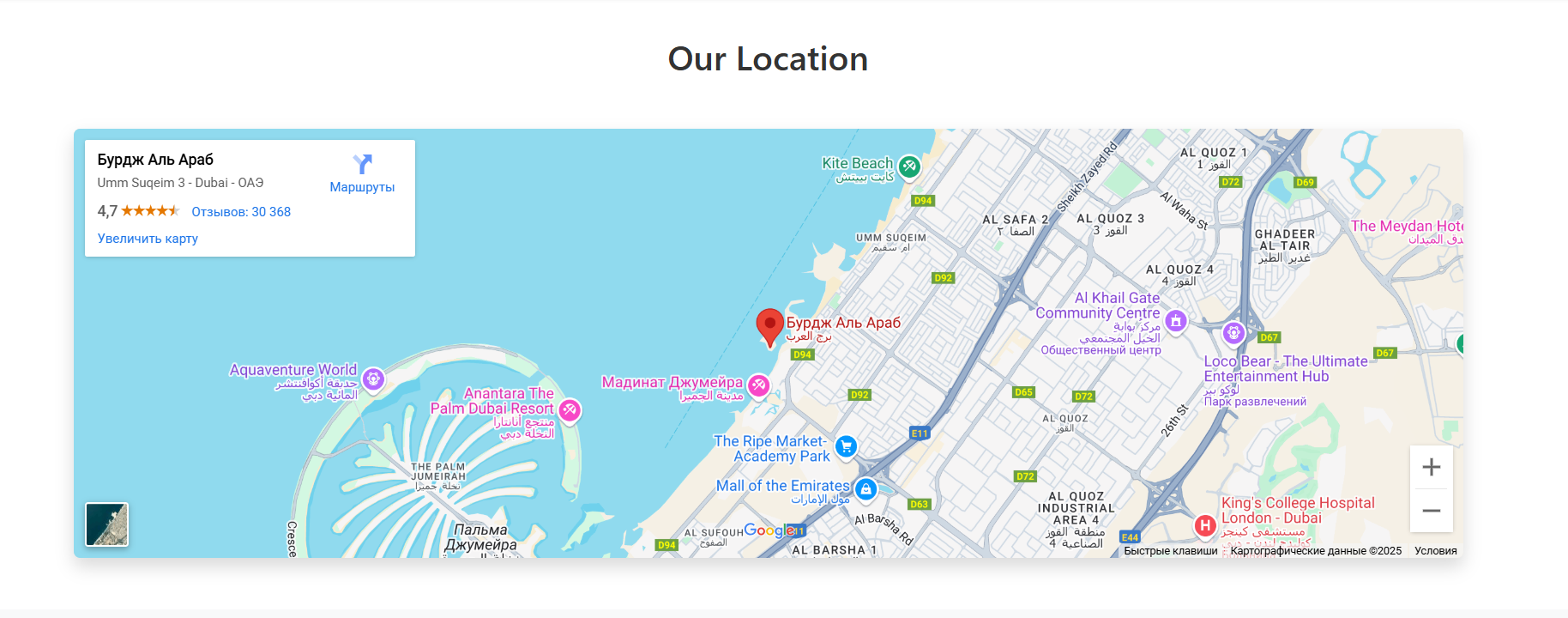


Рис. 2.21. Карта для сайта

Назначение кода (рис. 2.20) "Секция с картой"

Данный код создает раздел с интерактивной картой для сайта (рис. 2.21), который:

1. Отображает местоположение

Встраивает Google Maps через <iframe>

Показывает точный адрес или зону обслуживания

1. Структурные элементы

Заголовок "Our Location" (центрированный)

Контейнер для карты с визуальными эффектами:

Закругленные углы (rounded)

Тень (shadow)

Адаптивное поведение

1. Технические особенности

Отзывчивый дизайн (подстраивается под размер экрана)

Интеграция с API Google Maps

Оптимизированная загрузка (ленивая загрузка при необходимости)



