Оглавление

[ВВЕДЕНИЕ 3](#_Toc200010571)

[1. ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc200010572)

[1.1. Постановка задачи и цели проекта 5](#_Toc200010573)

[1.2. Анализ предметной области и аналогов 6](#_Toc200010574)

[1.3. Сбор и обработка требований 7](#_Toc200010575)

[1.4. Методы анализа: контент-анализ, моделирование 9](#_Toc200010576)

[2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 11](#_Toc200010577)

[2.1. Разработка технического задания 11](#_Toc200010578)

[2.2. Контекстная диаграмма и диаграмма декомпозиции 13](#_Toc200010579)

[2.3. Диаграмма потоков данных 14](#_Toc200010580)

[2.4. Диаграмма IDF3 14](#_Toc200010581)

[2.5. Проектирование пользовательского интерфейса 14](#_Toc200010582)

[3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ 16](#_Toc200010583)

[3.1. Подготовка среды разработки и выбор технологий 16](#_Toc200010584)

[3.2. Верстка и стилизация интерфейса 18](#_Toc200010585)

[3.3. Реализация логики с использованием JavaScript 21](#_Toc200010586)

[3.4. Работа сайта 25](#_Toc200010587)

[ЗАКЛЮЧЕНИЕ 29](#_Toc200010588)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30](#_Toc200010589)

**ВВЕДЕНИЕ**

В рамках Учебной практики была поставлена задача — создать клиентскую часть веб-сайта сервиса бронирования отелей, ориентированную на различные устройства: от мобильных телефонов до настольных компьютеров.

С развитием цифровых технологий всё большее количество пользователей предпочитает бронировать номера в отелях через интернет, что делает тему сервиса бронирования отелей особенно актуальной. Современный сайт должен быть не только информативным, но и удобным в использовании, адаптированным под экраны различных устройств, а также обеспечивать высокое качество пользовательского опыта. Учитывая эти требования, основной акцент в ходе практики был сделан на верстке и создании интерактивного пользовательского интерфейса.

В ходе проекта были использованы следующие инструменты и технологии:

Figma — для создания прототипа пользовательского интерфейса;

WebStorm — как основная среда разработки;

HTML, CSS и Sass — для разметки и стилизации страниц;

JavaScript — для реализации базовой интерактивности

медиазапросы— для обеспечения адаптивности сайта.

В процессе выполнения была проведена работа по сбору и анализу информации о том, какие функциональные возможности должны быть реализованы в сервисе. Это позволило сформировать общее представление о необходимом наборе функций. На основе этих данных был составлен список требований к сайту, а также определены основные модули и элементы интерфейса.

Для создания дизайна сайта использовался графический редактор Figma, где был разработан прототип пользовательского интерфейса. Макет включал в себя главную страницу, страницу отеля, форму бронирования и другие элементы, обеспечивающие логическую целостность и удобство использования. Дизайн строился с учётом современных трендов веб-дизайна и принципов юзабилити, что позволило создать приятный и понятный пользовательский опыт.

Особое внимание было уделено созданию интерактивного меню, которое должно корректно отображаться и работать на маленьких экранах. Для мобильных устройств было реализовано скрытое бургер-меню, которое открывается по клику, что улучшает юзабилити и делает навигацию более удобной. На десктопных устройствах меню отображается в полной форме, обеспечивая быстрый доступ ко всем разделам сайта.

Также была реализована базовая структура страницы сервиса, включающая такие элементы, как шапка сайта, секция с предложением услуг, примеры отелей и другие. Все блоки были сверстаны с учётом принципов модульности, чистоты кода и кроссбраузерности.

Сама верстка выполнялась в среде разработки WebStorm — мощном редакторе кода, поддерживающем широкий спектр современных веб-технологий. Для написания стилей использовалась библиотека Sass, которая предоставляет дополнительные возможности по сравнению с обычным CSS, такие как переменные, миксины, вложенные правила и другие полезные функции. Применение Sass значительно повысило читаемость и поддерживаемость кода, а также ускорило процесс разработки.

1. **ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**
   1. **Постановка задачи и цели проекта**

Целью являлась реализация клиентской части веб-сайта сервиса бронирования отелей на основе уже готового макета в Figma. Основной задачей стало создание современного, адаптивного под различные устройства пользовательского интерфейса с элементами интерактивности, с использованием современных веб-технологий и инструментов разработки.

Цели проекта:

- Реализовать фронтенд сайта сервиса бронирования отелей по предоставленному макету.

- Обеспечить корректное отображение сайта на различных устройствах: мобильных телефонах, планшетах и десктопах.

- Создать интерактивные элементы, включая меню навигации для мобильных устройств.

- Научиться применять на практике такие технологии, как HTML, CSS, препроцессор Sass и JavaScript.

- Получить опыт работы с профессиональными инструментами разработки: WebStorm, Figma.

Задачи проекта:

1. Анализ готового макета в Figma. Перед началом верстки была проведена работа по изучению структуры и компонентов макета. Это позволило понять, какие блоки необходимо реализовать, какова логика расположения контента и какие цветовые и типографические решения используются.

2. Верстка сайта на основе Figma-макета. С использованием HTML и CSS была выполнена точная реализация главной страницы сервиса бронирования отелей. Были воспроизведены все основные элементы: шапка сайта, секция поиска, карточки отелей, информационные блоки и футер. При этом применялись семантические теги для повышения читаемости кода и соответствия стандартам веб-разработки.

