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 12](#_Toc200010579)

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

[2.4. Диаграмма IDF3 15](#_Toc200010581)

[2.5. Проектирование пользовательского интерфейса 16](#_Toc200010582)

[3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ 18](#_Toc200010583)

[3.1. Подготовка среды разработки и выбор технологий 18](#_Toc200010584)

[3.2. Верстка и стилизация интерфейса 20](#_Toc200010585)

[3.3. Реализация логики с использованием JavaScript 22](#_Toc200010586)

[3.4. Работа сайта 25](#_Toc200010587)

[ЗАКЛЮЧЕНИЕ 30](#_Toc200010588)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc200010589)

**ВВЕДЕНИЕ**

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 верстки сайта был сделан на верстке и создании интерактивного пользовательского интерфейса.

В ходе проекта были использованы следующие инструменты и технологии:

1. Figma — для создания прототипа пользовательского интерфейса;
2. WebStorm — как основная среда разработки;
3. HTML, CSS и Sass — для разметки и стилизации страниц;
4. JavaScript — для реализации базовой интерактивности
5. 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.

Дизайн был взят из готового макета figma. Макет включал в себя главную страницу, страницу отеля, форму бронирования и другие элементы, обеспечивающие логическую целостность и удобство использования.

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**

Проект направлен на создание пользовательского интерфейса веб-сайта, предназначенного для поиска и бронирование отелей. Основной задачей является реализация HTML/CSS верстки, соответствующей современным стандартам веб-разработки, с акцентом на удобство использования и адаптивность под разные устройства.

Цели проекта:

1. Реализовать фронтенд сайта сервиса бронирования отелей по предоставленному макету;
2. Обеспечить корректное отображение сайта на различных устройствах: мобильных телефонах, планшетах и десктопах;
3. Создать интерактивные элементы, включая меню навигации для мобильных устройств;
4. Научиться применять на практике такие технологии, как HTML, CSS, препроцессор Sass и JavaScript;
5. 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 до 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.

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 необходимо выявить, структурировать и документировать функциональные и нефункциональные требования к верстке веб-сайта.

Источники сбора требований:

1. Готовый макет в Figma;
2. Анализ аналогов;
3. Личный опыт работы с веб-интерфейсами и современными стандартами разработки.

Исходя из личного опыта, а также из анализа макета и его аналогов были выявлены требования.

Функциональные требования:

1. Реализация главной страницы сервиса бронирования отелей;
2. Отображение карточек отелей;
3. 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, Edge;
3. Удобство использования. Интерфейс должен быть интуитивно понятным и соответствовать принципам юзабилити;
4. Производительность. Минимальная загрузка ресурсов;
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 - это систематическое изучение доступных материалов, направленное на выявление ключевых элементов и их роли в интерфейсе

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-сайта и определены основные разделы 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;
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 разнырх категорий пользователей при построении интерфейса и функциональных возможностях.

Моделирование – это процесс создания абстрактного представления структуры сайта, его компонентов и взаимодействий между ними.

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.

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:

1. Интерактивное меню для телефонов;
2. Адаптивный дизайн;
3. Все стили должны вынесены в отдельные компоненты согласно БЭМ;
4. 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.