Рис. 2.22

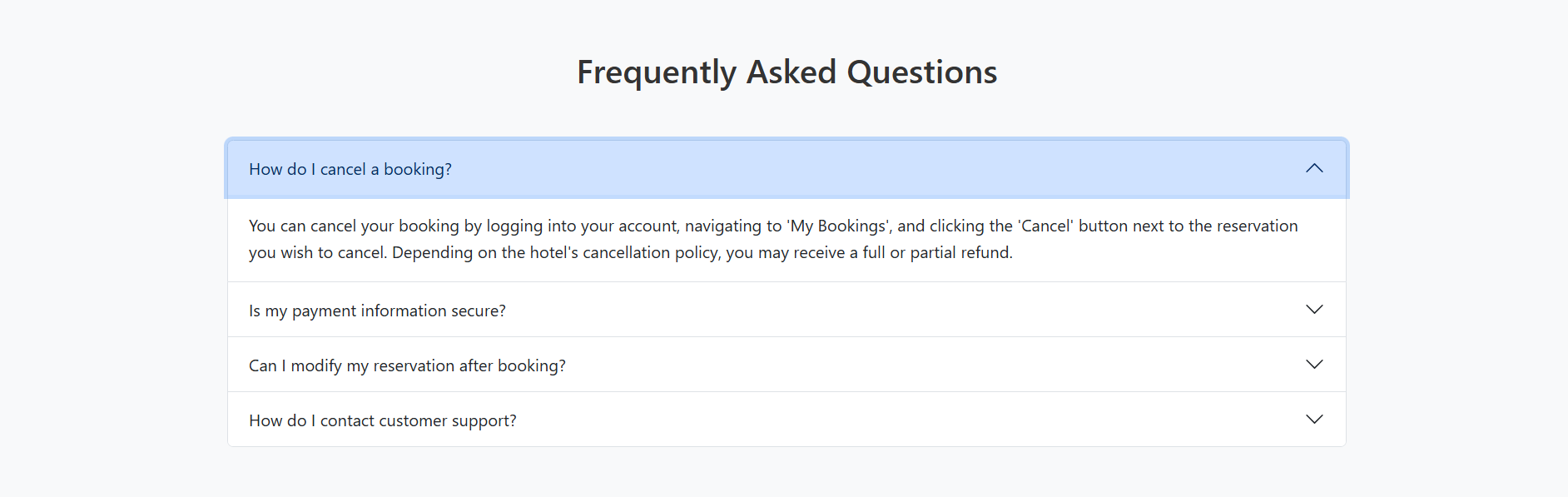


Рис. 2.23

Назначение кода "FAQ-секция"(рис. 2.22)

Данный код реализует раздел с часто задаваемыми вопросами (FAQ) на сайте, используя интерактивные аккордеоны (рис. 2.23). Основные функции:

1. Структура компонента:

Секция со светлым фоном (bg-light)

Центрированный заголовок "Frequently Asked Questions"

Адаптивная сетка Bootstrap (row, col-lg-10)

Аккордеон-система для вопросов/ответов

1. Интерактивные элементы:

Кликабельные вопросы (кнопки accordion-button)

Раскрывающиеся ответы (accordion-collapse)

Анимация открытия/закрытия

Поддержка нескольких вопросов (можно добавить больше items)

**2.4 Использование стилей CSS**

Стили CSS были использованы для оформления элементов на странице, создания визуально привлекательного интерфейса и обеспечения удобства пользователя. В проекте использована стандартная тема Bootstrap для оформления различных блоков: кнопок, карточек, заголовков и форм. Дополнительно был применен кастомный файл стилей (styles.css), который добавляет особенности, такие как тени, отступы, анимации и другие визуальные эффекты, что позволяет сделать страницу более современной и удобной для пользователя. Кастомные стили используются для более гибкой настройки страницы под конкретные требования пользователя.

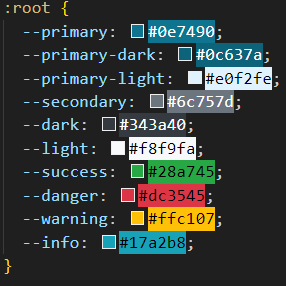


Рис. 2.24

Данный код (рис. 2.24) определяет корневые CSS-переменные для системы цветов проекта.

Основные функции:

1. Цветовая палитра:

-primary: Основной брендовый цвет (#0e7490 - сине-зеленый)

-primary-dark/-light: Темная и светлая вариации

-secondary: Дополнительный нейтральный цвет (#6c757d)

-dark/-light: Контрастные цвета для текста/фона

1. Системные цвета:

-success (зеленый)

-danger (красный)

-warning (желтый)

-info (синий)



Рис. 2.25

Данный код (рис. 2.25) создает настраиваемый оверлей - полупрозрачный слой, который может использоваться для различных целей на веб-странице.