3. Использование препроцессора Sass. Для написания стилей был выбран препроцессор Sass, который позволил организовать CSS-код в модульном виде, использовать переменные, миксины и вложенные правила. Это сделало стили более структурированными, удобными для чтения и дальнейшего редактирования.

4. Реализация адаптивного дизайна. Особое внимание было уделено созданию адаптивной верстки, позволяющей сайту корректно отображаться на экранах различного размера. Использовались медиазапросы, относительные единицы измерения (rem, %).

5. Создание интерактивного меню. Было реализовано интерактивное меню навигации, которое изменяет своё отображение в зависимости от типа устройства. На мобильных устройствах меню скрывается под кнопку "бургер", которая раскрывается при клике. Простая логика открытия/закрытия меню была реализована с помощью JavaScript, что добавило динамики и улучшило пользовательский опыт.

6. Тестирование и проверка совместимости. После завершения верстки были проведены тесты отображения сайта в различных браузерах (Chrome, Firefox) и на разных устройствах. Проверялось корректное отображение элементов, работа адаптивности и взаимодействие с пользователем.

7. Подготовка результатов и отчета. По окончании были собраны итоговые файлы проекта, подготовлены скриншоты и документация, а также оформлен отчёт о проделанной работе.

* 1. **Анализ предметной области и аналогов**

Для реализации фронтенда сервиса бронирования отелей был проведён сравнительный анализ трёх готовых макетов, разработанных в графическом редакторе Figma. Основная цель анализа — выбрать наиболее подходящий вариант для верстки с точки зрения удобства, простоты реализации и пользовательского опыта.

На основании проведённого анализа и сравнения макетов, для дальнейшей реализации был выбран макет. Этот выбор обусловлен следующими факторами:

- Простота и минимализм дизайна способствуют высокому уровню юзабилити.

Таблица 1 - Сравнительная таблица аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| критерии | TEZPlus | EasySet24 | Hotelo |
| Простой и чистый интерфейс | + | - | + |
| Ясная структура | + | + | + |
| Адаптивный дизайн | + | - | + |
| Ui- компоненты | + | - | - |

Таким образом, данный макет обеспечивает хороший баланс между эстетикой, функциональностью и возможностями технической реализации, что делает его оптимальным вариантом для сайта.

* 1. **Сбор и обработка требований**

Была поставлена задача реализовать клиентскую часть сайта сервиса бронирования отелей на основе готового макета из Figma. Для выполнения этой задачи были собраны и проанализированы функциональные и нефункциональные требования, а также определены технологии и подходы к верстке, включая использование методологии БЭМ.   
Источники сбора требований:

* Готовый макет интерфейса в Figma;
* Анализ аналогов;
* Личный опыт работы с веб-интерфейсами и современными стандартами разработки.

Функциональные требования:

1. Реализация главной страницы сервиса бронирования отелей.
2. Верстка формы поиска отелей с полями: город, даты заезда/выезда, количество гостей.
3. Отображение карточек отелей с изображением, названием, ценой и рейтингом.
4. Наличие шапки сайта с логотипом, меню навигации и кнопкой мобильного меню.

Нефункциональные требования:

1. Адаптивность. Сайт должен корректно отображаться на устройствах с различными размерами экранов: мобильные телефоны, планшеты, десктопы
2. Кроссбраузерность. Работоспособность должна быть обеспечена в современных браузерах: Chrome, Firefox, Safari, Edge.
3. Удобство использования. Интерфейс должен быть интуитивно понятным и соответствовать принципам юзабилити.
4. Производительность. Минимальная загрузка ресурсов, быстрая отрисовка страницы.
5. Поддерживаемость кода. Код должен быть чистым, структурированным, легко модифицируемым и масштабируемым

Обработка требований:   
Для удовлетворения требований к структуре и читаемости кода была применена методология БЭМ. Это позволило:

1. Разделить интерфейс на независимые блоки, которые можно легко пере использовать и изменять.
2. Избежать конфликтов в классах и сделать стили более предсказуемыми.
3. Упростить командную разработку благодаря стандартизированному подходу к именованию элементов.

Преимущества применения БЭМ:

1. Чёткая иерархия и структура HTML и CSS.
2. Возможность повторного использования блоков.
3. Упрощённая отладка и поддержка стилей.
4. Удобство масштабирования проекта при добавлении новых компонентов.

На данном этапе были определены ключевые характеристики будущего сайта, выбраны технологии и подходы к верстке, что позволило создать структурированный, понятный и легко поддерживаемый код. Все собранные требования были реализованы в ходе практики, и результатом стало создание рабочего прототипа фронтенда сервиса бронирования отелей.

* 1. **Методы анализа: контент-анализ, моделирование**

Контент-анализ проводился с целью выявления ключевых элементов и структуры сайта, необходимых для реализации функционала сервиса бронирования отелей. В рамках данного этапа были проанализированы:

1. Информационная архитектура — на основе Figma-макета была изучена структура веб-страниц, определены основные разделы и подразделы сайта. Это позволило понять логику навигации и взаимодействия пользователя с интерфейсом.
2. Элементы пользовательского интерфейса — проведён детальный разбор всех визуальных компонентов: кнопки, карточки отелей, меню и другие элементы. Анализ показал, что сайт ориентирован на удобство использования и минимизацию времени на поиск информации.
3. Типы контента — определены типы данных, которые будут отображаться на сайте: информация об отелях, фотографии, отзывы пользователей. На основе этого были подготовлены шаблоны отображения данных и продуманы пути их интеграции в будущее приложение.
4. Целевая аудитория — анализ макета позволил сделать вывод о том, что сайт рассчитан как на частных туристов, так и на корпоративных клиентов.

