

Министерство науки и высшего образования РФ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ О.В. Непомнящий

подпись                      инициалы, фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 2022 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника  
код и наименование специальности

Веб-сайт медицинского центра  
тема

Руководитель	_____	доцент, <u>канд. техн. наук</u>	<u>М. С. Медведев</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		<u>Н. В. Машковская</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	доцент, <u>канд. техн. наук</u>	<u>М. С. Медведев</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2022

Министерство науки и высшего образования РФ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ О.В. Непомнящий

подпись                      инициалы, фамилия

«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**в форме бакалаврской работы**

Студенту Машковской Наталье Вадимовне

фамилия, имя, отчество

Группа КИ18-06Б Направление (специальность) 09.03.01

номер

код

Информатика и вычислительная техника

наименование

Тема выпускной квалификационной работы Веб-сайт медицинского центра

Утверждена приказом по университету № \_\_\_\_\_ от \_\_\_\_\_

Руководитель ВКР М.С. Медведев, канд. тех. наук, доцент кафедры ВТ

инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: сформулировать цели и задачи, провести анализ существующих систем в предметной области, реализовать веб-сайт для медицинского центра травматологии «Боли нет».

Перечень разделов ВКР: анализ предметной области и обзор аналогичных систем, проектирование собственного веб-сайта, программная реализация.

Перечень графического материала: презентация в формате PowerPoint.

Руководитель ВКР

\_\_\_\_\_

подпись

М. С. Медведев

инициалы и фамилия

Задание принял к исполнению

Н. В. Машковская

подпись, инициалы и фамилия студента

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Веб-сайт медицинского центра» содержит 51 страницу, исследовано 13 источников, использовано 28 рисунков, 1 приложение.

ВЕБ-САЙТ, СЕРВЕР, КЛИЕНТ, LARAVEL, FRAMEWORK, PHP, MVC, КОНТРОЛЛЕР, МОДЕЛЬ.

Цель работы: создание веб-сайта для медицинского центра травматологии «Боли нет».

При выполнении работы был произведен обзор предметной области, задания на выпускную квалификационную работу, изучены существующие аналоги и сформированы требования, предъявляемые к серверной и клиентской части сайта.

Объект работы – клиентская и серверная часть веб-сайта медицинского центра.

Задачи:

- выполнить анализ существующих решений в предметной области;
- осуществить выбор программных средств разработки серверной и клиентской части сайта;
- выполнить программную реализацию клиентской части;
- выполнить программную реализацию серверной части сайта;
- протестировать разработанный веб-сайт.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Анализ технического задания .....	5
1.1 Анализ существующих систем .....	6
1.2 Выбор средств разработки .....	11
1.2.1 HTML + CSS .....	11
1.2.2 Java Script .....	11
1.2.3 PHP .....	12
1.2.4 Bootstrap .....	13
1.2.5 Laravel .....	13
1.2.6 Выбор системы управления базой данных .....	15
1.2.7 AJAX .....	15
1.2.8 Docker .....	16
1.3 Выводы по разделу 1 .....	17
2 Проектирование .....	18
2.1 Клиентская часть веб-сайта .....	18
2.2 Алгоритм взаимодействия сервера с клиентом .....	19
2.3 Динамические и статические сайты .....	20
2.4 Архитектура веб-сайта .....	21
2.4.1 Модель MVC .....	21
2.4.2 Модули сайта .....	23
2.4.2.1 Представления .....	23
2.4.2.2 Модели .....	24
2.4.2.3 Контроллеры .....	25
2.4.2.4 Маршрутизация .....	26
2.5 Проектирование базы данных .....	26
2.6 Панель управления сайтом .....	27
2.7 Интеграция с CRM-системой Битрикс24 .....	28
2.8 Оптимизация работы сайта .....	29

2.9 Выводы по разделу 2.....	30
3 Программная реализация.....	31
3.1 Главная страница сайта .....	31
3.2 Версия сайта для слабовидящих.....	35
3.3 Раздел «Врачи».....	36
3.4 Раздел «Отзывы» .....	37
3.5 Раздел «Контакты».....	38
3.6 Услуги.....	39
3.7 Панель управления.....	40
3.8 Структура базы данных .....	43
3.9 Интеграция форм на сайте с Битрикс24 .....	44
3.10 Выводы по разделу 3.....	46
СПИСОК СОКРАЩЕНИЙ.....	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	49
ПРИЛОЖЕНИЕ А .....	50