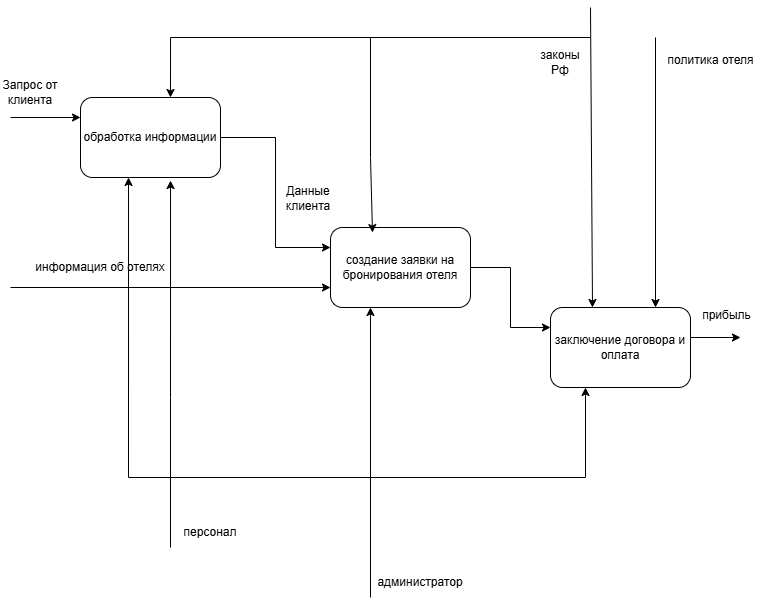


Рисунок 2 - Диаграмма декомпозиций

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.

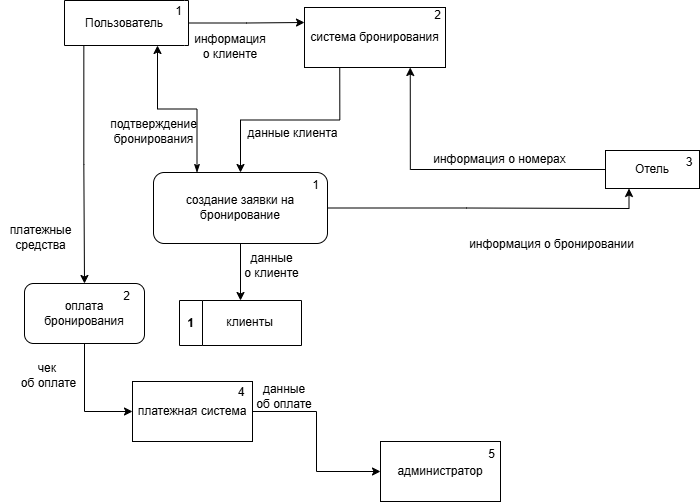


Рисунок 3 - Диаграмма потоков данных

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.

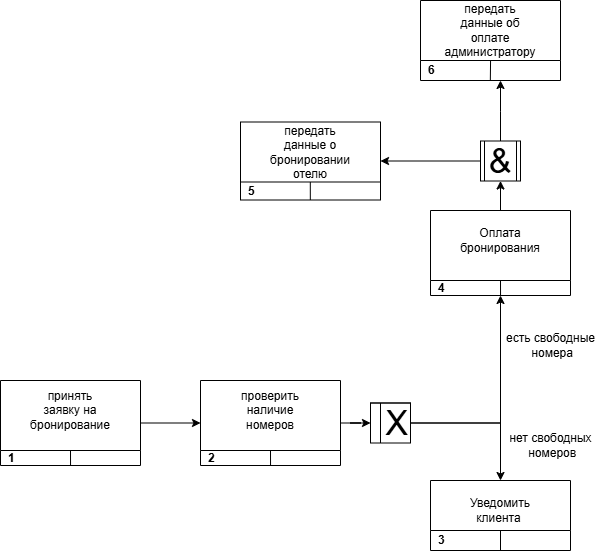


Рисунок 4 - IDF3 диаграмма

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**

На этапе Проектирования пользовательского интерфейса закладывается основа интерфейса между пользователем и системой. Целью данного этапа является создание интуитивно понятный и привлекательный интерфейс.