Основные характеристики:

1. Позиционирование и размеры:

Абсолютное позиционирование (position: absolute)

Полное покрытие родительского контейнера (width: 100%, height: 100%)

Фиксация в верхнем левом углу (top: 0, left: 0)

1. Фоновое оформление:

Использует изображение серого цвета в качестве фона

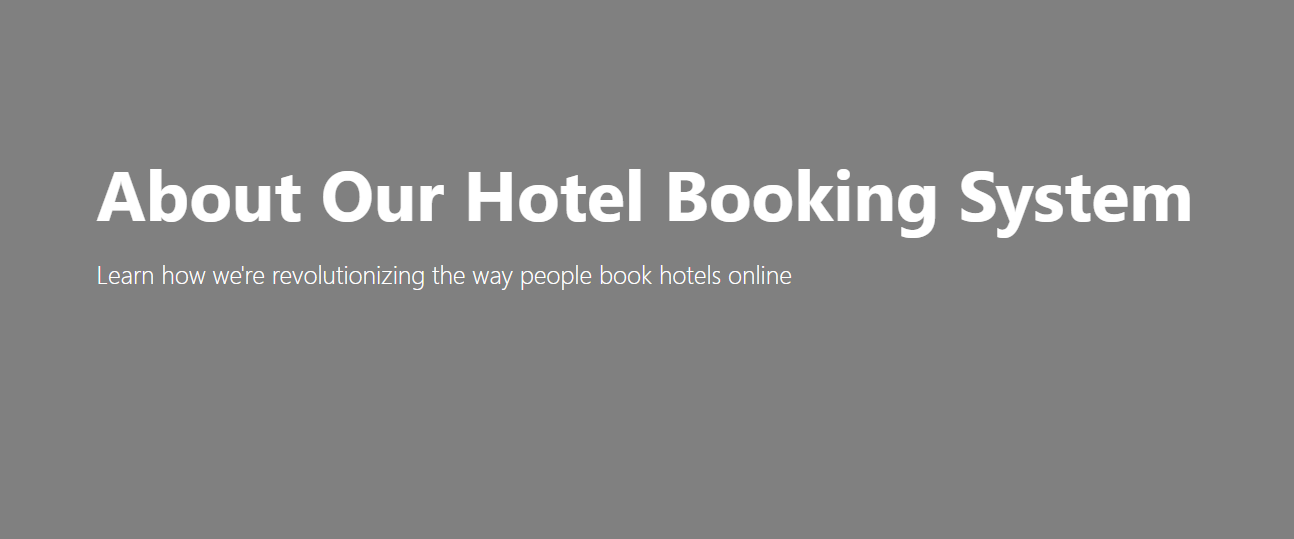
URL фонового изображения указывает на сплошной серый фон (рис. 2.26)

Рис. 2.26

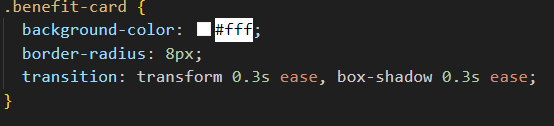


Рис. 2.27

Данный код (рис. 2.27) определяет базовые стили для карточек, отображающих преимущества или особенности сервиса/продукта.

Ключевые стилевые характеристики:

1. Визуальное оформление:

Белый фон (#fff)

Закругленные углы (border-radius: 8px)

Плавные анимации при наведении

1. Интерактивные эффекты:

Плавное изменение трансформации (transform 0.3s ease)

Анимация тени (box-shadow 0.3s ease)