Были учтены потребности разных категорий пользователей при построении интерфейса и функциональных возможностях.

Моделирование

На этапе моделирования были использованы следующие подходы:

1. Моделирование по методологии БЭМ— весь интерфейс был структурирован согласно принципам БЭМ. Это позволило унифицировать разметку и стили, упростить повторное использование компонентов, повысить читаемость кода и облегчить дальнейшее сопровождение проекта.
2. Адаптивное моделирование — на основе анализа макета были определены точки адаптации под различные устройства (мобильные телефоны, планшеты, десктоп). Для каждого типа устройств были спроектированы собственные стили и поведение интерфейса.
3. Моделирование работы интерактивного меню — было спроектировано поведение выпадающего меню, включая логику открытия/закрытия, анимации, работу с клавиатурой и сенсорными экранами. Это позволило создать более доступный и удобный интерфейс для всех пользователей.
4. Использование препроцессора SASS — на основе проведённого анализа и моделирования была организована структура файлов стилей с использованием SASS: отдельные файлы под переменные, миксины, базовые стили, блоки и т.д. Это обеспечило гибкость и простоту управления стилями.

Вывод:

Проведённый анализ и моделирование позволили четко определить структуру и составляющие элементы сайта, а также выбрать наиболее эффективные подходы к его верстке и дальнейшей разработке. Использование методологии БЭМ и препроцессора SASS способствовало организации модульной и масштабируемой архитектуры проекта. Адаптивность и интерактивность интерфейса были заложены уже на этапе проектирования, что положительно сказалось на качестве конечного продукта.

1. **ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**
   1. **Разработка технического задания**

На основе проведенного анализа, сбора сведений была разработано техническое задание на верстку сервиса бронирования отелей.

Разработка и реализация веб-сайта, предоставляющего пользователям возможность поиска, просмотра информации об отелях и бронирования номеров.

Страница сайта:

1. Header
2. Intro
3. Секция преимуществ
4. Лидеры Продаж
5. Секция Выгоды
6. Секция наград
7. Секция офисов
8. Отзывы
9. Footer

Инструменты разработки:

Редактор кода – WebStrom.

Препроцессор: SCSS.

Технические требования:

* Интерактивное меню для телефонов
* Адаптивный дизайн
* Все стили должны вынесены в отдельные компоненты согласно БЭМ
* Названия классов должны соответствовать правилам БЭМ

В результате был сверстан веб-сайт сервиса бронирования отелей с полностью адаптивным дизайном и интерактивным меню на мобильных устройствах

Техническое задание определило требования к сайту сервиса бронирования отелей. Реализация проекта ведётся с применением современных веб-технологий и методологий, обеспечивающих высокую читаемость кода, простоту его сопровождения и масштабируемости. Использование Figma-макета позволило точно воспроизвести задуманный дизайн, а применение методологии БЭМ и препроцессора SASS способствовало созданию модульной и гибкой структуры проекта.

* 1. **Контекстная диаграмма и диаграмма декомпозиции**

Контекстная диаграмма - это высокоуровневое визуальное представление, которое показывает взаимодействие показывает взаимодействие между разрабатываемой системы и ее внешними объектами(сущностями).



Рисунок 1 - Контекстная диаграмма

Элементы контекстной диаграммы:

Сервис бронирования отелей – центральный элемент диаграммы представляющий веб-приложение, управляющее бронированием отелей, номеров; информацией об отелях и платежами.

Внешние данные:

1. Информация о клиентах
2. Информация о отелях

Внешние сущности:

1. Администратор
2. Персонал

Диаграмма декомпозиций – это структурированное визуальное представление, разбивающее сложную систему на более простые компоненты с выявлением взаимосвязей.

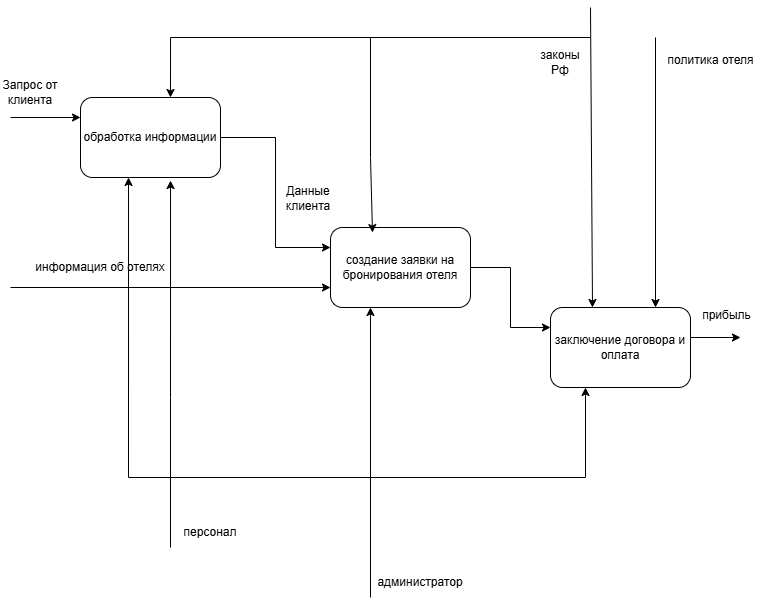


Рисунок 2 - Диаграмма декомпозиций

Подсистемы сервиса бронирования отелей:

1. Подсистема “ обработка информации” – отвечает за обработку данных клиента
2. Подсистема “создание заявки на бронирование” – отвечает за создание заявку клиента на бронирование номера в отеле.
3. Подсистема “Оплата” – обеспечивает безопасную оплату услуг.
   1. **Диаграмма потоков данных**

Диаграмма потоков данных - это визуальное представление перемещения данных в пределах процесса или системы. Она позволяет понять, откуда поступают данные и как они обрабатываются в системе.