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:

1. Кнопки — разработаны различные стили кнопок (основная, вторичная, компактная), учитывающие состояние наведения и клика;
2. Карточки направлений — унифицированные блоки с изображением, заголовком и ценой, используемые в секции лидеров продаж;
3. Иконки — плоский стиль, единые размеры и цветовое оформление для целостности дизайна;
4. Текстовые блоки — оформлены с учётом иерархии: заголовки, подзаголовки, основной текст;
5. 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:

* 1. Основной цвет: #015BC1;
  2. Акцентный цвет: # FFDB5D;
  3. Фоновые цвета: #F9F9FA и #FFF;
  4. 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**

Перед версткой любого проекта необходимо организовать рабочую среду и выбрать подходящие технологии

На данном этапе для реализации проекта «сервис бронирования отелей» была настроена локальная среда разработки, включающая современные инструменты и технологии:

* 1. Среда разработки: WebStorm — выбрана как основной редактор кода благодаря своей мощной функциональности, встроенной поддержке современных веб-технологий и глубокой интеграции с фреймворками и инструментами;
  2. Система контроля версий: Git — используется для управления изменениями в коде;
  3. Репозиторий: удалённый через GitHub;
  4. Менеджер пакетов: npm — применялся для установки и управления зависимостями;
  5. Браузеры для тестирования: Google Chrome, Mozilla Firefox.

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 |
| Авто дополнение и навигация | Умное авто дополнение, 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:

1. Встроенная поддержка всех ключевых веб-технологий без необходимости установки дополнительных плагинов.
2. Интеллектуальные подсказки, проверка ошибок и рефакторинг кода.
3. Удобство работы с проектами большой сложности.
4. Глубокая интеграция с системами контроля версий, сборщиками и Фреймворками.
5. Поддержка методологии БЭМ и препроцессоров, таких как Sass.

При разработке сайта были выбраны следующие технологии:

1. HTML5 / CSS3 — для создания структуры и оформления страниц.
2. Sass (SCSS) — препроцессор CSS, позволяющий организовать стили модульно и использовать переменные, миксины и другие удобства.
3. JavaScript — язык программирования для реализации интерактивных элементов: меню, модальных окон, форм.
4. 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**

После завершения этапов проектирования и подготовки среды следует этап верстки

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:

1. HTML — для структурирования содержимого страниц;
2. CSS3/ SCSS — для описания внешнего вида элементов;
3. Методология БЭМ — для организации классов и файловой структуры проекта;
4. Для обеспечения модульности, читаемости и масштабируемости кода была применена методология Блок — Элемент — Модификатор (БЭМ).

Примеры использования:

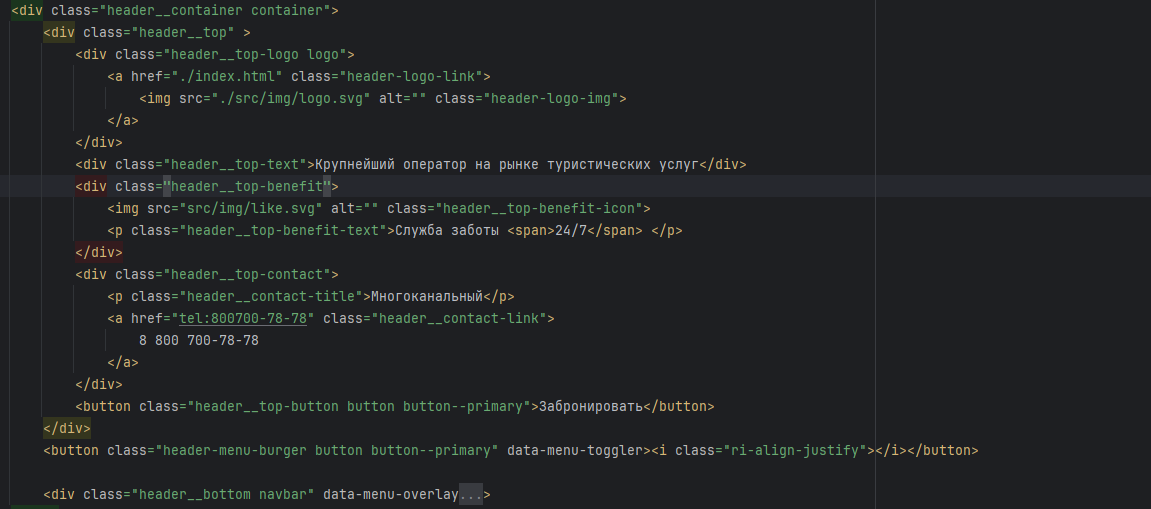


Рисунок 5 - пример названий классов по БЭМ

Использование БЭМ позволило:

1. Упростить понимание взаимосвязей между элементами;
2. Избежать конфликтов имен классов;
3. 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:

1. CSS-медиазапросы;
2. Flexbox и Grid Layout;
3. Относительные единицы измерения (rem, %).

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

1. Вложенные селекторы;
2. Переменные ($color-primary, $font-size-base);
3. Миксины (@mixin media(), @mixin btn-style());
4. Импорт файлов по структуре БЭМ.

Описание структуры:

1. \_fonts.scss: Настройки шрифтов и их импорта;
2. \_functions.scss: Пользовательские функции для SCSS (например, вычисление размеров);
3. \_globals.scss: Глобальные стили, такие как общие настройки body.
4. \_utils.scss: Утилитарные классы (например, .container.);
5. \_vars.scss: Хранение переменных (цвета, шрифты, отступы);
6. 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;
7. \_breakpoints.scss: Миксин для медиазапросов;
8. \_fonts-face. scss: Настройки шрифтов через @font-face;
9. \_layouts.scss: Миксины для различных лэйаутов (например, гриды или flexbox);
10. 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:

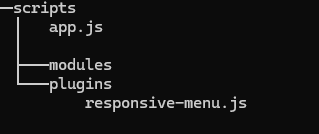


Рисунок 6 - структура файлов скриптов

Описание файлов:

1. plugins/responsive-menu.js — содержит логику открытия/закрытия мобильного меню;
2. app.js — главный файл, в котором подключаются и инициализируются компоненты.

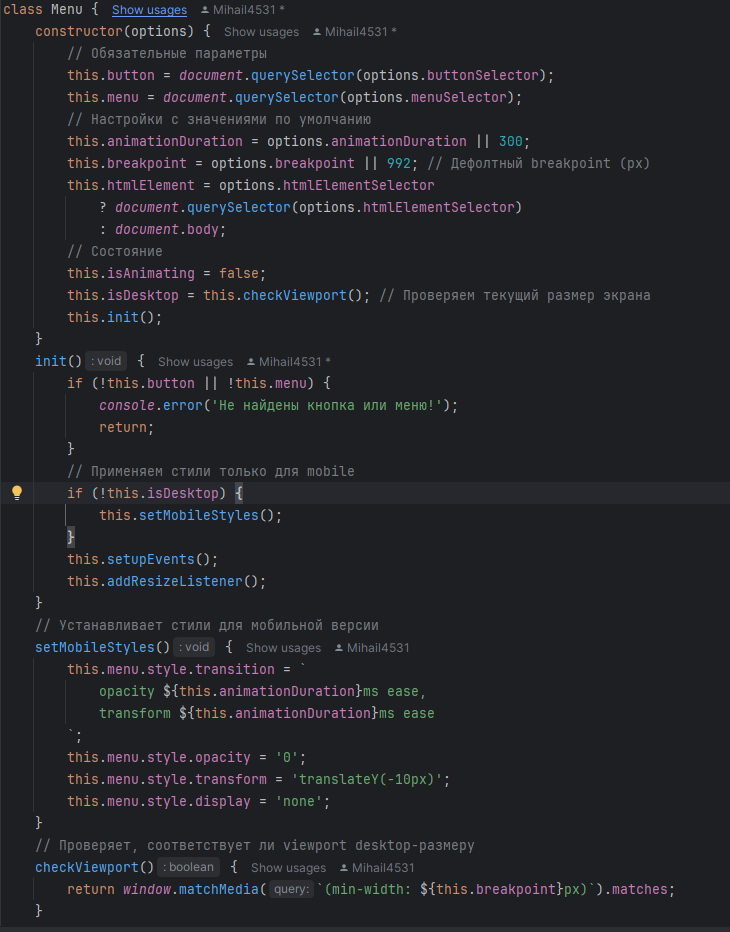


Рисунок 7 – JavaScript responsive-menu

Основные функции для интерактивного меню:

1. Открытие меню (openMenu): Показывает меню;
2. Закрытие меню (closeMenu): Скрывает меню.;
3. Адаптивность: Автоматически переключается между версиями меню для десктопа и мобильных устройств;
4. Обработка событий клика: Открывает/закрывает меню при нажатии на кнопку или клике вне области меню.