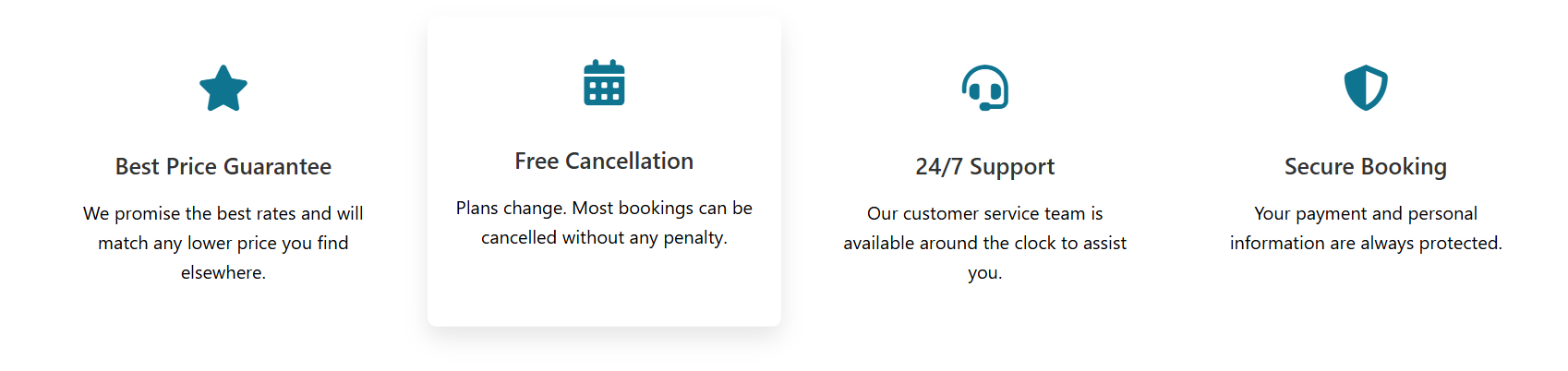


Рис. 2.28

**2.5 Использование JS**

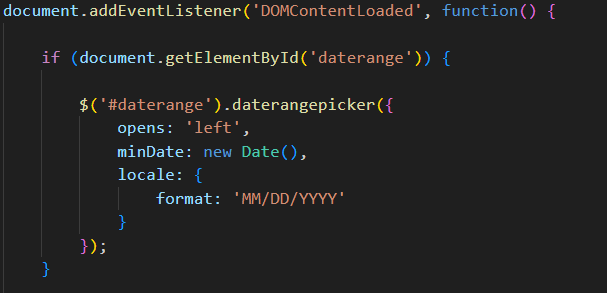
****

Рис. 2.29

Этот код (рис. 2.29) инициализирует плагин для выбора периода дат (daterangepicker) на веб-странице. Основные функции:

1. **Инициализация плагина**:

Активируется после полной загрузки DOM (DOMContentLoaded)

Проверяет наличие элемента с ID daterange

Настраивает Date Range Picker с параметрами:

Открытие календаря слева (opens: 'left')

Минимальная дата - текущий день (minDate: new Date())

Формат даты: ММ/ДД/ГГГГ

1. **Особенности реализации**:

Использует jQuery-селектор *($('#daterange'))*

Локализация формата даты для US-стандарта

Запрещает выбор прошедших дат



Рис. 2.30

Этот код (рис. 2.30) реализует клиентскую логику работы контактной формы с имитацией успешной отправки. Основные функции:

1. Обработка отправки формы:

Отменяет стандартное поведение формы (*preventDefault()*)

Скрывает форму после "отправки" (display: none)

Показывает сообщение об успехе (удаляет класс d-none)

1. Дополнительный функционал:

Кнопка "Отправить еще одно сообщение"

Сброс формы (*reset()*)

Переключение видимости формы и сообщения

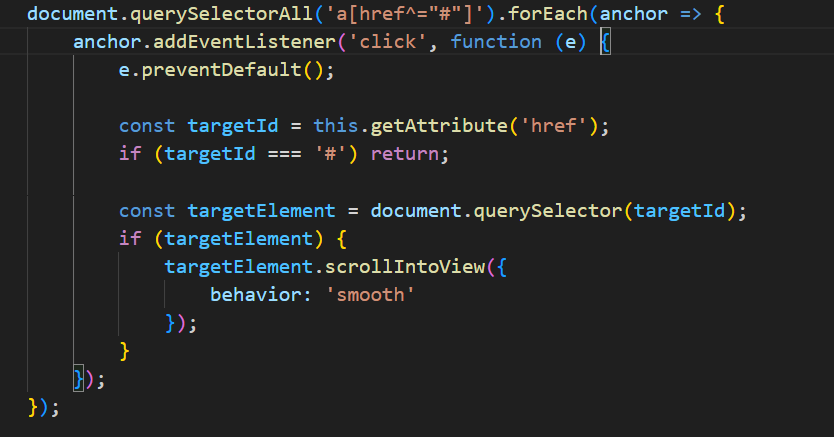


Рис. 2.31

Этот код (рис. 2.31) реализует плавную прокрутку к якорным ссылкам на странице. Основные функции:

