

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

« ____ » _____ 2023 г.

Разработка веб-сайта по оказанию туристических услуг

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 09.03.04.2023.308-330.ВКР

Научный руководитель,
профессор кафедры СП, д.г.н.,
к.ф.-м.н.

_____ С.М. Абдуллаев

Автор работы,
студент группы КЭ-404

_____ И.В. Игнатьев

Ученый секретарь
(нормоконтролер)

_____ И.Д. Володченко

« ____ » _____ 2023 г.

Челябинск, 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
06.02.2023 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы бакалавра
студенту группы КЭ-404
Игнатьеву Илье Валерьевичу,
обучающемуся по направлению
09.03.04 «Программная инженерия»

- 1. Тема работы** (утверждена приказом ректора от 25.04.2023 г. № 753-13/12)
Разработка веб-сайта по оказанию туристических услуг.
- 2. Срок сдачи студентом законченной работы:** 05.06.2023 г.
- 3. Исходные данные к работе**
 - 3.1. WordPress Support. [Электронный ресурс] URL: wordpress.org/support/ (дата обращения: 02.02.2023 г.).
 - 3.2. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – М.: Изд-во Наука и Техника, 2016. – 768 с.
 - 3.3. Изучение SQL, с помощью phpmyadmin. [Электронный ресурс] URL: phpmyadmin.net (дата обращения: 05.02.2023 г.).
- 4. Перечень подлежащих разработке вопросов**
 - 4.1. Анализ предметной области.
 - 4.2. Провести обзор аналогов сайтов туристических агентств.
 - 4.3. Проектирование веб-сайта.

4.4. Разработка веб-сайта.

4.5. Провести тестирование веб-сайта.

5. Дата выдачи задания: 06.02.2023 г.

Научный руководитель,
профессор кафедры СП, д.г.н., к.ф.-м.н.

С.М. Абдуллаев

Задание принял к исполнению

И.В. Игнатьев

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1. Описание предметной области	7
1.2. Сравнительный анализ аналогов.....	8
1.3. Средства разработки.....	10
2. ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА.....	13
2.1. Анализ требований	13
2.3. Проектирование базы данных	16
2.3. Диаграмма компонентов	21
3. РЕАЛИЗАЦИЯ СИСТЕМЫ	23
3.1. Реализация интерфейса	23
3.2. Реализация телеграм-бота для обратной связи.....	27
3.3. Реализация текстового телеграм-бота	29
4. ТЕСТИРОВАНИЕ	32
4.1. Функциональное тестирование	32
4.2. Проверка установки соединения с базой данных.....	36
ЗАКЛЮЧЕНИЕ	38
ЛИТЕРАТУРА.....	39

ВВЕДЕНИЕ

Актуальность

Привлечение новых клиентов является основной задачей для активно развивающихся туристических фирм. В наше время собственный веб-сайт стал самым простым и эффективным способом представить себя и свои услуги потенциальным клиентам.

При выборе туристического агентства многие путешественники обращаются к Интернет-ресурсам. Они хотят изучить ассортимент предлагаемых услуг, не выходя из дома и не совершая лишних телефонных звонков. В таких случаях веб-сайт может выступать в роли посредника, связывающего продавца и покупателя. Клиент сможет получить интересующую его информацию и выбрать понравившуюся фирму среди сотен других. Собственный сайт становится не только полезным ресурсом для туристов, но и необходимым инструментом в продвижении компании на рынке туристических услуг.

Кроме того, собственный веб-сайт является важным инструментом в продвижении компании на рынке туристических услуг. Он позволяет привлекать новых клиентов, улучшать узнаваемость бренда и укреплять позиции на рынке. С помощью сайта можно проводить маркетинговые кампании, размещать специальные предложения и акции, а также взаимодействовать с клиентами через различные онлайн-каналы связи.

Таким образом, наличие собственного веб-сайта для туристической фирмы является не только необходимостью в современном мире, но и мощным инструментом для привлечения новых клиентов и успешного продвижения на рынке. Он позволяет предоставить информацию, укрепить доверие и создать положительное представление о компании у потенциальных путешественников.

Постановка задачи

Целью выпускной квалификационной работы является разработка веб-сайта по оказанию туристических услуг. Для достижения поставленной цели необходимо решить следующие задачи:

- 1) проанализировать предметную область;
- 2) разработать функциональные и нефункциональные требования к веб-сайту;
- 3) выполнить проектирование веб-сайта;
- 4) реализовать веб-сайт;
- 5) провести тестирование веб-сайта.

Структура и содержание работы

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 40 страниц, объем списка литературы – 15 источников.

В первой главе «Анализ предметной области» проводится анализ предметной области, обзор аналогичных сайтов турагентств и средств разработки, которые потребуются для реализации поставленных задач.

Вторая глава «Проектирование веб-сайта» посвящена разработке функциональных и нефункциональных требований к системе, представляются диаграмма вариантов использования, диаграмма компонентов и их описание, составлена схема базы данных.

В третьей главе «Реализация веб-сайта» описывается архитектура проекта, процесс реализации веб-сайта, его основные функции, а также представлены листинги кода.

В четвертой главе «Тестирование» содержится результат функционального тестирования и проверка соединения с базой данных.

В заключении приведены результаты проделанной работы.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание предметной области

Реклама туристической компании и продвижение туристических услуг стали особенно необходимы в последнее десятилетие, в связи с быстрым ростом туристического рынка и появлением множества туроператоров и турагентств. Все сложнее становится рекламировать туристские услуги и туристические компании, все больше требуется как финансовых ресурсов, так и человеческих усилий для привлечения клиентов, и продажи турпродукта.

При этом современные тенденции в рекламе свидетельствуют о перемене акцента с рекламы в традиционных средствах массовой информации на онлайн-продвижение. Согласно данным исследований доля затрат на рекламу в традиционных рекламоносителях (на телевидении, радио, в прессе и на улицах) с каждым годом уменьшается, уступая место рекламе в сети Интернет.

В настоящее время принято считать, что всего две площадки для размещения рекламы способны ежедневно «доставать» потребителей – это наружная реклама и Интернет. При этом Интернет-реклама туристического агентства является, по мнению многих специалистов, одним из самых эффективных инструментов привлечения клиентов в туристическую компанию. Это происходит из-за того, что более 60% клиентов предпочитают искать информацию о предстоящем отдыхе в Интернете.

Хотелось бы отметить, что основой для успешной рекламной кампании в Интернете является та информационная база, к которой в конечном итоге в идеале должен обращаться потребитель, чтобы получить информацию об агентстве. Этой информационной базой должен стать официальный сайт туристического агентства.

Для того, чтобы создать конкурентоспособный и привлекающий веб-сайт, необходимо проанализировать сайты конкурентов в данной области, а также выявить достоинства и недостатки уже существующих сайтов. Рассмотрим существующие аналогичные сайты с оказанием туристических услуг. В качестве аналогов были рассмотрены 3 сайта, которые позволят выявить определенные особенности.

1.2. Сравнительный анализ аналогов

Веб-сайт туристического агентства «Травелата»

В базе «Травелата» есть предложения более 150 туроператоров. Интерфейс понятен на интуитивном уровне, фильтры стандартные, благодаря им на «Травелата» можно быстро найти подходящие туры. Уникальное предложение портала - отдельные фильтры поиска для семей с детьми. Можно сделать акцент на услугах няни, гаджетах для малышей, мини-клубе, качестве детских развлечений. Также можно просто посмотреть все горящие туры на одном сайте на ближайшую неделю, если направление отдыха не принципиально. Скриншот главной страницы веб-сайта турагентства «Травелата» представлен на рисунке 1.

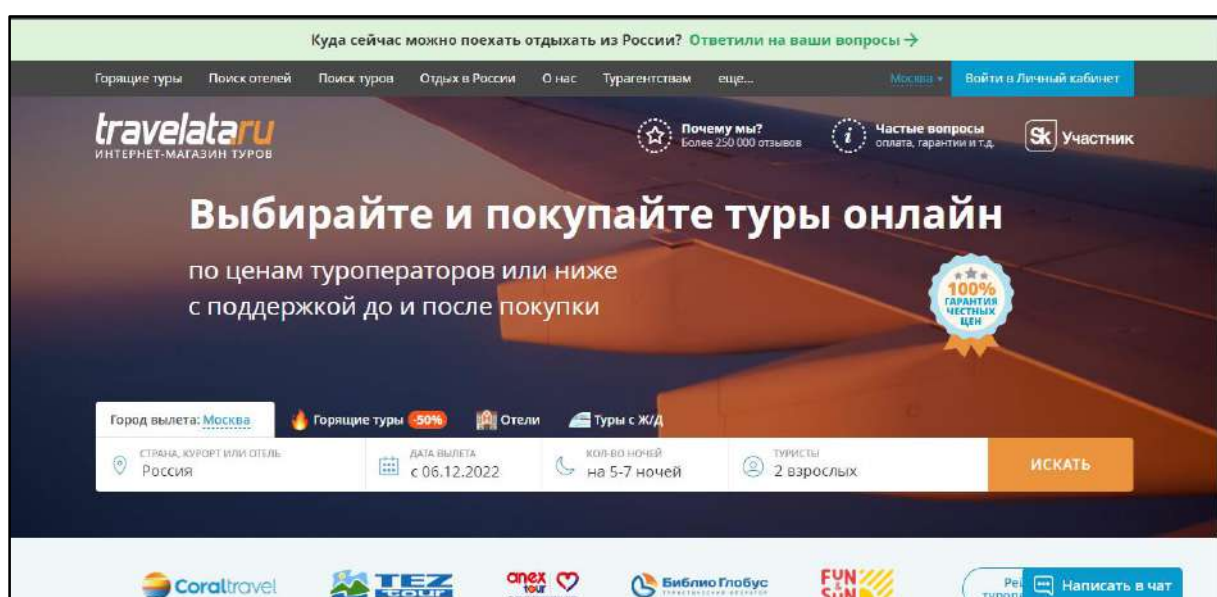


Рисунок 1 – Скриншот веб-сайта «Травелата»

Веб-сайт турагентства «Лэвел Трэвел»

Один из лидеров среди сайтов дешевых туров. Поисковик предлагает около 80 туристических направлений. «Лэвел Трэвел» аккумулирует горящие предложения и туры со скидками различных туроператоров в онлайн-режиме. Турист может задать такие параметры, как: даты, локация, желательная стоимость, звездность отеля, вариант питания, количество гостей. Учитываются даже такие пожелания, как отсутствие визы, детский отдых, наличие пляжа рядом, условия для серфинга, насыщенная развлекательная программа и т.д. Поисковая система сортирует подходящие варианты по стоимости или дате вылета. Скриншот главной страницы веб-сайта турагентства «Лэвел Трэвел» представлен на рисунке 2.