Описание функционала:

Конструктор

Инициализирует объект класса Menu с необходимыми элементами и настройками:

Параметры конструктора:

1. buttonSelector: CSS-селектор кнопки бургера;
2. menuSelector: CSS-селектор меню;
3. animationDuration: Длительность анимации (по умолчанию 300 мс);
4. breakpoint: Разрешение экрана, выше которого меню становится десктопным;
5. htmlElementSelector: Элемент, который будет блокировать скроллинг при открытом меню;

Методы управления состоянием меню

1. Открытие меню (openMenu) - Показывает меню с анимацией и блокирует скроллинг страницы, чтобы пользователь не мог прокручивать контент во время открытия меню;
2. Закрытие меню (closeMenu) -Скрывает меню с анимацией и разблокирует скроллинг страницы после закрытия меню;
3. Обработка событий (setupEvents) - Клик на кнопку бургера и Клик вне меню: Закрывает меню, если оно открыто. Нажатие клавиши Esc - закрывает меню, если оно открыто;
4. 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:

1. Header — содержит логотип, кнопку бургера (для мобильных устройств) и главное меню (для десктопных);
2. Intro — приветственное сообщение;
3. Секция преимуществ — демонстрирует ключевые особенности сервиса;
4. Лидеры продаж — отображает популярные направления с возможностью быстрой брони;
5. Секция выгоды — объясняет выгоды использования платформы;
6. Наши награды — демонстрирует достижения компании;
7. Отзывы - показывает отзывы пользователей.

Функциональность

* 1. Intro. На главной странице расположено приветственное сообщение c кратким описанием возможностей сервиса. Здесь же находится кнопка "Забронировать номер", которая направляет пользователя к форме бронирования;