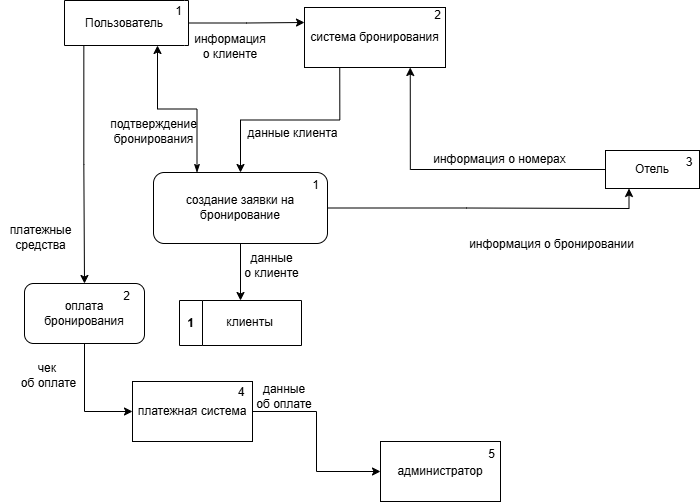


Рисунок 3 - Диаграмма потоков данных

Участники системы:

1. Пользователь
2. Система бронирования
3. Отель
4. Внешние сервисы (платежная система)

Основные процессы:

1. Пользователь отправляет заявку на бронирование
2. Система бронирования подтверждает заявку
3. Пользователь производить оплату
4. Платежная система отправляет данные об оплате администратору
5. Отель получает информацию о бронировании.
   1. **Диаграмма IDF3**

IDF3 Диаграмма – это графическое представление последовательности действий и связей между ними.

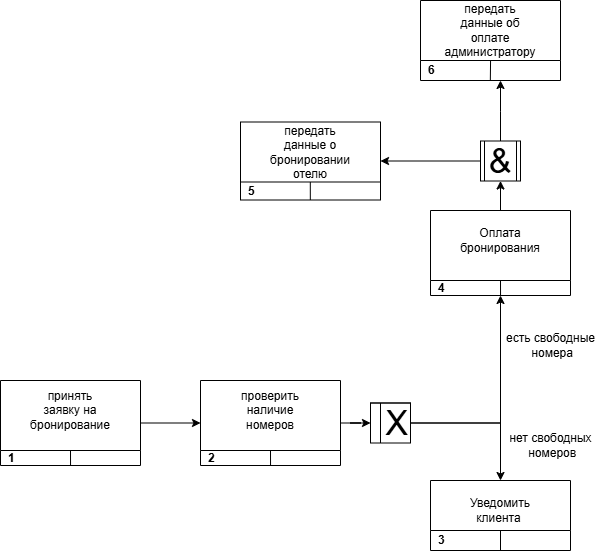


Рисунок 4 - IDF3 диаграмма

Описание процессов:

1. Принять заявку на бронирование - система получает запрос на бронирование от клиента и данные клиента.
2. Проверить наличие номеров – система проверяет наличие номеров, если номеров нет процесс переходит “уведомить клиента”.
3. Уведомить клиента – система отправляет клиенту сообщение о том, что бронирование невозможно.
4. Передать данные о бронировании отелю – система передает данные о бронировании отелю.
5. Передать данные об оплате – Администратор получает информацию об успешной оплате клиента.
   1. **Проектирование пользовательского интерфейса**

При проектировании пользовательского интерфейса сайта «Сервис бронирования отелей» были применены принципы юзабилити, направленные на создание интуитивно понятного и удобного для пользователя дизайна. Основной задачей стало точное воспроизведение макета Figma, при этом сохранить логическую структуру и последовательность расположения элементов.

В рамках проектирования был создан набор базовых компонентов, из которых строится весь интерфейс:

Кнопки — разработаны различные стили кнопок (основная, вторичная, компактная), учитывающие состояние наведения и клика.

Карточки направлений — унифицированные блоки с изображением, заголовком и ценой, используемые в секции лидеров продаж.

Иконки — плоский стиль, единые размеры и цветовое оформление для целостности дизайна.

Текстовые блоки — оформлены с учётом иерархии: заголовки, подзаголовки, основной текст.

Каждый компонент был протестирован на соответствие общей стилистике сайта и простоте использования.

Цветовая палитра и визуальная идентификация

Цветовая схема была взята напрямую из Figma-макета, чтобы обеспечить полное соответствие дизайну:

Основной цвет: #015BC1.

Акцентный цвет: # FFDB5D.

Фоновые цвета: #F9F9FA и #FFF.

Цвет текста: #000 и #FFF.

В результате выполнения этапа проектирования пользовательского интерфейса был создан набор унифицированных компонентов, обеспечивающих целостность и читаемость дизайна.

1. **РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ**
   1. **Подготовка среды разработки и выбор технологий**

Для реализации проекта «сервис бронирования отелей» была настроена локальная среда разработки, включающая современные инструменты и технологии:

Среда разработки: WebStorm — выбрана как основной редактор кода благодаря своей мощной функциональности, встроенной поддержке современных веб-технологий и глубокой интеграции с фреймворками и инструментами.