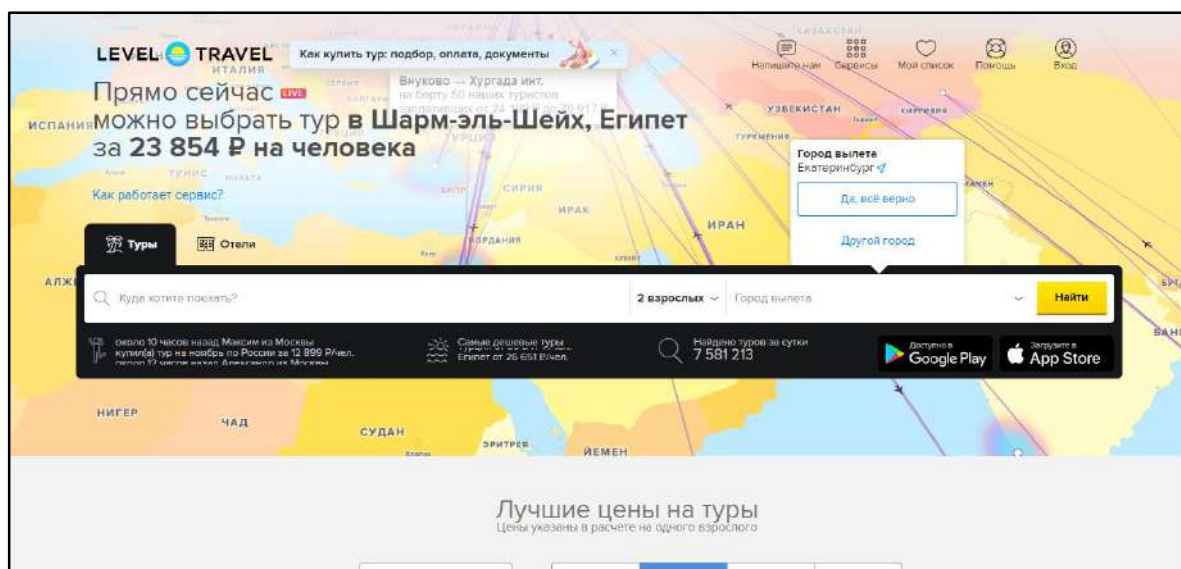


Рисунок 2 – Скриншот веб-сайта «Лэвел Трэвел»

Веб-сайт турагентства «Онлайн Турс»

Поисковик туров по всем туроператорам «Онлайн Турс» начал свою работу почти 10 лет назад – в 2012 году. Его партнеры – более 130 проверенных операторов. Именно на надежности этот поиск делает особый акцент. Задав свои параметры, вы увидите самые дешевые туры, которые в этот момент имеются на рынке. Учитывая большое количество предложений, туры со скидкой 30-50% есть всегда, если повезет - поймаете скидку до

70%. Оплатить понравившийся тур можно сразу же онлайн. Сначала нужно перевести 50%, а остальную сумму – в течение 3 дней. Скриншот главной страницы веб-сайта турагентства «Онлайн Турс» представлен на рисунке 3.

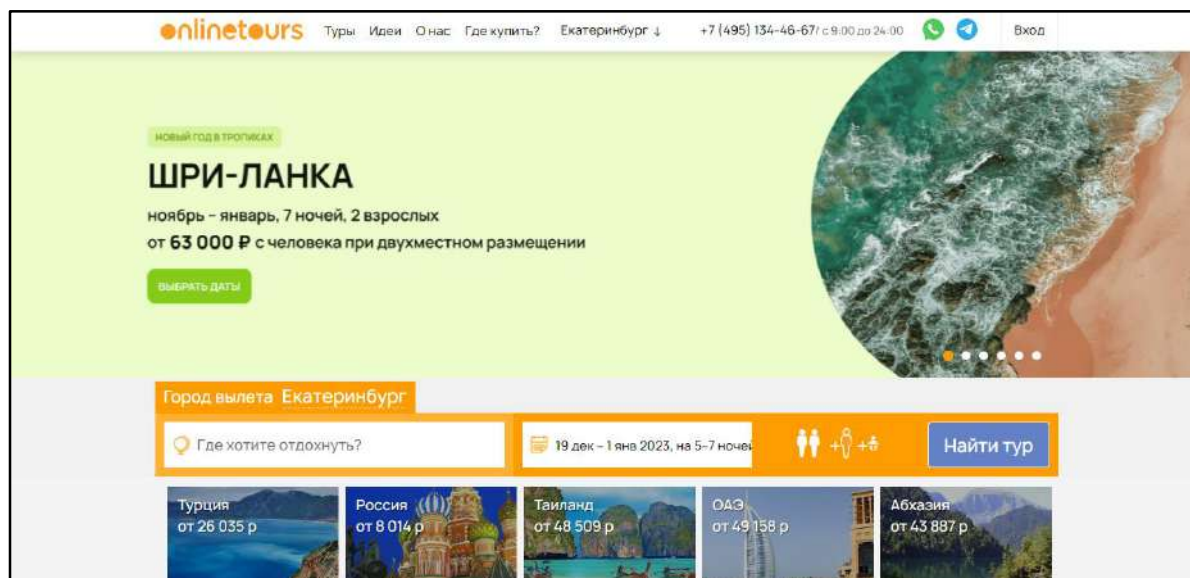


Рисунок 3 – Скриншот веб-сайта «Онлайн Турс»

1.3. Средства разработки

CMS (Content Management System) – это программное обеспечение, которое позволяет управлять контентом: создавать и публиковать фотографии, текст, размещать виджеты, менять дизайн и другие элементы, управлять функционалом. CMS часто называют движком сайта.

На CMS создают удобные сайты с разнообразными функциями: чтобы обрабатывать заказы, анализировать продажи и выполнять другие процессы.

CMS помогает в трех основных направлениях работы:

- 1) создает UX/UI, то есть пользовательский интерфейс: разделы сайта, содержимое и внешний вид страниц без работы в коде;
- 2) хранит файлы и изображения, детали дизайна и другие элементы, которые размещены на сайте;
- 3) содержит базу данных – информацию о наполнении и профилях пользователей.

CMS обычно используется для управления корпоративным контентом (ECM) и управления веб-контентом (WCM). ECM обычно поддерживает нескольких пользователей в среде совместной работы путем интеграции управления документами, управления цифровыми активами и хранения записей. В отличие от этого, WCM является совместной разработкой для веб-сайтов и может включать в себя текст и встраиваемую графику, фотографии, видео, аудио, карты и программный код, которые отображают контент и взаимодействуют с пользователем. ECM обычно включает функцию WCM. Заметим, что существует два типа установки CMS: локальная и облачная. Облачная CMS размещается в среде поставщика. При таком подходе программное обеспечение CMS не может быть изменено клиентом.

Примерами известных облачных CMS являются SquareSpace, Contentful, WordPress.org, Webflow, Ghost и WIX. Локальная установка означает, что программное обеспечение CMS может быть установлено на сервере. Этот подход обычно используется, когда хотят гибкости в своей настройке. Популярными локальными CMS являются Wordpress.com, Drupal, Joomla, ModX и другие. В нашем случае очевидно выбрать серверную Wordpress.org, как популярную CMS, имевшую в 2021 году 42% установок, в сравнении следующей за ней CMS Drupal, которая имеет 15% установок.

Кроме того, в ходе разработки веб-сайта, будет использован популярный плагин Elementor и плагин MetForm. Elementor - это мощный инструмент для создания и редактирования страниц на основе WordPress. С его помощью можно создавать макеты страниц, настраивать внешний вид и стиль элементов, а также добавлять разнообразные функциональные возможности, чтобы обеспечить более удобное и интерактивное взаимодействие с посетителями веб-сайта.

При разработке веб-сайта, для его локальной версии обычно используется локальный сервер, на котором хранится вся необходимая информация. Одним из наиболее популярных локальных серверов является Open Server. Open Server представляет собой специально созданный для разработчиков локальный веб-сервер. Он широко применяется для разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях.

Базовый пакет локального сервера Open Server содержит:

- 1) веб-сервер Apache или Nginx: эти серверы обеспечивают обработку HTTP-запросов и доставку веб-страниц клиентам;
- 2) языки программирования PHP, Perl, Ruby: эти языки позволяют создавать динамические веб-страницы и взаимодействовать с базами данных;
- 3) системы управления базами данных MySQL, PostgreSQL, MongoDB: эти системы позволяют хранить и управлять данными, которые используются на веб-сайте;
- 4) PHP-фреймворки Symfony, Laravel, Yii, CodeIgniter, Zend Framework: эти фреймворки предоставляют набор инструментов и функциональность для ускорения процесса разработки веб-приложений;
- 5) дополнительные модули Memcached, Xdebug, PhpMyAdmin: эти модули облегчают отладку кода, управление базами данных и улучшают производительность веб-приложений.

Благодаря этим компонентам, разработчики могут создавать и разрабатывать веб-приложения на различных языках программирования, использовать различные системы управления базами данных и PHP-фреймворки для более эффективного и быстрого процесса разработки.

Вывод по первой главе

В первой главе было рассмотрено описание предметной области, обзор существующих аналогов и произведен выбор средств реализации.

2. ПРОЕКТИРОВАНИЕ ВЕБ-САЙТА

2.1. Анализ требований

Перед разработкой любого программного продукта необходимо определить и сформулировать требования к нему. Эти требования можно разделить на две основные группы: функциональные и нефункциональные.

Функциональные требования описывают, какие конкретные функции и возможности должны быть реализованы в процессе работы программного продукта. Они определяют, какие действия может совершать пользователь и как система должна на них реагировать. Например, функциональные требования могут включать такие аспекты, как вход и регистрацию пользователей, обработку данных, генерацию отчетов, взаимодействие с внешними системами и другие функциональности, которые необходимы для достижения целей программного продукта.

Нефункциональные требования описывают свойства и ограничения, которым должна соответствовать разработка. Они описывают «каким образом» будет реализована функциональность системы и определяют её качество. Нефункциональные требования могут включать такие аспекты, как производительность, безопасность, надежность, доступность, удобство использования, масштабируемость, поддержка различных платформ и технологий, а также соответствие стандартам и законодательству.

Важно учесть обе группы требований при разработке программного продукта, так как функциональные требования определяют его функциональность и полезность для пользователей, а нефункциональные требования гарантируют качество и соответствие разработки заданным стандартам и ожиданиям пользователей.

Веб-сайт должен отвечать следующим функциональным требованиям:

- 1) пользователь должен иметь возможность просмотра туров на веб-сайте;

- 2) пользователь должен иметь возможность оставления обратной связи на веб-сайте;
- 3) пользователь должен иметь возможность ознакомиться с информацией о турах в текстовом телеграм-боте;
- 4) пользователь должен иметь возможность оставить обратную связь в телеграм-боте сайта;
- 5) администратор должен иметь возможность создавать, редактировать и удалять информацию о турах.

Веб-сайт должен отвечать следующим нефункциональным требованиям:

- 1) корректное отображение веб-сайта при открытии его в различных браузерах, а именно в таких, как: Яндекс Браузер, Google Chrome, Opera, Mozilla Firefox;
- 2) удобная и понятная навигация;
- 3) загрузка веб-сайта в течении 3–5 секунд;
- 4) веб-сайт должен быть создан с помощью WordPress;
- 5) веб-сайт должен использовать локальный сервер Open Server;
- 6) веб-сайт должен использовать плагин Elementor;
- 7) веб-сайт должен использовать плагин MetForm.

2.2. Диаграмма вариантов использования

На основе сформулированных функциональных и нефункциональных требований была составлена диаграмма вариантов использования.

Диаграмма вариантов использования (англ. use-case diagram) – это диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

На диаграммах вариантов использования изображаются актеры и варианты использования, между которыми существуют отношения.

Актёр – это роль объекта вне системы, который прямо взаимодействует с ее частью – конкретным элементом.

Прецедент – это описание последовательности действий (или нескольких последовательностей), которые выполняются системой и производят для отдельного актёра видимый результат.

В ходе анализа разрабатываемого веб-сайта были выявлены основные варианты использования, представленные на рисунке 4.