1. Механизм работы:

Находит все якорные ссылки (a[href^="#"])

Отменяет стандартное поведение браузера

Плавно прокручивает к целевому элементу

1. Ключевые особенности:

Обработка всех ссылок с href, начинающимся на #

Игнорирование пустых ссылок (href="#")

Использование нативного scrollIntoView с опцией smooth

Современный синтаксис

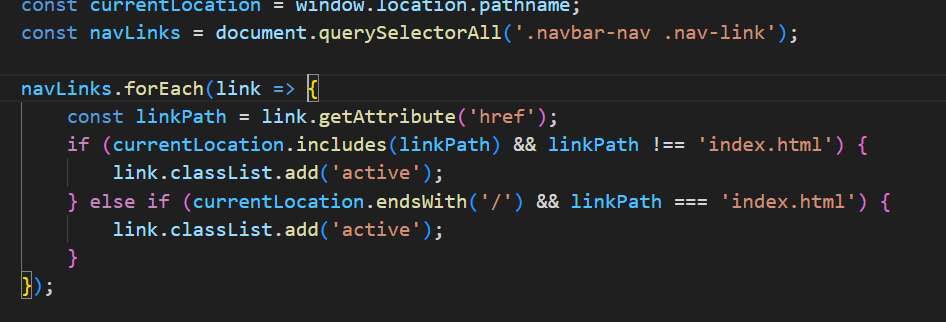


Рис. 2.32

Этот код (рис. 2.32) автоматически отмечает активный пункт меню в навигационной панели, основываясь на текущем URL страницы.

Основные функции:

1. Определяет текущий путь (*window.location.pathname*)
2. Находит все ссылки в навигации (.*navbar-nav .nav-link*)
3. Добавляет класс active для:

Страниц, соответствующих пути ссылки (кроме главной)

Главной страницы при нахождении в корне (*/*)

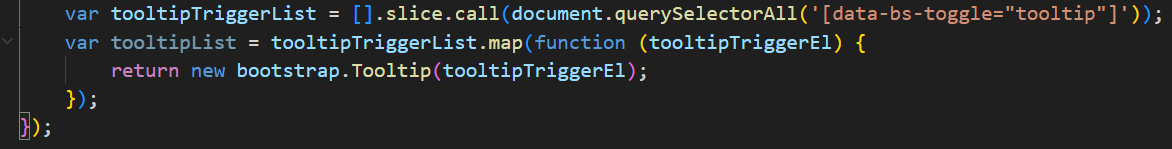


Рис. 2.33

Этот код (рис. 2.33) активирует стандартные всплывающие подсказки Bootstrap 5 для всех элементов, помеченных соответствующим атрибутом.

Основные функции:

1. Находит все элементы с атрибутом *data-bs-toggle="tooltip"*
2. Инициализирует для каждого экземпляр Tooltip из Bootstrap
3. Создает массив экземпляров для возможного последующего управления

**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-murguzov.vercel.app/>

**4****. ЛИТЕРАТУРА**

1. **Даккар, Д.** HTML и CSS. Разработка и дизайн веб-сайтов. — Санкт-Петербург: Питер, 2021. — 512 с
2. **Крамер, Д.** HTML5 и CSS3: современная разработка сайтов. — Москва: Эксмо, 2020. — 384 с.
3. **Фримен Э., Робсон Э.** Изучаем HTML, XHTML и CSS. — СПб: Вильямс, 2018. — 720 с.
4. **Шеферд Б.** CSS. Карманный справочник. — СПб: БХВ-Петербург, 2019. — 256 с.
5. **Mozilla Developer Network (MDN Web Docs).**  
   *HTML:* [*https://developer.mozilla.org/ru/docs/Web/HTML*](https://developer.mozilla.org/ru/docs/Web/HTML)  
   *CSS:* [*https://developer.mozilla.org/ru/docs/Web/CSS*](https://developer.mozilla.org/ru/docs/Web/CSS)

**5. ПРИЛОЖЕНИЯ**

Репозиторий:

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

Страница сайта на хостинге GitHub Pages:

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