Система контроля версий: Git — используется для управления изменениями в коде.

Репозиторий: удалённый через GitHub/GitLab.

Менеджер пакетов: npm — применялся для установки и управления зависимостями.

Браузеры для тестирования: Google Chrome, Mozilla Firefox.

6.2 Выбор среды разработки (WebStorm)

Для разработки сайта был выбран WebStorm — мощная IDE от JetBrains, специально разработанная для веб-разработки.

Таблица 2 - сравнительная таблица сред разработки

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **WebStorm** | **Visual Studio Code** | **Sublime Text** |
| Тип | Профессиональная IDE | Открытый редактор с плагинами | Лёгкий текстовый редактор |
| Встроенная поддержка технологий | HTML, CSS, JavaScript, TypeScript, SASS, React и др. | Требует расширений | Ограниченная поддержка |
| Интеграция с Git | Полноценная поддержка через интерфейс | Через расширения или терминал | Необходимы плагины |
| Автодополнение и навигация | Умное автодополнение, поиск по проекту | Необходимы плагины | Базовая поддержка |
| Производительность | Средняя | Высокая | Очень высокая |
| Стоимость | Платная (есть бесплатная пробная версия) | Бесплатная | Условно-бесплатная |
| Гибкость настройки | Высокая | Очень высокая | Средняя |

Выбор WebStorm обусловлен следующими преимуществами:

Встроенная поддержка всех ключевых веб-технологий без необходимости установки дополнительных плагинов.

Интеллектуальные подсказки, проверка ошибок и рефакторинг кода.

Удобство работы с проектами большой сложности.

Глубокая интеграция с системами контроля версий, сборщиками и фреймворками.

Поддержка методологии БЭМ и препроцессоров, таких как Sass.

При разработке сайта были выбраны следующие технологии:

HTML5 / CSS3 — для создания структуры и оформления страниц.

Sass (SCSS) — препроцессор CSS, позволяющий организовать стили модульно и использовать переменные, миксины и другие удобства.

JavaScript — язык программирования для реализации интерактивных элементов: меню, модальных окон, форм.

Проект был организован по принципам БЭМ

В ходе подготовки среды разработки была выбрана профессиональная IDE — WebStorm, которая обеспечила максимальную производительность и комфорт при работе с современными веб-технологиями. Её преимущество перед такими редакторами, как Visual Studio Code и Sublime Text, заключается в глубокой интеграции с инструментами веб-разработки и минимальной зависимости от сторонних плагинов. Использование Sass, JavaScript и методологии БЭМ позволило создать модульный, читаемый и легко поддерживаемый код.

* 1. **Верстка и стилизация интерфейса**

В рамках учебной практики была выполнена вёрстка сайта «Сервис бронирования отелей» на основе готового макета, разработанного в Figma. Основной задачей являлось точное воспроизведение дизайна с учётом адаптивности и семантической корректности HTML-разметки.

Верстка выполнялась с использованием следующих инструментов:

HTML — для структурирования содержимого страниц.

CSS3/ SCSS — для описания внешнего вида элементов.

Методология БЭМ — для организации классов и файловой структуры проекта.

Для обеспечения модульности, читаемости и масштабируемости кода была применена методология Блок — Элемент — Модификатор (БЭМ).

Примеры использования:

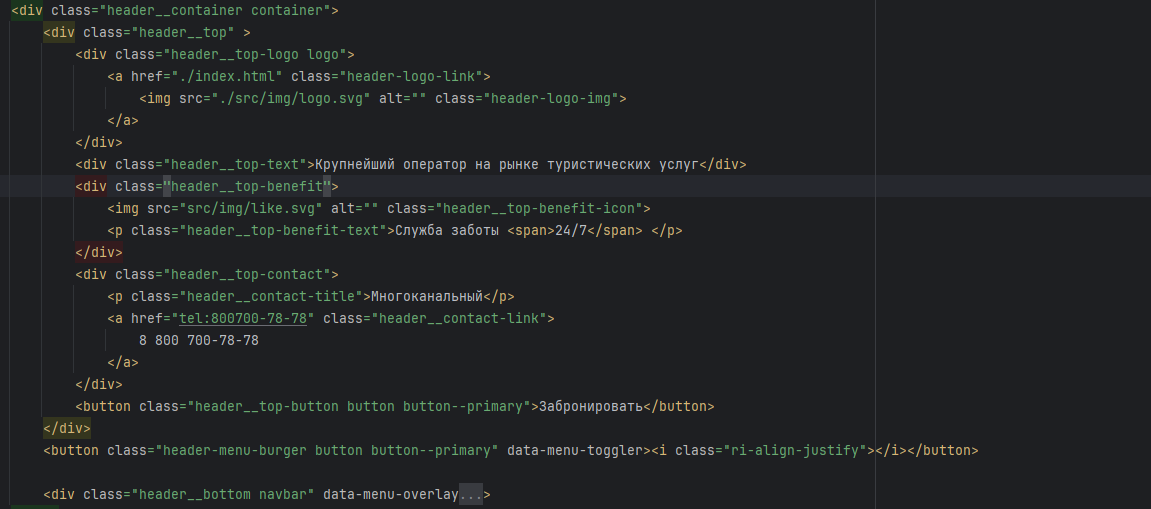


Рисунок 5 - пример названий классов по БЭМ

Структура файлов в папке scss:



Рисунок 6 - структура файлов по БЭМ

Использование БЭМ позволило:

Упростить понимание взаимосвязей между элементами.

Избежать конфликтов имен классов.