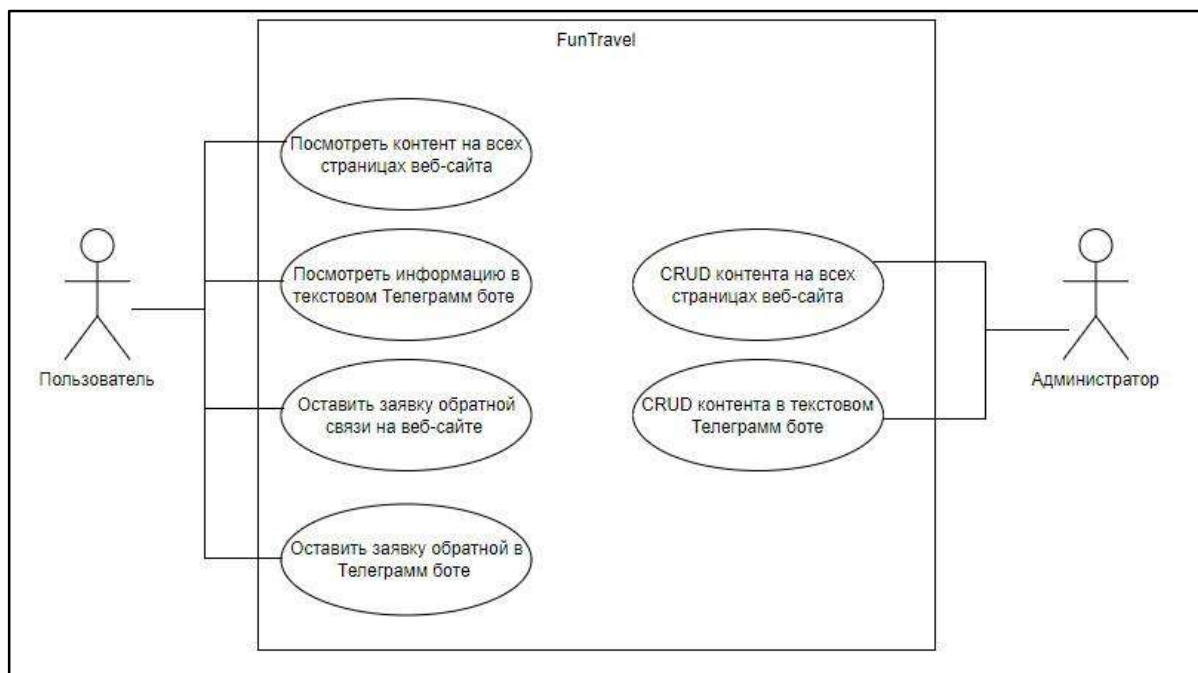


Рисунок 4 – Диаграмма вариантов использования

В проектируемой системе представлены 2 актёра.

1. Пользователь – любой человек, который зашел на сайт.
2. Администратор – редактор веб-сайта, который имеет доступ к редактированию информации на сайте.

Краткое описание вариантов использования представлены ниже.

1. Посмотреть контент на всех страницах веб-сайта – пользователь может ознакомиться со всем контентом, что есть на веб-сайте.
2. Посмотреть информацию в текстовом телеграм-боте – пользователь может ознакомиться с турагентством в телеграм-боте.

3. Оставить заявку обратной связи на веб-сайте – возможность любому пользователю сайта оставить заявку в форме обратной связи.
4. Оставить заявку обратной связи в телеграм-боте – возможность любому пользователя оставить заявку обратной связи в телеграм-боте.
5. CRUD контента на всех страницах веб-сайта – возможность создавать, просматривать, редактировать и удалять любой контент на сайте.
6. CRUD контента в текстовом телеграм-боте – возможность создавать, просматривать, редактировать и удалять любой контент в телеграм-боте.

2.3. Проектирование базы данных

Для реализации веб-сайта часто требуется создание и использование базы данных. Платформа WordPress предоставляет возможность работать с различными системами управления базами данных (СУБД).

Система управления базами данных (СУБД) представляет собой набор инструментов, которые облегчают управление базами данных. Она позволяет добавлять, удалять, фильтровать и находить элементы, а также изменять структуру базы данных и создавать резервные копии. СУБД является прослойкой между самой базой данных и пользовательскими запросами к ней. На рисунке 5 представлена визуальная схема взаимодействия СУБД с базой данных и пользователями.

Основные функции СУБД:

- 1) создание баз данных, изменение, удаление и объединение их по заданным признакам;
- 2) хранение данных, в том числе больших массивов, в структурированном виде и нужном формате;
- 3) защита данных от несанкционированного доступа и нежелательных изменений путем предоставления различных уровней доступа разным группам пользователей;

4) выгрузка и сортировка данных с помощью SQL-запросов, позволяющих выбирать нужные данные и применять фильтры;

5) поддержка целостности баз данных, включая резервное копирование и восстановление данных после сбоев или потери информации.

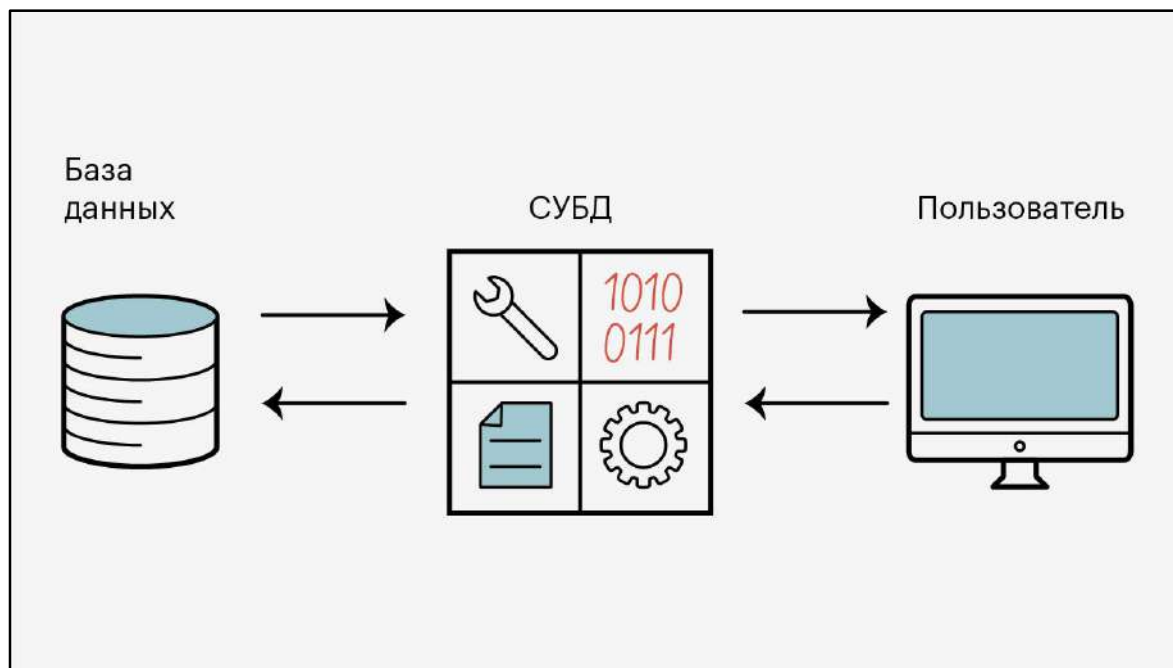


Рисунок 5 – Пример взаимодействия СУБД с базой данной
и пользователем

Рассмотрим актуальные базы данных на 2023 год. Первой рассматриваемой базой данных будет Oracle 12с. Актуальная версия на момент написания предназначена для облачных сред и может быть размещена на одном или нескольких серверах, это позволяет управлять базами данных, которые содержат миллиарды записей.

Достоинства:

1) самые актуальные инновации и функционал уже внедрены в этой базе данных, поскольку компания Oracle стремится держать планку даже на фоне других разработчиков;

2) Oracle является очень надежной среди подобных систем.

Недостатки:

- 1) стоимость Oracle может оказаться высокой, особенно для небольших организаций;
- 2) система может потребовать значительных ресурсов, поэтому возможно потребуются модернизировать оборудование для внедрения Oracle.

Второй рассматриваемой базой данных будет PostgreSQL. PostgreSQL является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных веб-сайтов. Это весьма старая система, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными.

Достоинства:

- 1) является масштабируемым решением и позволяет обрабатывать терабайты данных;
- 2) поддерживает формат json;
- 3) включает множество predefined функций;
- 4) доступен ряд интерфейсов.

Недостатки:

- 1) документация не совсем ясна, поэтому ответы на некоторые вопросы нужно будет искать самому;
- 2) конфигурация не рассчитана на новичков;
- 3) скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения.

Третьей рассматриваемой базой данных будет MySQL. MySQL – одна из самых популярных СУБД для веб-приложений. Это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность.

Достоинства:

- 1) распространяется бесплатно;
- 2) имеет понятную документацию;
- 3) предлагает множество функций, даже в бесплатной версии;
- 4) поддерживает набор пользовательских интерфейсов;
- 5) может работать с другими базами данных.

Недостатки:

- 1) отсутствует встроенная поддержка XML и OLAP;
- 2) для бесплатной версии доступна только платная поддержка.

После рассмотрения трех наиболее популярных СУБД, выявления их достоинств и недостатков, было принято решение использовать СУБД MySQL.

На диаграмме ниже представлен обзор базы данных CMS WordPress, созданной в ходе установки CMS. В таблице ниже описаны дополнительные сведения о таблицах.

На рисунке 6 представлена схема базы данных с предустановленными таблицами WordPress.

В таблице 1 содержатся названия таблиц и их описание.

Таблица 1 – Описание баз данных

Название	Описание
wp_users	Содержит в себе информацию о зарегистрированных пользователях сайта.
wp_usermeta	Используется для хранения метаинформации о пользователях.
wp_comments	Используется для хранения комментариев, оставленных пользователями.
wp_commentmeta	Используется для хранения метаинформации о комментариях.
wp_posts	Хранит в себе большинство данных – записи, страницы и весь контент.
wp_postmeta	Используется для хранения метаинформации о записях (постах и страницах).
wp_terms	Хранит в себе категории как для записей так и для ссылок, а также теги для записей.
wp_term_meta	Хранит в себе метаданные о терминах (категориях, метках или пользовательских таксономиях).
wp_term_taxonomy	Хранит в себе метаданные о терминах (категориях, метках или пользовательских таксономиях).

wp_term_meta	Хранит в себе метаданные о терминах (категориях, метках или пользовательских таксономиях).
wp_term_taxonomy	Описывает таксономию (категорию, ссылку или тег) для записей в таблице wp_terms.
wp_term_relationships	Содержит связи записей (постов) с таксономиями (категориями, метками).
wp_options	Используется для хранения настроек и параметров, связанных с установкой и функционированием WordPress.

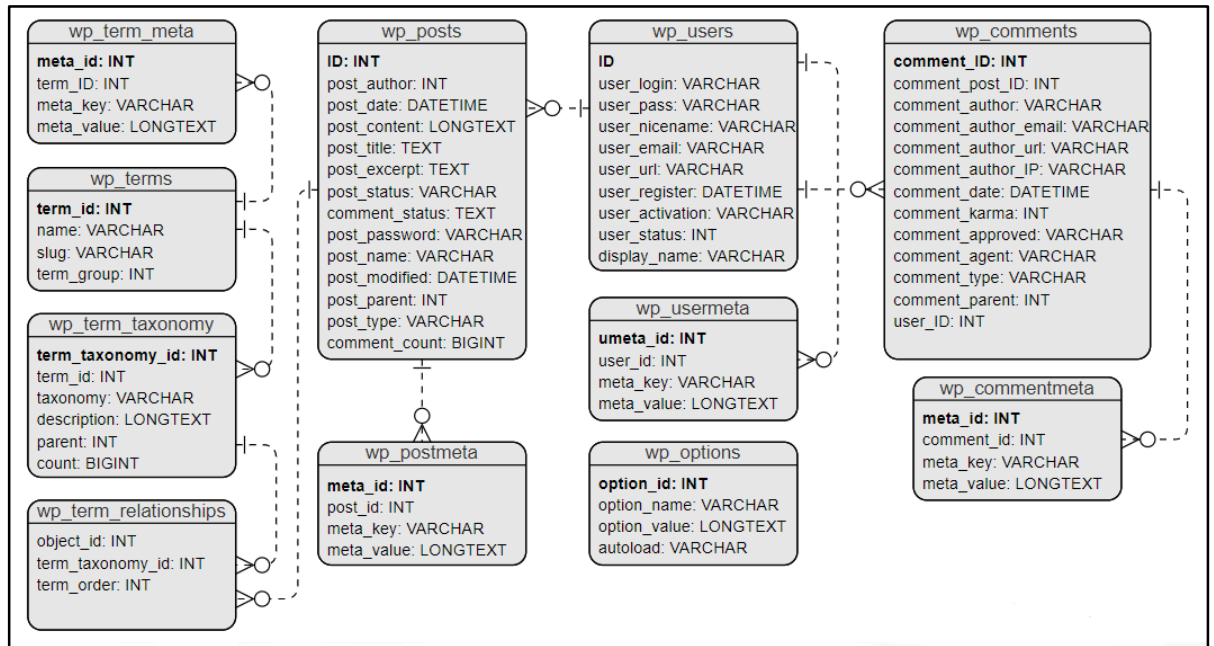


Рисунок 6 – Схема базы данных

В WordPress данные о страницах, созданных с помощью плагина Elementor, хранятся в двух основных таблицах базы данных: `wp_posts` и `wp_postmeta`. Это позволяет WordPress эффективно управлять и отображать созданный контент на веб-сайте. В таблице `wp_posts` хранятся основные сведения о страницах, включая их заголовки, содержимое, дату создания и другую информацию. В таблице `wp_postmeta` хранятся дополнительные метаданные (данные о данных) для записей в таблице `wp_posts`.