## **ВВЕДЕНИЕ**

В современном мире стало популярным представлять и продавать свои услуги посредством интернета – через веб-сайты. Зачастую современные сайты носят не только информативный характер, но и содержат различную интерактивность – например, онлайн-заявки, онлайн-запись на приём ко врачу, облегчающие работу операторов. Создание таких сайтов является частой практикой у коммерческих компаний. Это необходимо для организации обратной связи с потенциальными клиентами с помощью сбора информации через формы, а также для повышения уровня продаж и эффективности деятельности компании.

Плюсы таких сайтов есть и для посетителей. При записи на прием онлайн больше нет необходимости стоять в очередях, снижается нагрузка на персонал. Особенно актуально это в нынешних условиях коронавирусной инфекции.

В ходе выполнения работы были рассмотрены сайты медицинских клиник: многопрофильный медицинский холдинг СМ-КЛИНИКА, медицинский центр Лорри и клиника iNEO. После изучения веб-сайтов были выделены их недостатки и преимущества, которые были учтены при выполнении проекта.

Таким образом, сайт должен иметь версию сайта для слабовидящих, онлайн-запись на прием и форму для заказа обратного звонка. Наличие перечисленных компонентов является огромным плюсом для сайта, так как повышается количество продаж (записей), пользователь получает оперативно всю нужную информацию, а также бесплатную связь с менеджером, что повышает лояльность к компании.

## **1 Анализ технического задания**

Целью выпускной квалификационной работы является разработка веб-сайта для медицинского центра травматологии «Боли нет».

Разрабатываемый проект должен отвечать следующим требованиям:

- возможность просмотра информации о компании и предоставляемых услугах;
- наличие версии сайта для слабовидящих, регламентированной ГОСТом Р 52872-2019 [1];
- адаптивная мобильная версия;
- наличие интерактивной карты;
- возможность оставить отзыв о враче;
- онлайн-запись на прием;
- возможность оставить заявку для заказа обратного звонка;
- наличие панели управления сайтом;
- интеграция веб-сайта с CRM-системой Битрикс24.

Модель системы – клиент-серверное взаимодействие.

Технические требования к сайту: серверная часть сайта разрабатывается на языке программирования PHP версии 8.1, с использованием системы управления базами данных MySQL и фреймворком Laravel для реализации общей логики сервера. Клиентская часть разрабатывается с помощью фреймворка bootstrap для создания адаптивной верстки сайта.

Графическое оформление страниц сайта представлено в формате Photoshop Document (PSD). PSD – формат хранения графической информации, созданный специально для программы Adobe Photoshop.



## 1.1 Анализ существующих систем

В интернете представлено огромное количество похожих сайтов клиник, с примерно похожим функционалом, а именно – с возможностью записи на прием и просмотром услуг. Ниже приведены некоторые примеры:

- СМ-КЛИНИКА многопрофильный медицинский холдинг.
- Медицинский центр Лорри-мед.
- Многопрофильная медицинская клиника iNEO.

Рассматриваемые сайты позволяют получить информацию о предоставляемых услугах, врачах и акциях. Сайты предоставляют обширное количество информации о различных медицинских исследованиях, о видах анализов и порядке их сдачи, о самом медицинском центре и ценах на услуги. Также есть возможность прочесть отзывы людей и оставить свой комментарий. На сайтах имеется возможность записи-онлайн и записи по телефону.

«СМ-Клиника» – официальный сайт одного из крупнейших сетей многопрофильных медицинских центров для взрослых и детей [2]. Сервис включает карту сайта, которая поможет сориентироваться на сайте (рис. 1.2).

Основные разделы сайта представлены на рисунке 1.1 и 1.2.