Повторно использовать компоненты в разных частях сайта.

Адаптивность сайта была реализована с помощью медиазапросов. Это позволило создать удобный интерфейс как для десктопных, так и для мобильных устройств.

Для управления адаптивностью применялись:

CSS-медиазапросы.

Flexbox и Grid Layout.

Относительные единицы измерения (rem, %).

Для упрощения работы со стилями был выбран препроцессор Sass (SCSS). Он предоставил следующие возможности:

Вложенные селекторы.

Переменные ($color-primary, $font-size-base).

Миксины (@mixin media(), @mixin btn-style()).

Импорт файлов по структуре БЭМ.

Описание структуры:

Описание структуры:

\_fonts.scss: Настройки шрифтов и их импорта.

\_functions.scss: Пользовательские функции для SCSS (например, вычисление размеров).

\_globals.scss: Глобальные стили, такие как общие настройки body.

\_utils.scss: Утилитарные классы (например, .container.)

\_vars.scss: Хранение переменных (цвета, шрифты, отступы).

components/

Разбитие на отдельные файлы для каждого компонента и секции (например, \_buttons.scss для кнопок, \_header.scss для заголовка).

Каждый компонент содержит только его собственные стили, что упрощает поддержку и повторное использование.

mixins/

\_breakpoints.scss: Миксин для медиазапросов

\_fonts-face.scss: Настройки шрифтов через @font-face.

\_layouts.scss: Миксины для различных лэйаутов (например, гриды или flexbox).

Main.scss - Главный файл, который объединяет все остальные части проекта.

Препроцессор позволил значительно повысить читаемость и поддерживаемость кода, а также ускорить процесс разработки.

* 1. **Реализация логики с использованием JavaScript**

Для обеспечения интерактивности элементов интерфейса в проекте был применён язык программирования JavaScript.

Структура JavaScript-файлов

Проект включает минимальную JS-логику, организованную по модульному принципу:

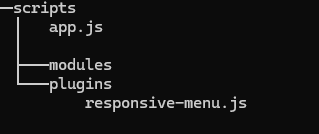


Рисунок 7 - структура файлов скриптов

Описание файлов:

plugins/responsive-menu.js — содержит логику открытия/закрытия мобильного меню.

app.js — главный файл, в котором подключаются и инициализируются компоненты.

Основные функции для интерактивного меню:

Открытие меню (openMenu) : Показывает меню.



Рисунок 8 - код на открытия меню

Закрытие меню (closeMenu) : Скрывает меню.

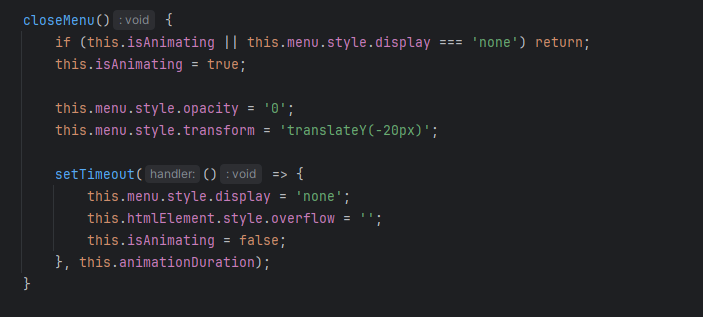


Рисунок 9 - код на закрытие меню

Обработка событий клика: Открывает/закрывает меню при нажатии на кнопку или клике вне области меню.

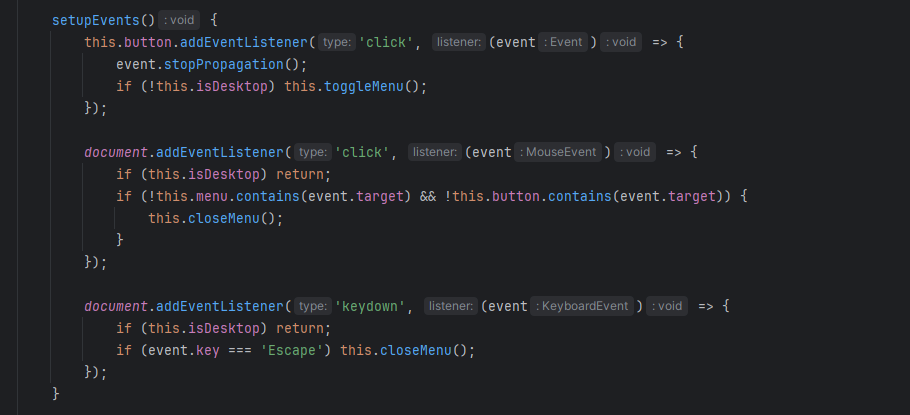


Рисунок 10 - код на обработку клика

Адаптивность: Автоматически переключается между версиями меню для десктопа и мобильных устройств.



Рисунок 11 - код для стилей на мобильных устройствах

Описание функционала

1. Конструктор (constructor)

Инициализирует объект класса Menu с необходимыми элементами и настройками:

Параметры конструктора:

buttonSelector: CSS-селектор кнопки бургера.

menuSelector: CSS-селектор меню.

animationDuration: Длительность анимации (по умолчанию 300 мс).

breakpoint: Разрешение экрана, выше которого меню становится десктопным (по умолчанию 992 пикселя).

htmlElementSelector: Элемент, который будет блокировать скроллинг при открытом меню.

Методы управления состоянием меню

1. Открытие меню (openMenu):

Показывает меню с анимацией.