2.3. Диаграмма компонентов

Классическое веб-приложение обычно реализует трехзвенную архитектуру, которая состоит из веб-браузера, веб-сервера и сервера базы данных. Рассмотрим каждый компонент подробнее.

1. Веб-браузер – это клиентское приложение, которое пользователь использует для взаимодействия с веб-приложением. Он отображает веб-страницы, обрабатывает пользовательские действия, такие как клики и ввод данных, и отправляет HTTP-запросы на веб-сервер.

2. Веб-сервер – это программное обеспечение, которое обрабатывает HTTP-запросы, полученные от веб-браузера. В случае WordPress, веб-сервер принимает запросы, передает их в ядро WordPress и обрабатывает их с помощью соответствующих файлов и плагинов. Веб-сервер также может выполнять другие задачи, такие как обработка статических файлов, кэширование и безопасность.

3. Сервер базы данных – это серверное программное обеспечение, которое управляет хранением и доступом к данным. В случае WordPress, сервер базы данных MySQL хранит данные, связанные с контентом, настройками, пользователями и т.д. Веб-сервер отправляет запросы на чтение, запись и обновление данных в базу данных, и сервер базы данных возвращает соответствующие результаты.

Взаимодействие между веб-сервером и сервером базы данных осуществляется через язык запросов к базе данных, такой как SQL (Structured Query Language). Веб-сервер отправляет SQL-запросы на базу данных для выполнения операций с данными, таких как извлечение, обновление и удаление. База данных обрабатывает запросы и возвращает результаты веб-серверу, который затем формирует и отправляет ответ пользователю через веб-браузер.

Диаграмма компонентов, представленная на рисунке 7, показывает взаимодействие между клиентским интерфейсом (веб-браузером),

веб-сервером и сервером базы данных. Клиентский интерфейс отправляет запросы на веб-сервер, который в свою очередь обращается к базе данных для получения необходимых данных, и затем отправляет ответ обратно клиенту для отображения на веб-странице.

Такая трехзвенная архитектура позволяет разделить ответственность между различными компонентами системы, что облегчает разработку, масштабирование и поддержку веб-приложения.

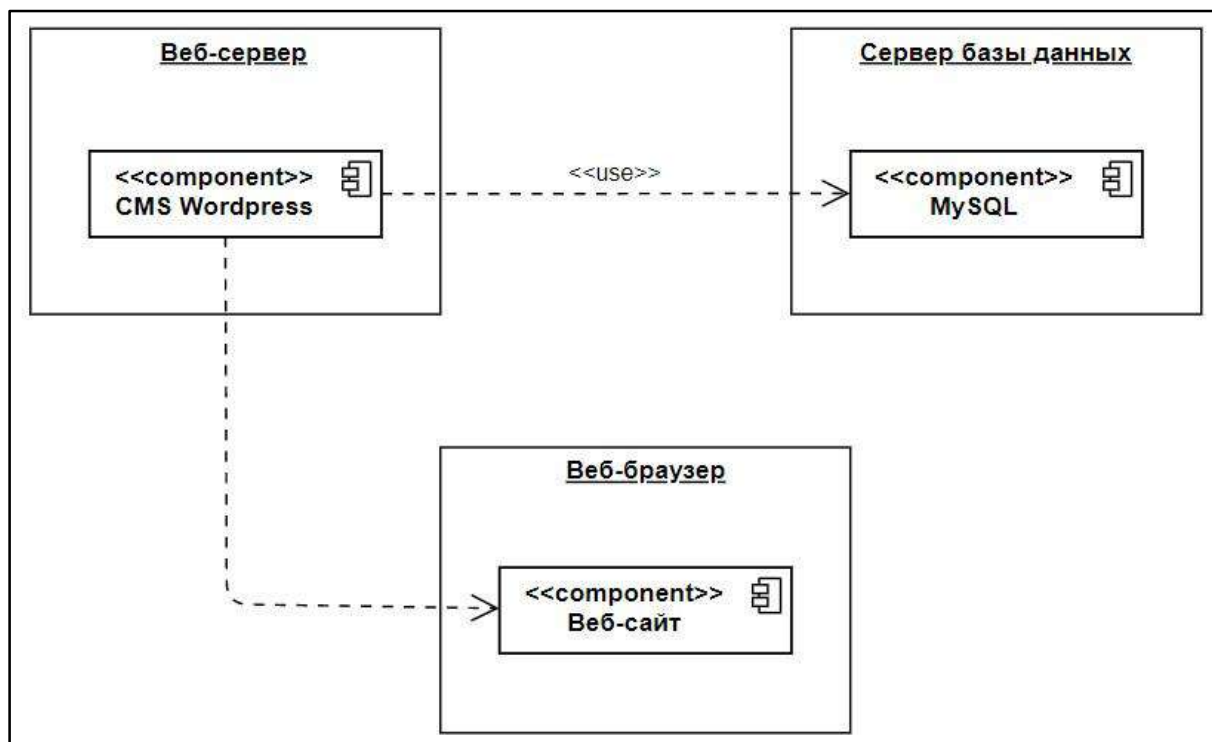


Рисунок 7 – Диаграмма компонентов веб-сайта

Вывод по второй главе

Во второй главе были рассмотрены требования к системе, изображены и описаны диаграмма вариантов использования, диаграмма компонентов, а также рассмотрена база данных веб-сайта.

3. РЕАЛИЗАЦИЯ СИСТЕМЫ

3.1. Реализация интерфейса

Для работы над веб-сайтом на базе CMS WordPress необходимо выбрать подходящую начальную тему и создать необходимые страницы. Для последующей реализации я установил недостающий плагин Elementor, который предоставляет обширный функционал и позволяет создавать страницы с разнообразным контентом, используя простой и интуитивно понятный интерфейс.

Для веб-сайта турагентства была выбрана тема Hello, которая является подходящей для создания веб-сайта данной тематики. Выбор начальной темы и использование плагина Elementor дают широкие возможности для создания уникального и профессионального веб-сайта туристической фирмы, а также обеспечивают удобство в работе над его контентом и дизайном. На рисунке 8 изображено окно настройки основных элементов сайта.

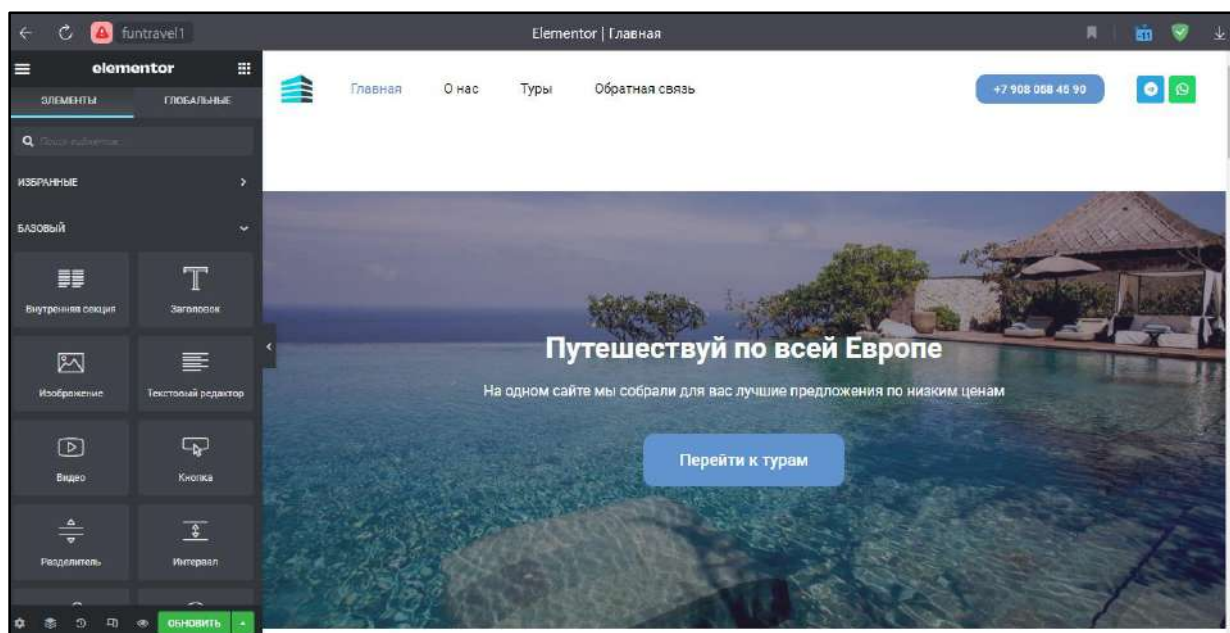


Рисунок 8 – Настройка основных элементов веб-сайта

Для заполнения веб-сайта нужной информацией, первоочередной задачей является создание страниц, которые будут содержать необходимую информацию для пользователей.

Страницы на веб-сайте могут содержать разнообразное содержимое в зависимости от целей и задач. Они могут включать таблицы, изображения, медиафайлы, текстовые блоки, а также встроенные Google карты или другие элементы. На рисунке 9 показаны созданные страницы. В листинге 1 демонстрируется код создания страниц.

<input type="checkbox"/>	Главная — Главная страница, Elementor	admin
<input type="checkbox"/>	О нас — Elementor	admin
<input type="checkbox"/>	Обратная связь — Elementor	admin
<input type="checkbox"/>	Политика конфиденциальности — Страница политики конфиденциальности	admin
<input type="checkbox"/>	Тур Барселона — Elementor	admin
<input type="checkbox"/>	Тур Берн — Elementor	admin

Рисунок 9 – Создание страниц веб-сайта

Листинг 1 – Код создания страниц

```
register_post_type(
    'page',
    array(
        'labels' => array(
            'name_admin_bar' => _x( 'Page', 'add new from admin bar' ),
        ),
        'public' => true,
        'publicly_queryable' => false,
        '_builtin' => true,
        '_edit_link' => 'post.php?post=%d',
        'capability_type' => 'page',
        'map_meta_cap' => true,
        'menu_position' => 20,
        'menu_icon' => 'dashicons-admin-page',
        'hierarchical' => true,
        'rewrite' => false,
        'query_var' => false,
        'delete_with_user' => true,
        'supports' => array( 'title', 'editor', 'author', 'thumbnail', 'page-at-tributes', 'custom-fields', 'comments', 'revisions' ),
        'show_in_rest' => true,
        'rest_base' => 'pages',
        'rest_controller_class' => 'WP_REST_Posts_Controller',
    )
);
```


Далее необходимо создать «шапку» веб-сайта, с элементами, которые характерны веб-сайту турагентства. Такими элементами являются:

- 1) переход на страницу «Туры», которая содержит информацию об активных турах;
- 2) переход на страницу «О нас», которая содержит информацию о сотрудниках и информацию о внутреннем устройстве компании;
- 3) переход на страницу «Обратная связь», на которой содержится блок с контактной формой;
- 4) переход на страницу «Главная» со второстепенных страниц, которая содержит информацию о популярных местах и блок с отзывами.