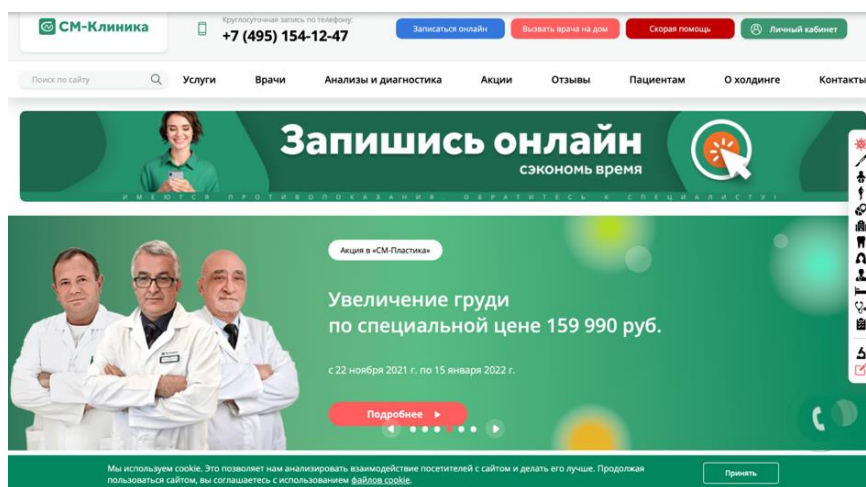


Рисунок 1.1 – Главная страница сайта СМ-Клиника

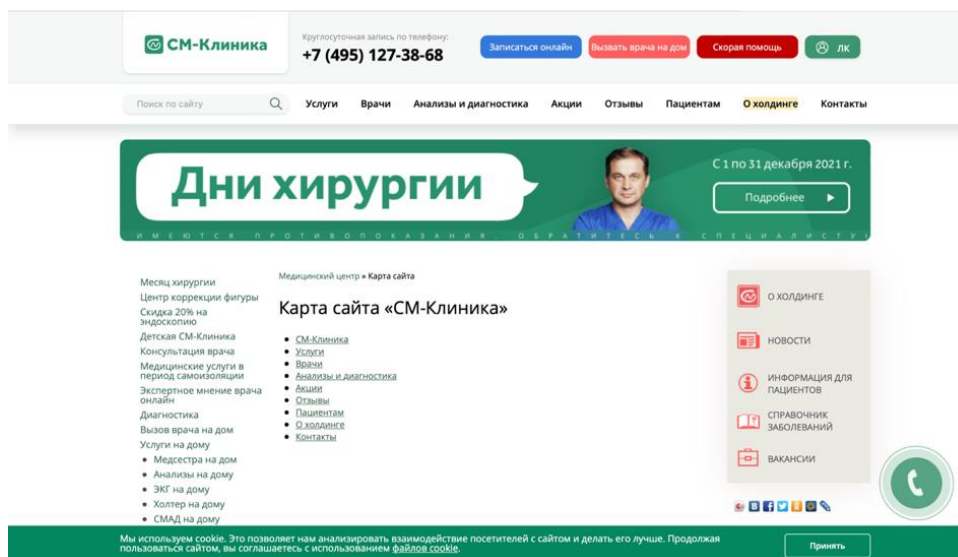


Рисунок 1.2 – Карта сайта СМ-Клиника

К достоинствам сервиса можно отнести возможность поиска по сайту. Сайт содержит форму для заказа обратного звонка. С ее помощью можно проконсультировать клиента. Если на сайте отсутствует данная форма, есть вероятность потерять клиентов.

Недостатками сервиса является достаточно большое количество информации, в которой легко запутаться, и существенный минус – это отсутствие версии для слабовидящих.

Частная клиника Лорри представляет собой целый комплекс разнообразных услуг в области здоровья [3]. Лорри – это многопрофильный центр с широким спектром диагностики, консультаций узких специалистов и услуг восстановительной медицины.

Веб-сайт имеет простой и понятный интерфейс. Пользователи сайта найдут там подробную информацию о видах диагностики заболеваний, о мерах безопасности в клинике. Также на сайте представлены фотографии лицензии медицинской деятельности.

Основные разделы сайта представлены на рисунке 1.3 и 1.4.