Блокирует скроллинг страницы, чтобы пользователь не мог прокручивать контент во время открытия меню.

Использует requestAnimationFrame для плавной анимации.

1. Закрытие меню (closeMenu):

Скрывает меню с анимацией.

Разблокирует скроллинг страницы после закрытия меню.

1. Обработка событий (setupEvents)

Клик на кнопку бургера: Открывает/закрывает меню.

Клик вне меню: Закрывает меню, если оно открыто.

Нажатие клавиши Esc: Закрывает меню, если оно открыто.

1. Обработка изменения размера экрана (addResizeListener)

При изменении размера экран автоматически переключает стиль меню между мобильной и десктопной версиями.

* 1. **Работа сайта**

Сайт представляет собой одностраничный интерфейс, разбитый на несколько основных секций, которые адаптивно перестраиваются под размер экрана. Каждая секция предназначена для выполнения определённой задачи: демонстрации преимуществ сервиса, отображения популярных направлений, показа отзывов и наград.

Основные секции:

Header — содержит логотип, кнопку бургера (для мобильных устройств) и главное меню (для десктопных).

Intro — приветственное сообщение.

Секция преимуществ — демонстрирует ключевые особенности сервиса.

Лидеры продаж — отображает популярные направления с возможностью быстрой брони.

Секция "Почему TEZPlus" — объясняет выгоды использования платформы.

Наши награды — демонстрирует достижения компании.

Отзывы - показывает отзывы пользователей.

Функциональность

1. Intro

На главной странице расположено приветственное сообщение о том, что сервис позволяет легко и выгодно забронировать отель, апартаменты или виллу в любой точке планеты. Здесь же находится кнопка "Забронировать номер", которая направляет пользователя к форме бронирования.

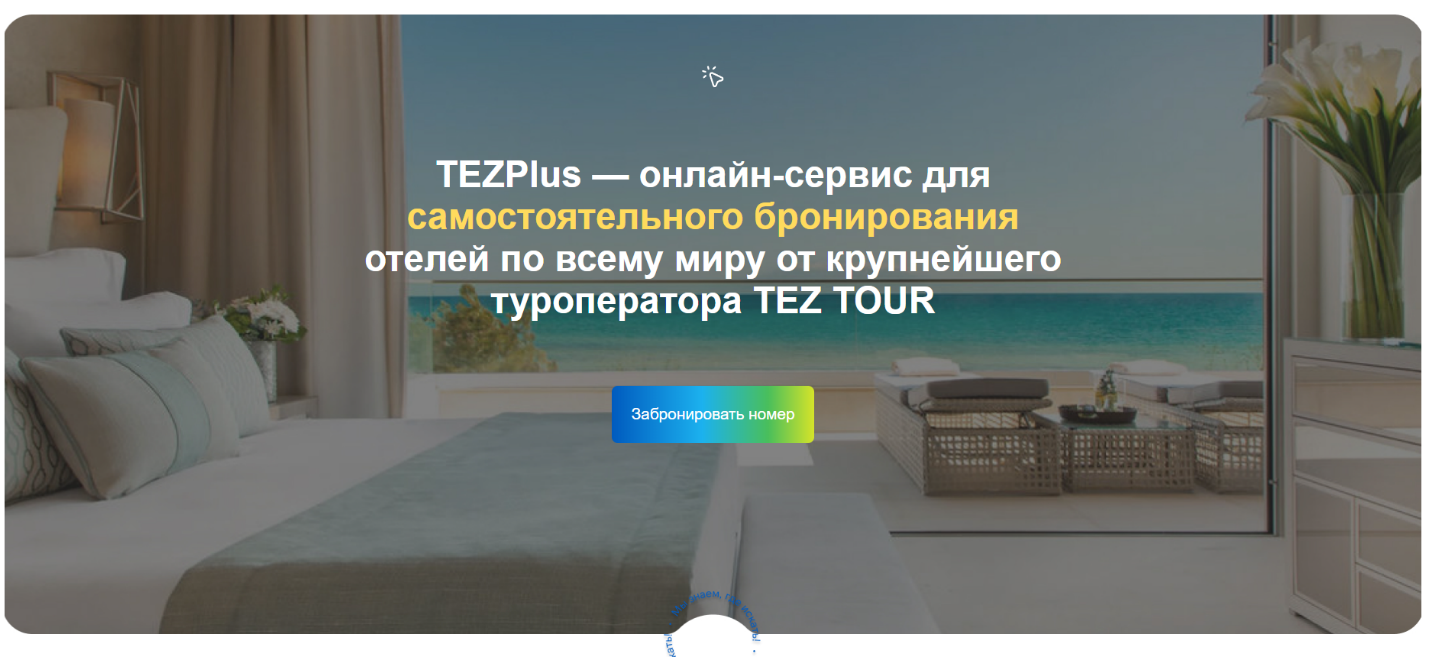


Рисунок 12 - Intro

2. Секция преимуществ

Секция подробно описывает ключевые особенности сервиса



Рисунок 13 - Преимущества

3. Лидеры продаж

В этой секции отображаются популярные направления. Пользователи могут быстро выбрать нужное направление и начать процесс бронирования.