На рисунке 10 изображены элементы «шапки» веб-сайта в админ-панели WordPress. В листинге 2 демонстрируется код создания меню сайта.

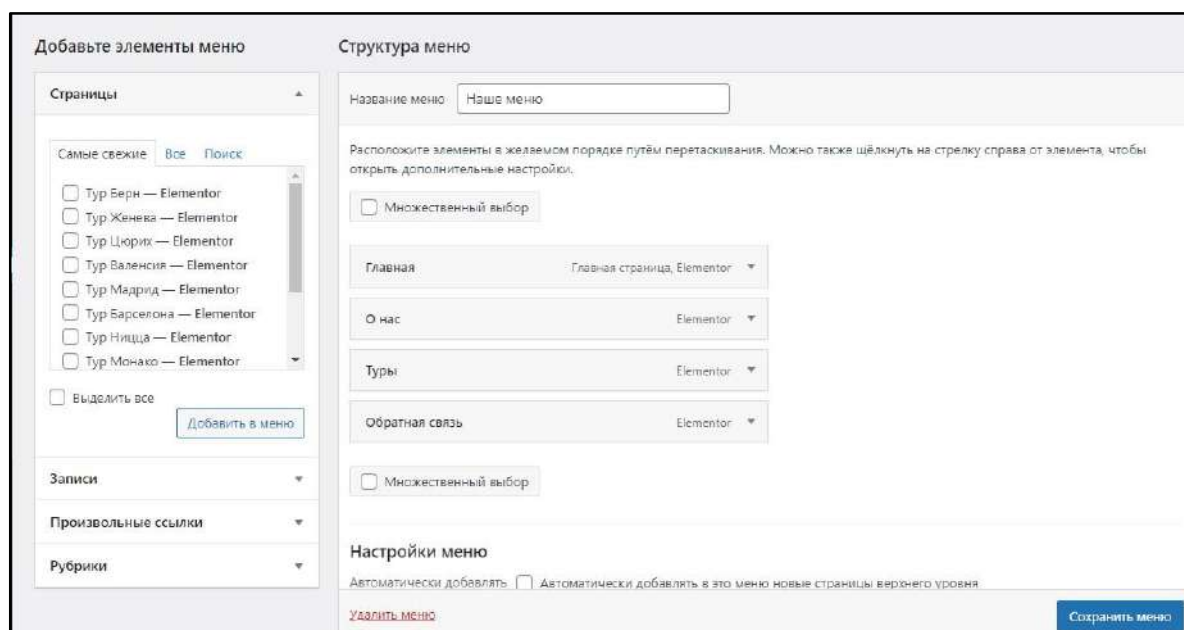


Рисунок 10 – Создание «шапки» веб-сайта

Листинг 2 – Код созданию меню веб-сайта

```
function wp_nav_menu( $args = array() ) {  
    static $menu_id_slugs = array();  
    $defaults = array(  
        'menu' => '',  
        'container' => 'div',  
        'container_class' => '',
```

```

'container_id' => '',
'container_aria_label' => '',
'menu_class' => 'menu',
'menu_id' => '',
'echo' => true,
'fallback_cb' => 'wp_page_menu',
'before' => '',
'after' => '',
'link_before' => '',
'link_after' => '',
'items_wrap' => '<ul id="%1$s" class="%2$s">%3$s</ul>',
'item_spacing' => 'preserve',
'depth' => 0,
'walker' => '',
'theme_location' => '',
);

```

Для возможности оставления обратной связи на сайте я использовал плагин MetForm на странице «Обратная связь». Плагин Metform является инструментом для создания и управления формами обратной связи на веб-сайте, разработанном на CMS WordPress. Он предоставляет удобный и интуитивно понятный интерфейс для создания различных типов форм.

Для организации возможности оставления заявок обратной связи необходимо настроить данный плагин. Для этого потребуется установить MetForm через раздел «Плагины» в административной панели Wordpress и активировать его. После необходимо настроить поля формы в соответствии с дизайном сайта, добавив необходимые поля, такие как имя, фамилия, электронная почта, дата поездки и другие. Для каждого поля необходимо настроить различные параметры, такие как обязательное заполнение и тип данных. Также необходимо настроить параметры формы, такие как заголовок, описание, кнопку отправки и другие опции. После завершения визуальных настроек, нужно настроить SMTP для того, чтобы уведомления о заявках обратной связи приходили на электронную почту. SMTP (Simple Mail Transfer Protocol) – это протокол, который позволяет отправлять электронные письма через почтовый сервер. Для этого понадобится доступ к данным SMTP-сервера, которые обычно предоставляет почтовый сервис. На рисунке 11 представлена настройка формы обратной связи.

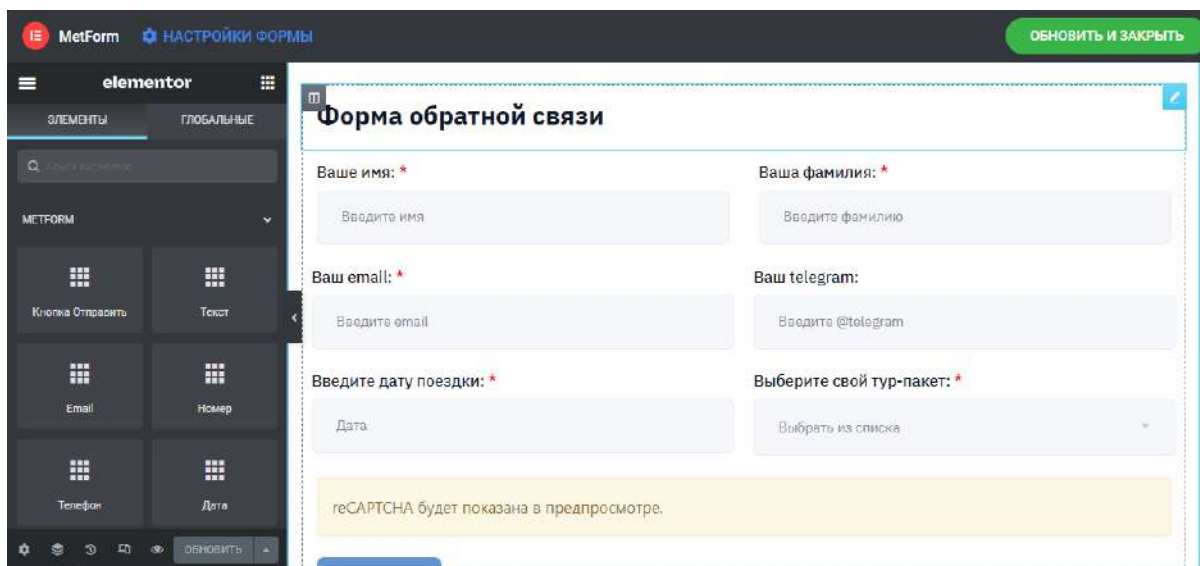


Рисунок 11 – Настройка формы обратной связи

3.2. Реализация телеграм-бота для обратной связи

Дополнительным функционалом на сайте является телеграм-бот обратной связи, в котором можно быстро связаться с оператором турагентства и уточнить окончательную стоимость тура, написав дату поездки, выбранную страну, город и указав выбранный турпакет.

Данный бот был создан с помощью LivegramBot. LivegramBot – это бот для Telegram, который предоставляет возможность организовать обратную связь с посетителями веб-сайта через мессенджер Telegram. Основная цель LivegramBot состоит в том, чтобы упростить процесс коммуникации и обмена информацией между веб-сайтом и его посетителями, предоставляя удобный способ обратной связи с посетителями веб-сайта и позволяя легко получать и отвечать на их заявки или вопросы через Telegram.

Чтобы осуществить переход в телеграм-бот, необходимо нажать на кнопку «Перейти». Данную кнопку можно найти на страницах «Главная», «О нас» и на страницах каждого тура. А также, можно перейти в бот через «шапку» веб-сайта, нажав на иконку Telegram. На рисунке 12 указана главная страница веб-сайта с возможностью перехода в телеграм-бот.

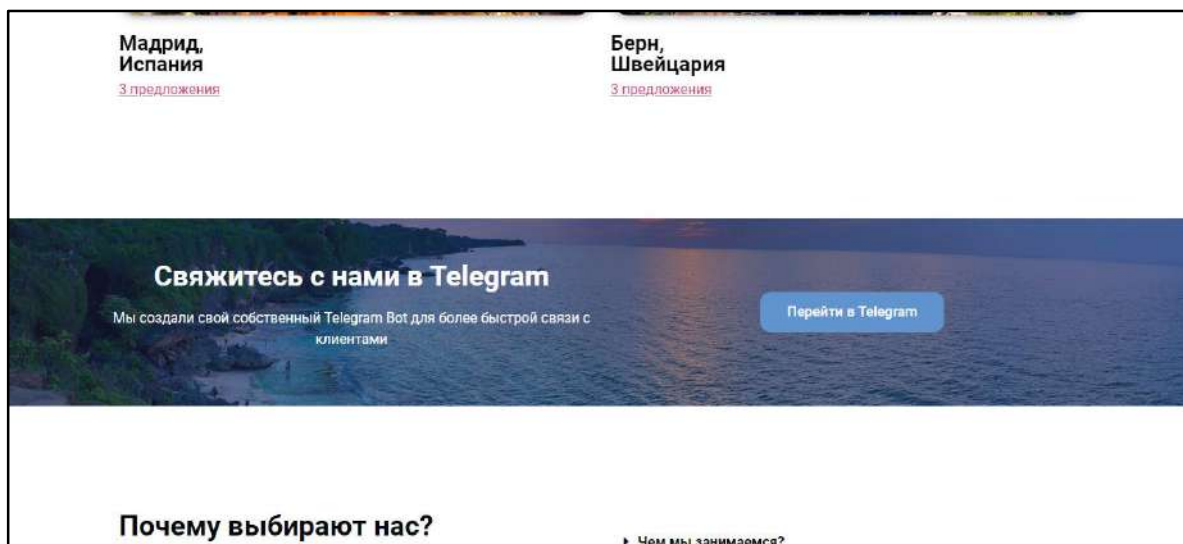


Рисунок 12 – Главная страница сайта с переходом в Telegram

Перейдя по ссылке, мы попадаем в телеграм-бот обратной связи. После оставления обратной связи оператор связывается с клиентом и озвучивает окончательную стоимость тура. На рисунке 13 показан функционал телеграм-бота.