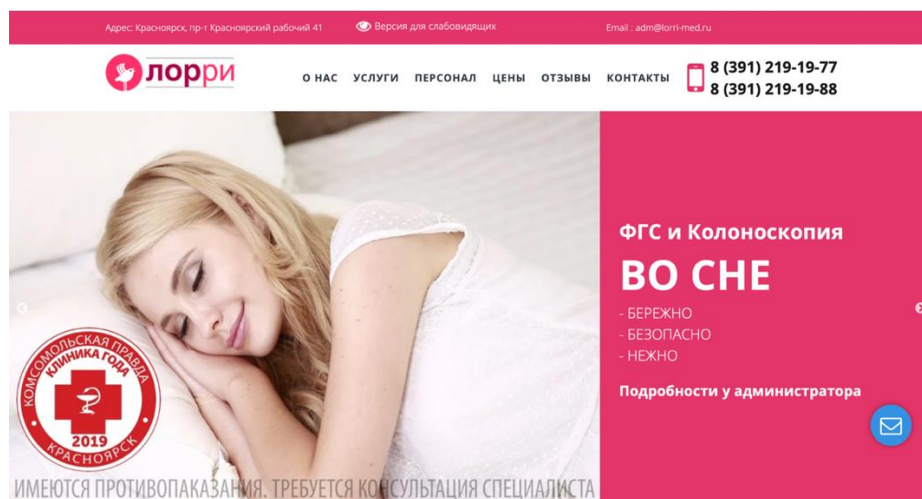


Рисунок 1.3 – Главная страница сайта СМ-Клиника



Рисунок 1.4 – Страница записи на прием

Основными достоинствами сайта является удобство интерфейса, наличие версии сайта для слабовидящих, и режим работы мед. центра, представленный на главной странице. На сайте можно оставить отзыв о клинике. Также имеется страница, содержащая ссылки на лицензию медицинской деятельности, что является обязательным.

Недостаток сервиса – отсутствие формы для заказа обратного звонка.

Многопрофильная медицинская клиника iNEO сочетает в себе научно-практический подход к выбору методов диагностики и лечения, оборудования

и медицинских кадров, что в свою очередь гарантирует, что пациент получит высококачественную медицинскую помощь [4].

Главный экран клиники представлен на рисунке 1.5.

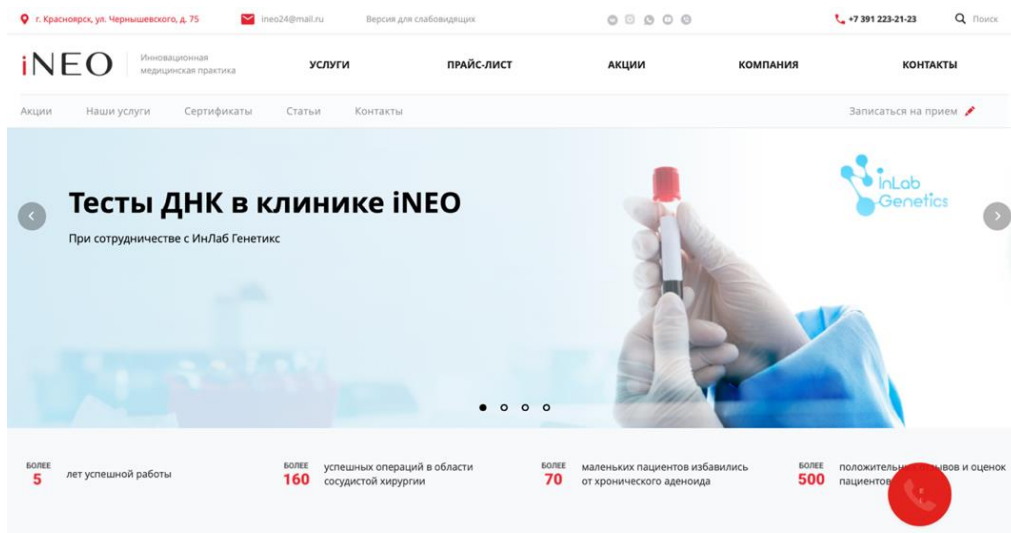


Рисунок 1.5 – Главный экран iNEO

Форма записи на прием показана на рисунке 1.6.