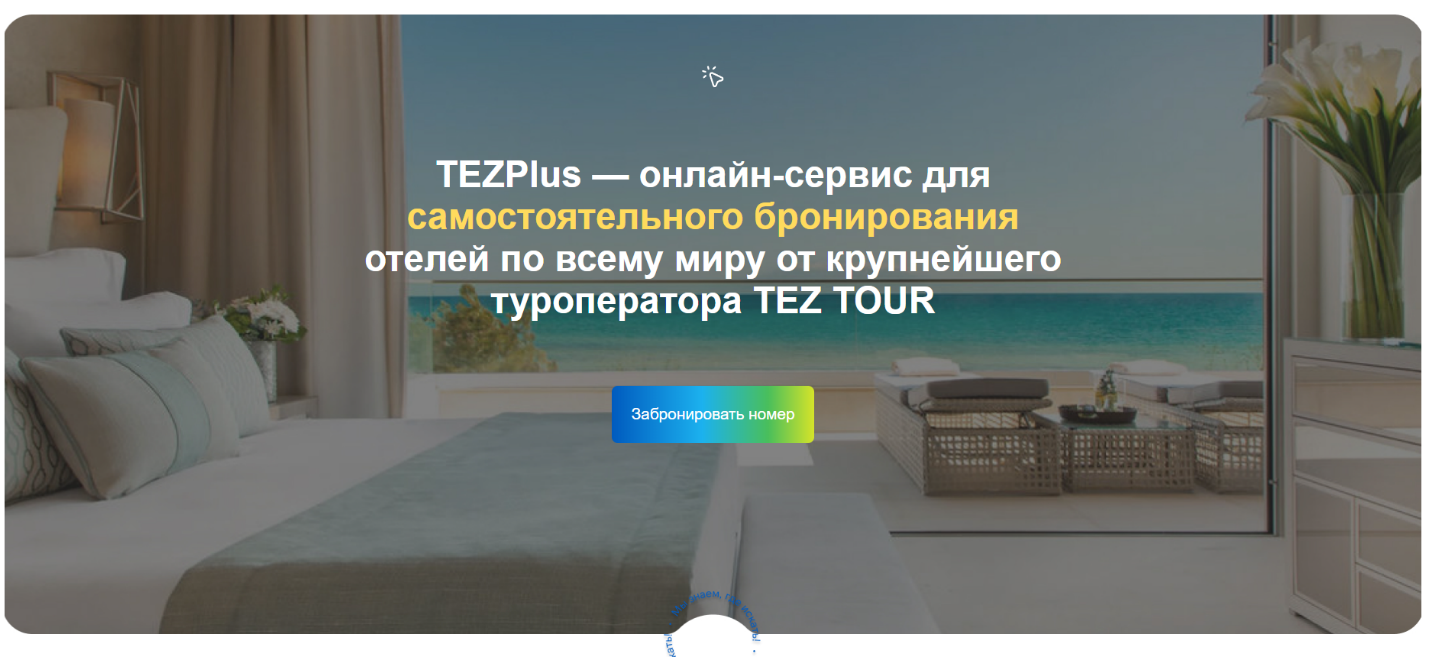


Рисунок 8 - Intro

* 1. Секция преимуществ. Секция, на которой пользователь может ознакомиться с ключевыми особенностями платформы такие как:

1. возможный выбор тарифа;
2. удобная и безопасная система бронирования;
3. доступ к 1000000 отелей;
4. более 1000 экскурсий на русском языке.



Рисунок 9 - Преимущества

* 1. Лидеры продаж. В этой секции отображаются популярные направления. Пользователи могут быстро выбрать нужное направление и начать процесс бронирования. Содержит:

1. Фотографии отелей;
2. Название стран;
3. Кнопки быстрого перехода к бронированию.