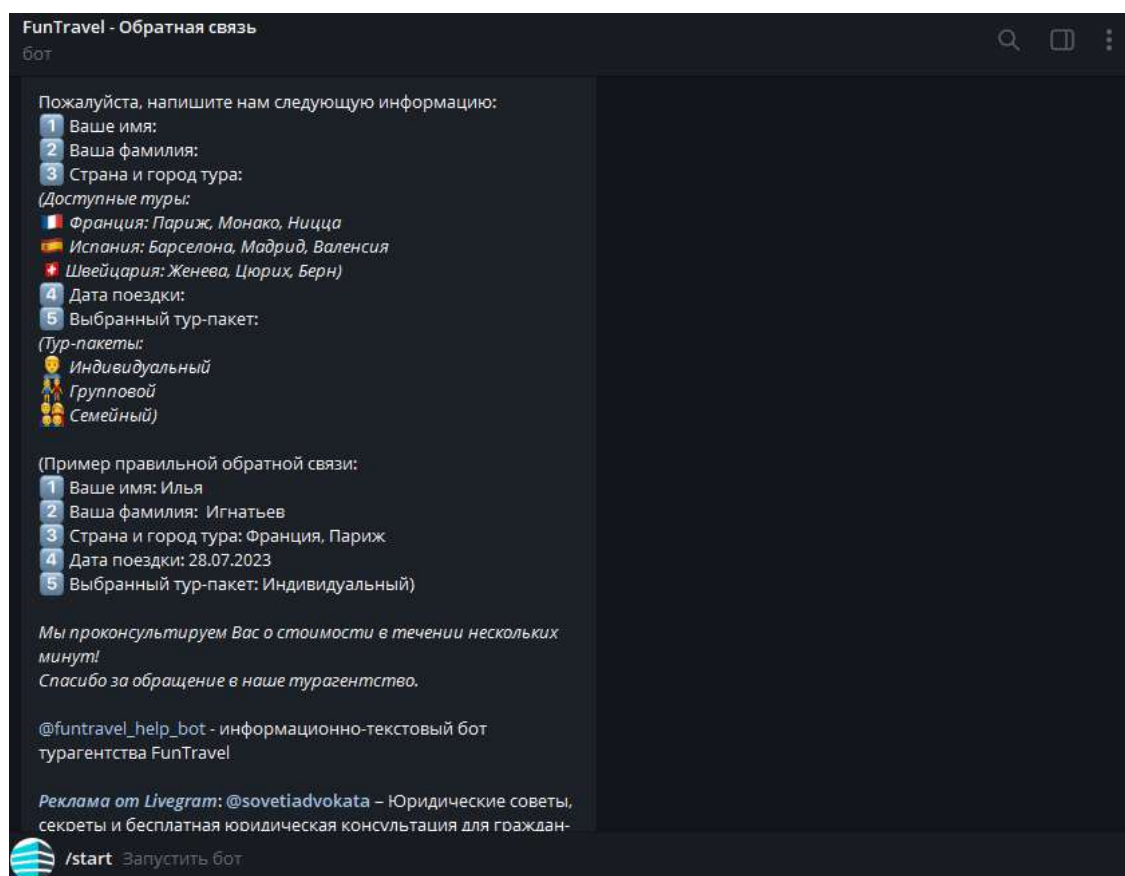


Рисунок 13 – Функционал телеграм-бота

3.3. Реализация текстового телеграм-бота

Еще одним дополнительным функционалом веб-сайта является текстовый телеграм-бот, дублирующий информацию на сайте. Он обеспечивает легкий доступ к информации веб-сайта, улучшая общий опыт пользователей и повышая их удовлетворенность.

Чтобы воспользоваться телеграм-ботом, необходимо нажать на ссылку, расположенную на главной странице веб-сайта. Перейти в текстовый бот также можно через телеграм-бота для обратной связи. На рисунке 14 показана главная страница с ссылкой для перехода в телеграм-бот.

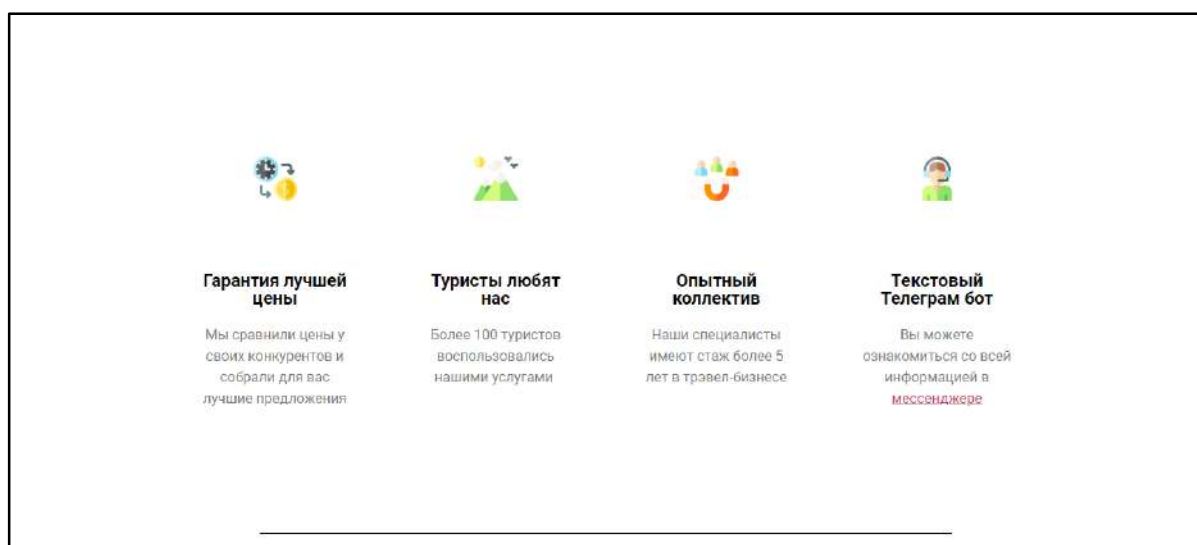


Рисунок 14 – Главная страница с ссылкой на телеграм-бот

Перейдя по ссылке, мы попадаем в бот, в меню которого включены такие команды, как: /start, /help, /company, /tours, /tourpackages. Каждая команда выполняет свое действие:

- 1) команда /start перезапускает бот;
- 2) команда /help передает информацию о всех имеющихся командах бота;
- 3) команда /company содержит информацию о турагентстве;
- 4) команда /tours содержит описание экскурсий по главным достопримечательностям в каждом городе тура;

5) команда /tourpackages содержит информацию о турпакетах.

В листинге 3 представлена реализация телеграм-бота на языке JavaScript. На рисунке 15 показан функционал телеграм-бота.

Листинг 3 – Код телеграм-бота

```
const { Telegraf, Markup } = require('telegraf');
const { message } = require('telegraf/filters');
const text = require('./const');
const bot = new Telegraf(process.env.BOT_TOKEN);
bot.start((ctx) => ctx.replyWithHTML('<b>Добро пожаловать в информационно-  
текстовый Telegram-Bot турагентства FunTravel!</b>'));
bot.help((ctx) => ctx.reply(text.commands));
bot.command('tours', async ctx => {
  try {
    await ctx.replyWithHTML("&<b>Пожалуйста, выберите страну:</b>", Markup.inlineKeyboard (
      [
        [Markup.button.callback('Франция', 'btn_1')],
        [Markup.button.callback('Испания', 'btn_2')],
        [Markup.button.callback('Швейцария', 'btn_3')],
      ]
    ));
    bot.command('company', async ctx => {
      try {
        await ctx.replyWithHTML("<b>О нашей команде:</b>")
      } catch(e) {
        console.error(e)
      }
    });
    bot.command('tourpackages', async ctx => {
      try {
        await ctx.replyWithHTML("&#9658 <b>Индивидуальный тур</b>")
      }
      function addCityActionBot(name, src, text) {
        bot.action(name, async (ctx) => {
          try {
            await ctx.answerCbQuery()
          }
          await ctx.replyWithHTML(text, {
            disable_web_page_preview: true
          })
        })
      }
      function addActionBot(name) {
        bot.action(name, async (ctx) => {
          try {
            await ctx.answerCbQuery()
            if (name === 'btn_1') {
              await ctx.replyWithHTML('<b>Пожалуйста, выберите город:</b>', Markup.inlineKeyboard (
                [
                  [Markup.button.callback('Париж', 'city_1')],
                  [Markup.button.callback('Монако', 'city_2')],
                  [Markup.button.callback('Ницца', 'city_3')]
                ]
              ));
            }
          }
        });
      }
    });
  }
});
bot.launch();
```

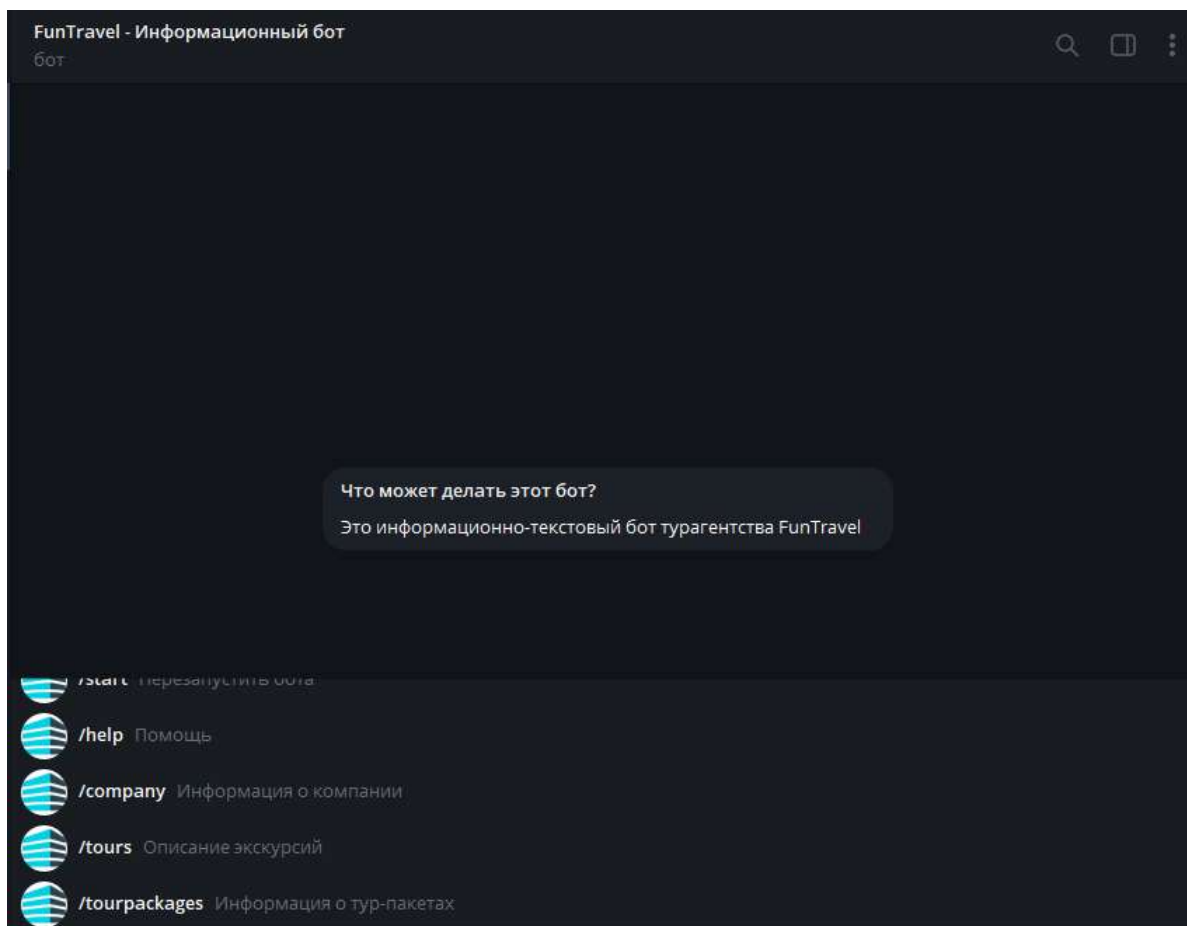


Рисунок 15 – Функционал телеграм-бота

Вывод по третьей главе

В третьей главе были установлены необходимые компоненты для веб-сайта, такие как локальный сервер Open Server и система управления контентом Wordpress. Сформированы бэкенд и фронтенд, которые успешно функционируют в веб-браузере и локальном сервере. А также были разобраны и показаны текстовой телеграм-бот для веб-сайта и телеграм-бот для обратной связи.