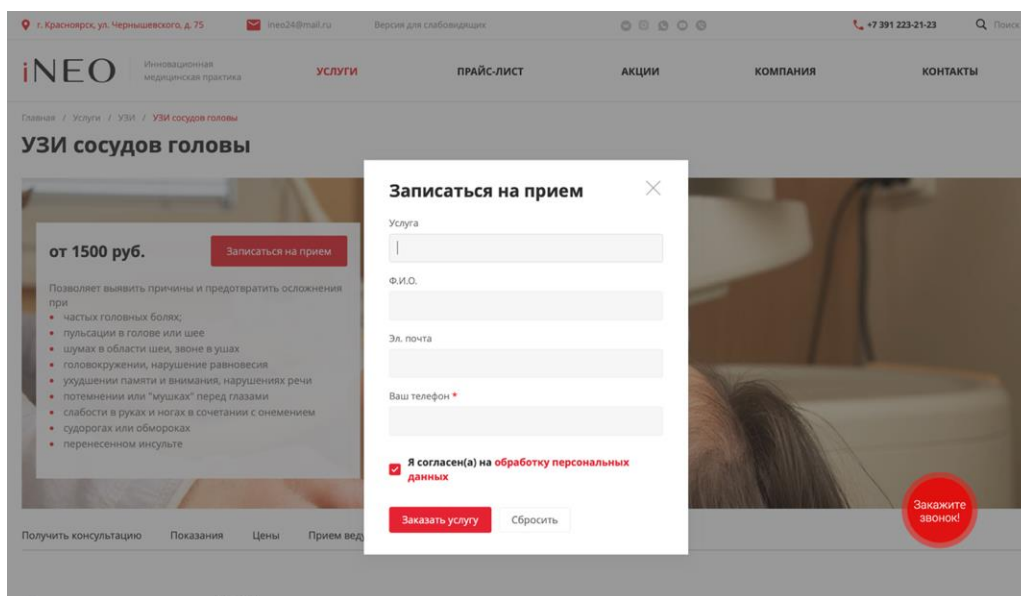


Рисунок 1.6 – Форма записи на прием

Сайт медицинского центра имеет версию сайта для слабовидящих, онлайн-запись на прием, а также содержит информацию о клинике, услугах и о врачах.

Далее продемонстрирована сравнительная таблица основных параметров существующих сайтов (Таблица 1.1).

Таблица 1.1 – Основные характеристики

Функции	СМ- клиника	Лорри-мед	iNEO
Понятный интерфейс	-	+	+
Возможность онлайн-записи	+	+	+
Версия для слабовидящих	-	+	+
Адаптивность сайта под мобильные устройства	+	+	+
Заказ обратного звонка	+	-	+

Исходя из анализа рассмотренных вариантов, можно сделать вывод о том, что полезными функциями на медицинском сайте является онлайн-запись на прием и запись по телефону.

На сайте обязательно должна присутствовать версия сайта для слабовидящих, в соответствии с требованиями доступности для людей с инвалидностью и других лиц с ограничениями жизнедеятельности.

Также значительную роль для клиники играют отзывы – поэтому необходимо реализовать возможность оставить отзыв и ознакомиться с ними.

Раздел «Специалисты» является одним из самых важных на сайте медицинского центра. На странице важно кратко описать врача, показать фотографию, имя и фамилию, а также квалификацию.

## **1.2 Выбор средств разработки**

Рассмотрим средства разработки веб-сайта, которые были использованы при реализации данного проекта.

### **1.2.1 HTML + CSS**

Для построения каркаса сайта мы будем использовать язык разметки HTML и таблицу стилей CSS. HTML и CSS – две неразделимые составляющих любого сайта.

HTML – это язык гипертекстовой разметки документа или страницы. Разметка используется для структурирования и отображения веб-страницы с контентом. HTML-документ содержит в себе теги и текст. Теги преобразуются в визуальные объекты в браузере пользователя. Например, тег `<img>` предназначен для размещения и отображения изображений на странице.

CSS – это каскадные таблицы стилей, которые отвечают за оформление внешнего вида сайта. CSS определяет стили для оформления документов, в том числе дизайн, верстку и вид макета для различных устройств. Большинство современных веб-сайтов работают на основе связки HTML и CSS.

### **1.2.2 Java Script**

Для того, чтобы оживить сайт и сделать его интерактивным, используется язык программирования JavaScript. К примеру, после заполнения определенной формы на сайте и нажатия кнопки «отправить», реакция на это

действие обычно обеспечивается скриптом, написанным на JavaScript. Важная особенность JavaScript – развитая инфраструктура. Разработчикам доступны мощные инструменты, различные библиотеки и фреймворки.