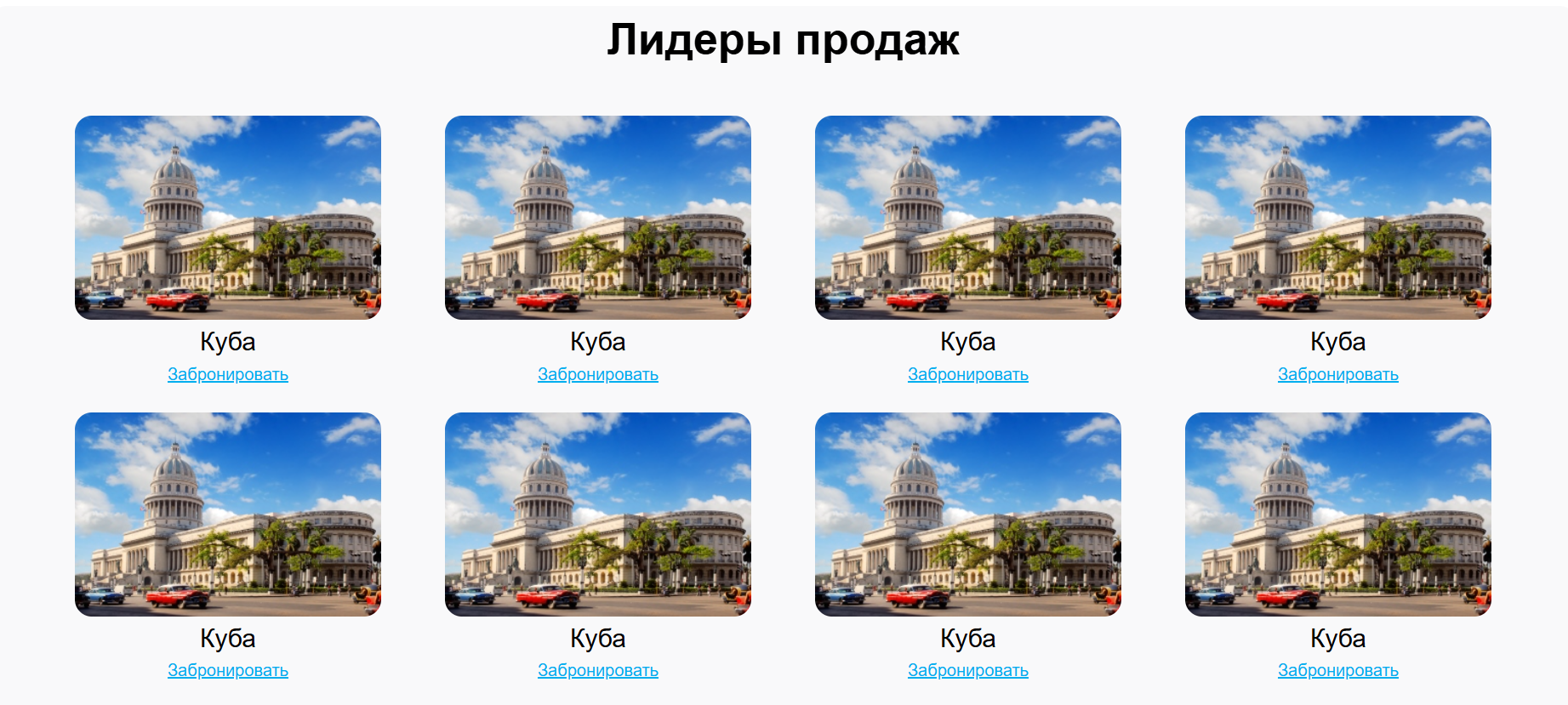


Рисунок 14 - Лидеры продаж

4. Секция "Почему TEZPlus"

Эта секция объясняет, почему пользователи должны выбирать именно этот сервис.



Рисунок 15 - секция Выгоды

5. Наши награды

Секция демонстрирует различные награды и сертификаты, полученные компанией:



Рисунок 16 - Награды

6. Footer.

Сайт «Сервис бронирования отелей» успешно реализует функционал демонстрации преимуществ платформы, популяризации популярных направлений и демонстрации надёжности компании через награды. Благодаря использованию современных технологий и методологий, сайт работает быстро и эффективно как на мобильных устройствах, так и на десктопах.

**ЗАКЛЮЧЕНИЕ**

В ходе прохождения учебной практики была успешно реализована одностраничная верстка сайта «Сервис бронирования отелей», соответствующая заданным требованиям и дизайну из Figma. Были применены современные технологии и подходы к веб-разработке, такие как HTML5, CSS3, препроцессор Sass и методология БЭМ. Также использовался JavaScript для реализации интерактивного меню на мобильных устройствах, что позволило повысить удобство использования сайта.

Проект был реализован с учётом принципов адаптивности, что обеспечило корректное отображение и функциональность на различных устройствах: от мобильных телефонов до десктопных компьютеров. Верстка была выполнена с чётким следованием структуре, заданной в макете, а также с соблюдением семантической разметки и модульной организации кода.

Ключевые результаты практики:

Выполнен полный цикл подготовки проекта: от анализа требований до финальной верстки.

Создана адаптивная и кросс-браузерная верстка сайта.

Реализовано интерактивное меню с использованием JavaScript.

Применена методология БЭМ для упрощения структуры классов и повышения читаемости кода.

Использован препроцессор Sass для упрощения написания и поддержки стилей.

В процессе практики были закреплены навыки работы с современными инструментами веб-разработки, такими как WebStorm, Git и Figma.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Figma. — Режим доступа: <https://www.figma.com/design/vxCIqKhZRnPiWzQ5YNEGzK/%D0%B1%D1%80%D0%BE%D0%BD%D1%8C-%D0%BE%D1%82%D0%B5%D0%BB%D0%B5%D0%B9?node-id=0-1&p=f&t=zGyB3Yp4ANlJ39R7-0>