4. ТЕСТИРОВАНИЕ

Для тестирования веб-приложения было применено функциональное тестирование созданной системы, а также интеграционное тестирование приложения.

4.1. Функциональное тестирование

Функциональное тестирование – это вид тестирования, направленный на проверку корректности работы функциональности сайта и соответствия его внешнего вида и поведения поставленным задачам. Оно осуществляется с целью убедиться, что веб-приложение реализует функциональные требования, заданные в спецификации.

В ходе функционального тестирования проводится проверка различных функций и возможностей сайта, таких как навигация по страницам, заполнение и отправка форм, обработка данных и другие функциональные аспекты. При этом основное внимание уделяется проверке правильности работы функциональности и соответствия ее ожидаемому поведению.

При загрузке сайта появляется его главная страница. Она представлена на рисунке 16.

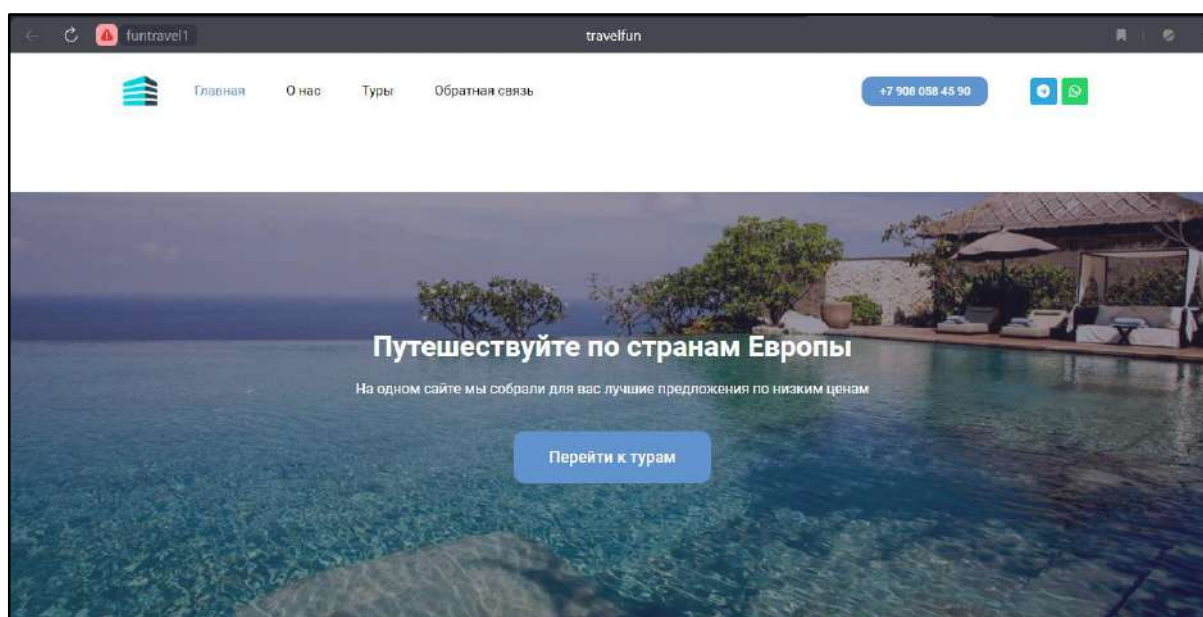


Рисунок 16 – Главная страница сайта

На главной странице сайта находится блок с популярными местами, состоящий из картинок и описания количества актуальных предложений по данному туру. На рисунке 17 представлен блок с популярными местами.

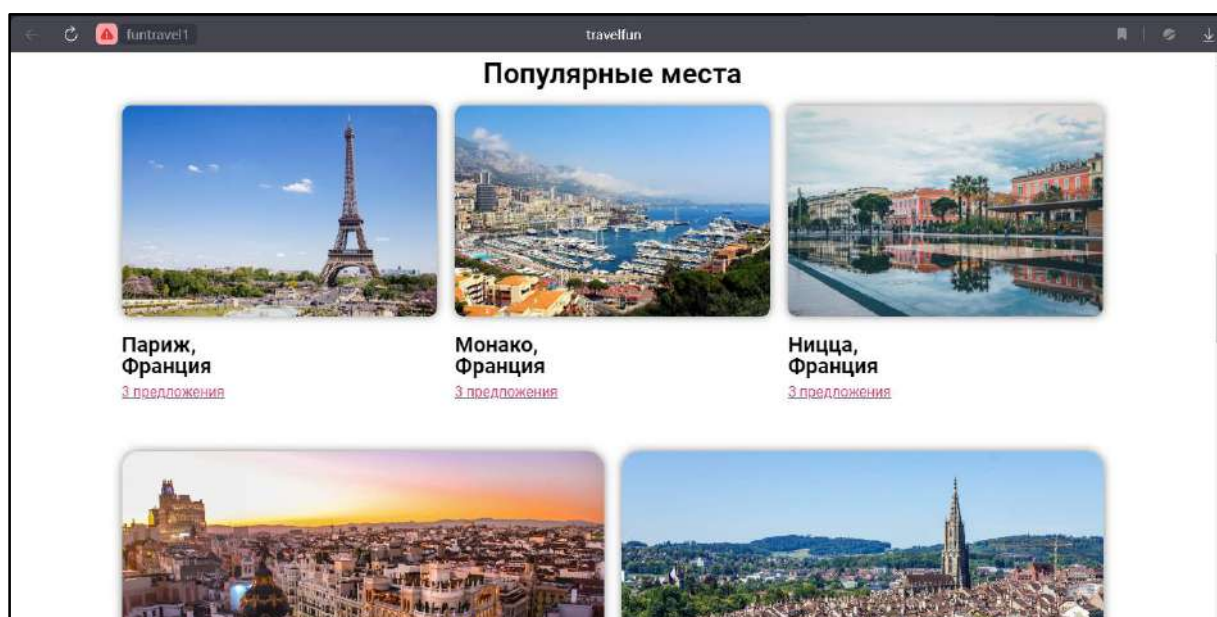


Рисунок 17 – Популярные места

При нажатии на ссылку осуществляется переход на страницу тура, где указаны главные достопримечательности города, описание экскурсий, информация о турпакетах и возможность перехода на страницу обратной связи. На рисунке 18 представлен скриншот страницы города.



Рисунок 18 – Скриншот страницы города

На рисунке 19 изображена страница с формой обратной связи, с помощью которой пользователи могут заполнить обязательные поля и получить информацию о стоимости тура на свою электронную почту, указанную ранее. После заполнения всех полей на сайте появляется уведомление об успешной отправке заявки.

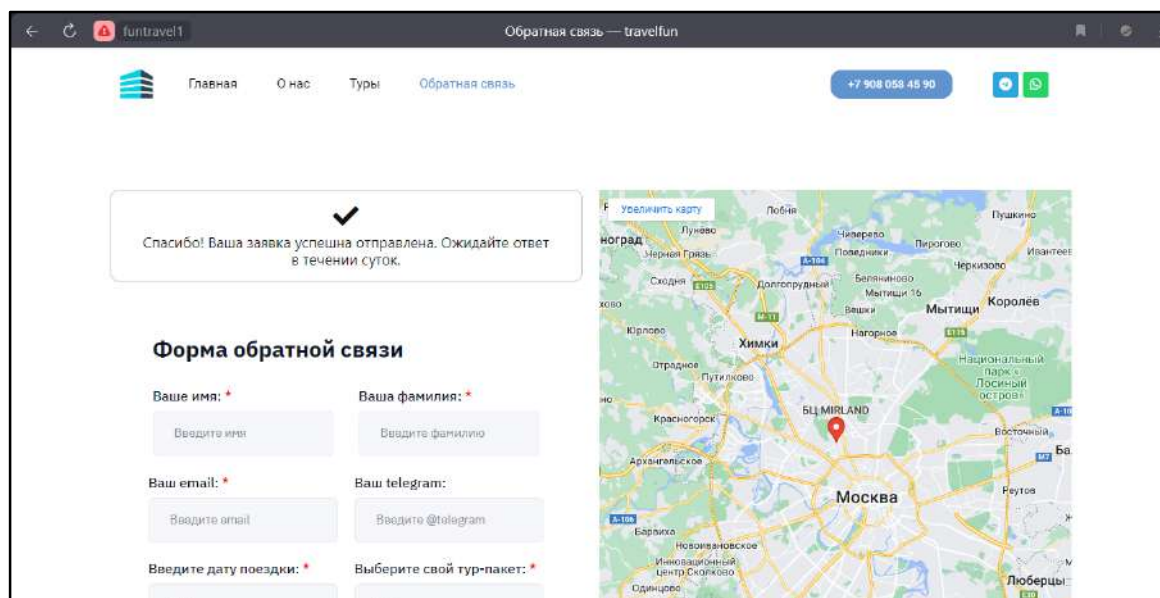


Рисунок 19 – Страница с формой обратной связи

Также пользователи могут воспользоваться возможностью перейти в телеграм-бот обратной связи и оставить свою заявку прямо в мессенджере. Для удобства перехода в бот предусмотрены несколько способов.

Во-первых, пользователи могут найти ссылку на телеграм-бот в шапке веб-сайта. Она размещается в верхней части страницы и является заметным значком.

Во-вторых, пользователи могут перейти в телеграм-бот с главной страницы веб-сайта. На главной странице размещается информация о возможности оставить заявку через Telegram, и пользователю предлагается перейти в бот, кликнув на соответствующую кнопку.

Наконец, для удобства пользователей, на странице каждого доступного тура предусмотрена возможность перехода в телеграм-бот. Скриншот телеграм-бота для обратной связи представлен на рисунке 20.

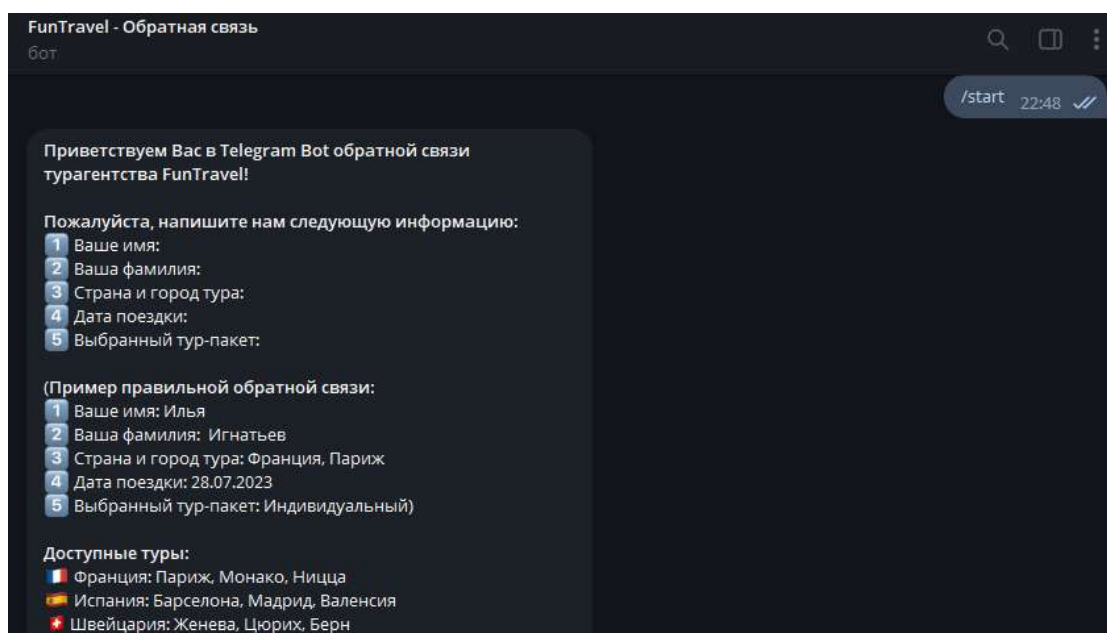


Рисунок 20 – Скриншот телеграм-бота для обратной связи

У пользователей также есть возможность узнать информацию о тур-пакетах и прочитать описание экскурсий по каждому из городов в текстовом телеграм-боте, переход в который осуществляется с главной страницы веб-сайта или из телеграм-бота для обратной связи. На рисунке 21 представлен скриншот текстового телеграм-бота.