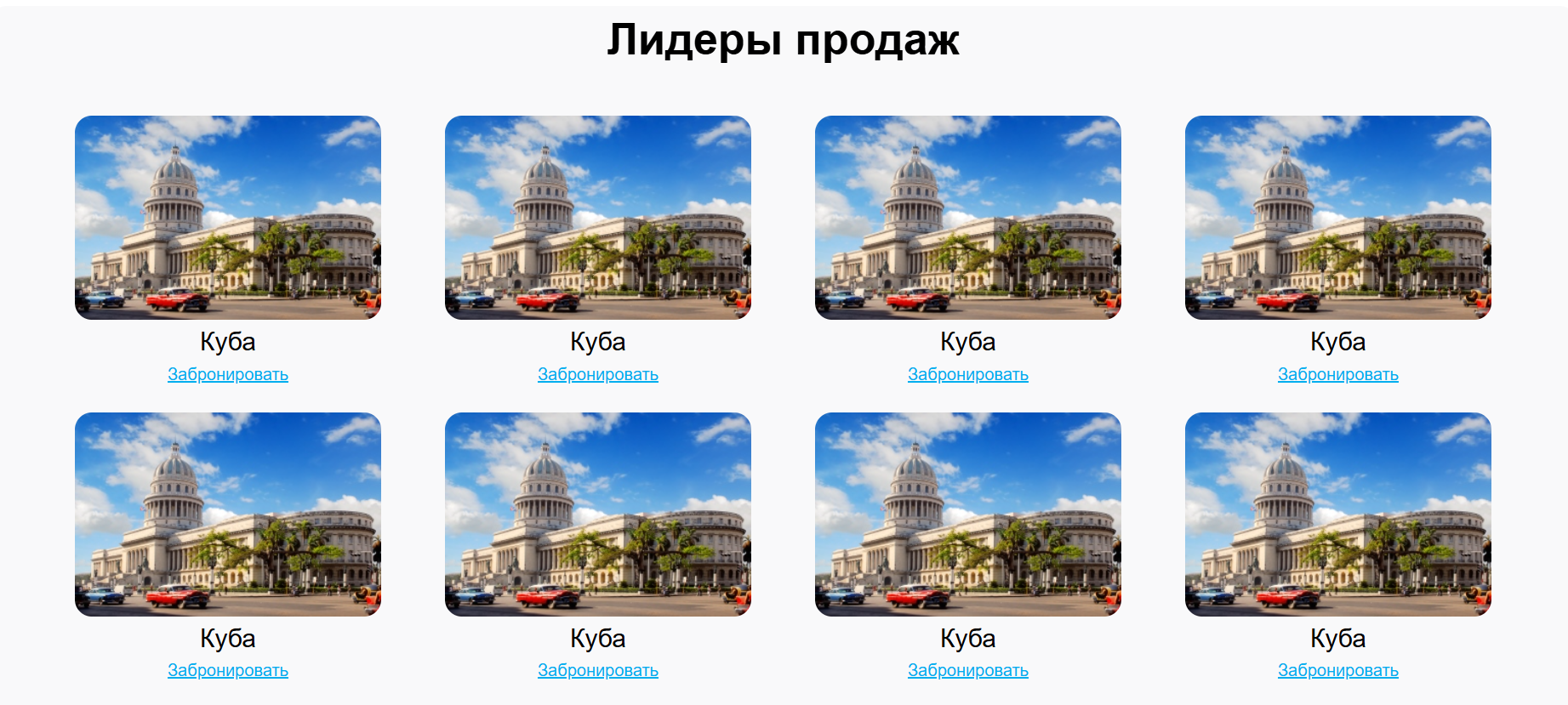


Рисунок 10 - Лидеры продаж

1. Секция выгоды. Секция объясняет, почему пользователи должны выбирать именно этот сервис:
2. Оплата картой банка РФ;
3. Выгодные цены;
4. Отели в любой точке мира;
5. Гибка оплата;
6. Удобный поиск и умные фильтры;
7. Служба заботы 24/7.



Рисунок 11 - секция Выгоды

1. Наши награды. Секция демонстрирует различные награды и сертификаты и победы, полученные компанией:
2. Победитель премии “Туроператор года”;
3. Сертифицированный агент авиакомпаний.



Рисунок 12 - Награды

1. 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 проекта 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 на различных устройствах. Верстка была выполнена с чётким следованием структуре, заданной в макете, а также с соблюдением семантической разметки и модульной организации кода.

Ключевые результаты:

1. Выполнен полный цикл подготовки проекта: от анализа требований до финальной верстки.
2. Создана адаптивная верстка сайта.
3. Реализовано интерактивное меню с использованием JavaScript.
4. Применена методология БЭМ для упрощения структуры классов и повышения читаемости кода.
5. Использован препроцессор Sass для упрощения написания и поддержки стилей.
6. В процессе 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>;
2. GitHub репозиторий - Режим доступа: <https://github.com/Arnd25/hotel>;
3. WebStorm документация Режим доступа: <https://www.jetbrains.com/ru-ru/webstorm/learn/>;
4. MDN Web Docs – справочная информация по html , css и javascript Режим доступа: <https://developer.mozilla.org/ru/docs/Web>;
5. БЭМ документация Режим доступа: <https://ru.bem.info/methodology/>;
6. Sass документация: Режим доступа: <https://sass-scss.ru/documentation/>;
7. Habr – статьи по веб-разработке. Режим доступа: https://habr.com/ru/articles/;
8. Stack Overflow – форум вопросов и ответов по программированию Режим доступа: <https://stackoverflow.com>
9. Dev.to – платформа для разработчиков, блоги и обсуждения. Режим доступа: <https://dev.to>;
10. W3schools – справочный ресурс по веб-технологиям. Режим доступа: <https://w3schools.com>.