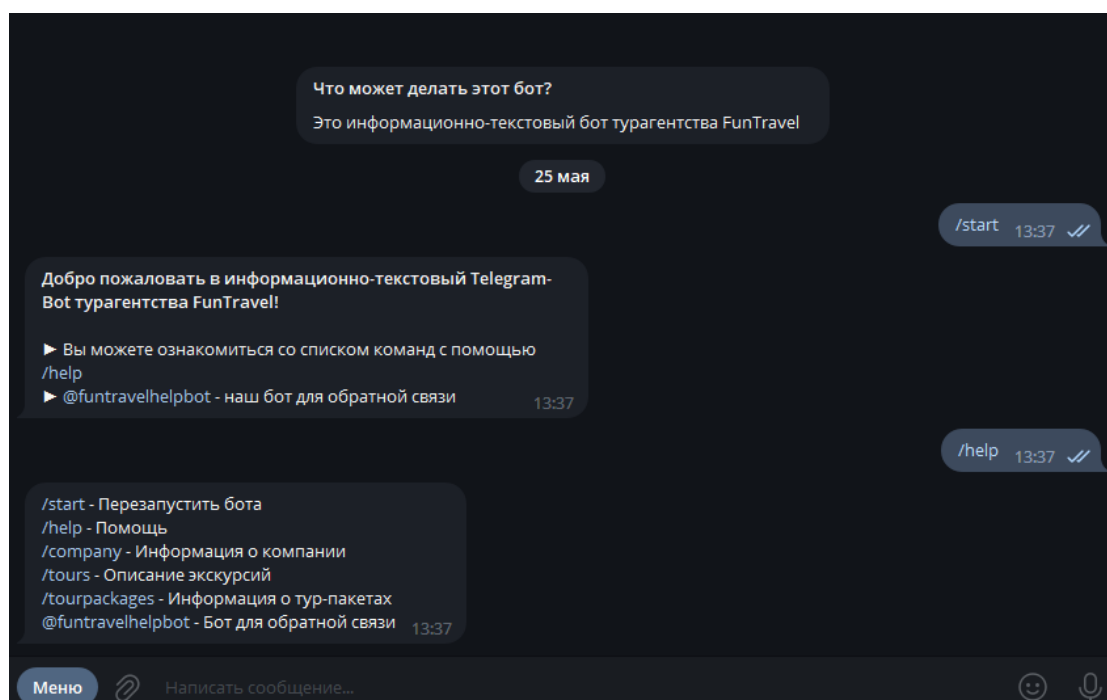


Рисунок 21 – Скриншот текстового телеграм-бота

В таблице 2 представлены результаты функционального тестирования сайта и телеграм-бота.

Таблица 2 – Функциональное тестирование сайта и телеграм-бота

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
1	Проверка корректности отображения контента на всех страницах сайта.	Перейти на любую страницу сайта.	Правильное отображение страницы.	Да
2	Проверка возможности перехода на все страницы сайта.	Перейти на ссылки в контенте сайта и в шапке сайта.	Переход на другую страницу.	Да
3	Проверка получения информации в текстовом телеграм-боте.	Выбрать в меню любую команду.	Отображение информации в боте по выбранной команде.	Да
4	Проверка корректности отправки заявки через форму обратной связи на странице «Обратная связь».	Оставить заявку в форме обратной связи.	Заявка отправлена, пользователю пришло уведомление об успешной отправке.	Да
5	Проверка корректности отправки заявки в телеграм-боте для обратной связи.	Оставить заявку обратной связи по указанному примеру.	Заявка отправлена, пользователю выслано сообщение об успешной отправке.	Да
6	Проверка сайта на адаптивность.	Рассмотреть сайт в разрешении для компьютера, планшета и телефона.	Отображение адаптивной страницы для каждого устройства.	Да

4.2. Проверка установки соединения с базой данных

Проверка установки связи с базой данных – это важный процесс проверки соединения и взаимодействия веб-приложения с базой данных MySQL. В разработке онлайн-веб-приложений, особенно на основе CMS WordPress, установка и поддержание связи с базой данных MySQL является критическим аспектом.

В процессе проверки установки связи с базой данных, осуществляется проверка правильности соединения между веб-приложением и базой данных MySQL. Это включает проверку доступности базы данных, успешное установление соединения и выполнение необходимых запросов для получения данных.

Успешное установление связи с базой данных MySQL является неотъемлемой частью функционирования веб-приложений, созданных на основе CMS WordPress. База данных MySQL хранит и обрабатывает данные, необходимые для работы приложения. Если база данных недоступна, приложение не сможет получить необходимую информацию, что может привести к нарушению функциональности приложения. На рисунке 22 представлена подобная ошибка.

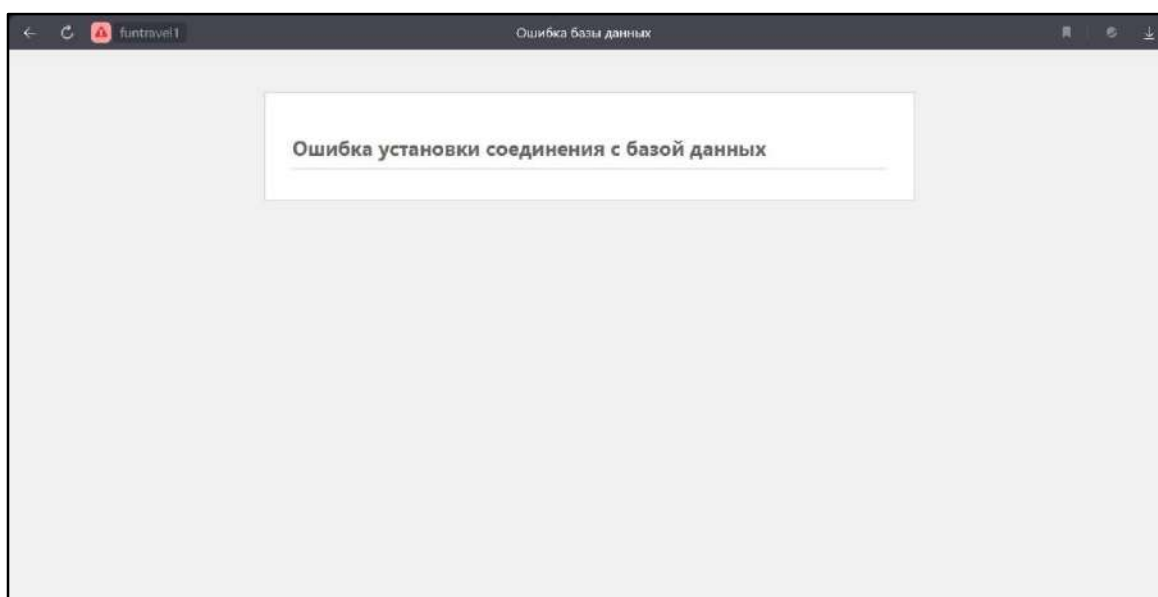


Рисунок 22 – Возможная ошибка соединения с базой данных

Вывод по четвертой главе

В четвертой главе было произведено функциональное тестирование веб-сайта и проверка соединения с базой данной. В ходе тестирования не было выявлено ошибок в функциональном тестировании веб-сайта.

ЗАКЛЮЧЕНИЕ

В современном мире почти каждое турагентство или частное лицо, занимающееся туристической деятельностью, осознают важность наличия собственного веб-сайта. Веб-сайт турагентства является неотъемлемым инструментом для привлечения клиентов и предоставления информации о предлагаемых турах и услугах.

Туристический веб-сайт является ценным ресурсом для клиентов, поскольку предоставляет полезную информацию и помогает им сориентироваться в широком спектре туров. На сайте турагентства пользователи могут найти подробные описания экскурсий туров, фотографии, а также оставить заявку обратной связи для связи с агентством. Веб-сайт обеспечивает удобство использования и позволяет пользователям получить точную и актуальную информацию о турах, что помогает им принять взвешенное решение и выбрать подходящий вариант для своих путешествий.

В данном проекте был разработан веб-сайт по оказанию туристических услуг с использованием системы управления контентом (CMS) WordPress и базы данных MySQL. Эта комбинация позволяет эффективно управлять контентом на сайте, обновлять информацию о турах, добавлять новые предложения и взаимодействовать с клиентами. База данных MySQL используется для хранения и обработки данных, связанных с турами, клиентами и другой важной информацией, необходимой для работы турагентства. При этом были решены следующие задачи:

- 1) поставлена задача по разработке веб-сайта;
- 2) произведен анализ существующих веб-сайтов турагентств;
- 3) определены требования и разработана архитектура сайта;
- 4) реализован веб-сайт;
- 5) протестирован веб-сайт;
- 6) подготовлен отчет.

ЛИТЕРАТУРА

1. Travelata. [Электронный ресурс] URL: <https://travelata.ru/> (дата обращения: 16.05.2023 г.).
2. Level Travel. [Электронный ресурс] URL: <https://level.travel/> (дата обращения: 16.05.2023 г.).
3. Online Tours. [Электронный ресурс] URL: <https://www.onlinetours.ru/> (дата обращения: 16.05.2023 г.).
4. 15 самых известных сайтов на WordPress. [Электронный ресурс] URL: <https://www.kv.by/blog/users/alionatarankova/1050858-15-samyhizvestnyh-saytov-na-wordpress> (дата обращения: 16.05.2023 г.).
5. Что такое CMS сайта. Какие бывают движки по назначению и принципу работы. [Электронный ресурс] URL: <https://www.unisender.com/ru/blog/idei/cms/> (дата обращения: 16.05.2023 г.).
6. Сравнение 5 популярных CMS. [Электронный ресурс] URL: <https://habr.com/ru/companies/vdsina/articles/514764/> (дата обращения: 16.05.2023 г.).
7. Официальный сайт WordPress. [Электронный ресурс] URL: <https://ru.wordpress.org/> (дата обращения: 16.05.2023 г.).
8. Официальный сайт Open Server. [Электронный ресурс] URL: <https://ospanel.io/> (дата обращения: 16.05.2023 г.).
9. Использование диаграммы вариантов использования UML при проектировании программного обеспечения. [Электронный ресурс] URL: <https://habr.com/ru/post/566218/> (дата обращения: 18.05.2023 г.).
10. Visual Paradigm. [Электронный ресурс] URL: <https://online.visual-paradigm.com/> (дата обращения: 18.05.2023 г.).
11. Elementor Builder. [Электронный ресурс] URL: <https://elementor.com/> (дата обращения: 19.05.2023 г.).
12. Bringing MySQL to the web. [Электронный ресурс] URL: <https://www.phpmyadmin.net/> (дата обращения: 19.05.2023 г.).

13. Тема для Wordpress. [Электронный ресурс] URL: <https://ru.wordpress.org/themes/hello-elementor/> (дата обращения: 19.05.2023 г.).

14. MetForm Elementor Contact Form Builder. [Электронный ресурс] URL: <https://wordpress.org/plugins/metform/> (дата обращения: 21.05.2023 г.).

15. LivegramBot. [Электронный ресурс] URL: <https://livegram.io/> (дата обращения: 21.05.2023 г.).