### **1.2.3 PHP**

Программные части сайта разрабатываются на языке программирования PHP, с использованием базы данных MySQL. При написании программного кода использовался PHP версии 8.1. PHP 8 является более быстрым и надежным чем его предшественники.

PHP нужен для динамического формирования страниц на основе информации из базы данных. База данных – информационная модель, позволяющая хранить данные об объекте или группе объектов, которые обладают набором свойств и которые можно категоризировать. Данные хранятся упорядоченно.

В области веб-программирования PHP является одним из самых популярных сценарных языков. Основная задача PHP – это «оживление» HTML страниц [5]. PHP был разработан для создания динамических веб-сайтов, его код просто встраивается в HTML, что позволяет легко конвертировать статические страницы в динамические.

Это делает данный язык самым подходящим для создания полноценных сайтов. Он принимает входящий запрос от веб-сервера, выполняет сценарий и возвращает веб-серверу готовую HTML-страницу. Полученный результат сервер отправляет в браузер, в котором уже и отображается готовая страница пользователю. После этого видно свежий курс валют, погоду, и многое другое, где информация меняется и нужно отображать ее актуальное состояние.

### 1.2.4 Bootstrap

Фреймворк [6] – заготовка для быстрой разработки, на основе которой можно дописать собственный код. При работе с фреймворком уже есть структура приложения, правила и необходимый набор инструментов для создания проекта.

В качестве фронтенд-фреймворка, который будет отвечать за внешний вид проекта, будем использовать bootstrap.

Bootstrap [7] – это открытый и бесплатный HTML, CSS и JS фреймворк, необходимый разработчикам для быстрой вёрстки адаптивных дизайнов сайтов.

Фреймворк Bootstrap представляет собой набор готовых блоков, классов и функций, и состоит из CSS и JavaScript файлов. Для начала работы необходимо подключить данные файлы к странице. После этого станут доступны инструменты фреймворка: колоночная система (сетка Bootstrap), формы, классы и компоненты.

Например, в нашем проекте, с помощью компонента modal будут реализованы модальные окна.

### 1.2.5 Laravel

В качестве серверного фреймворка будем использовать Laravel. Laravel [8] – это фреймворк для web-приложений, который позволит упростить решение основных задач, таких как маршрутизация, кэширование, архитектура приложения, работа с базой данных.

Laravel имеет собственный обработчик шаблонов «Blade» и синтаксис, облегчающий выполнение рутинных операций, например – управление маршрутизацией.

Назначение Laravel – разработка приложений с использованием архитектуры MVC. При работе с таким приложением, пользователь



взаимодействует с представлением и контроллером. Представление – это то, что видит пользователь, визуальный интерфейс. Контроллеру пользователь отдает команды. Архитектура MVC позволяет писать код, который удобен для чтения и редактирования, а также делает процесс разработки комфортным, разделяя работу frontend и backend разработчиков.

Исходя из всего вышесказанного, можно сделать вывод о том, что наиболее подходящим для нашего проекта является фреймворк Laravel.

Ниже рассмотрим возможности фреймворка Laravel.

Artisan – интерфейс командной строки, включенный в Laravel. Он предлагает ряд полезных команд – генерация файлов модели, контроллеров и уведомлений из командной строки. Это в разы удобнее, так как нет необходимости копировать откуда-либо шаблон класса или писать его вручную. Например, с помощью команды `php artisan make:model` можно создать файл модели.

ORM – это технология для связи базы данных и языка программирования, благодаря которой, для работы с БД программист может использовать язык программирования, удобный для него, вместо написания запросов на SQL. Ускорение разработки достигается за счет использования ORM. Laravel пользуется собственной реализацией ORM – библиотекой Eloquent и работает по схеме, где которой каждой таблице в базе соответствует один класс. Таким образом, можно напрямую взаимодействовать с объектом на том же языке, на котором программируем.

Шаблонизатор Blade. Шаблонизатор преобразует HTML-шаблоны в готовые страницы. Шаблоны представляют собой заготовки для будущих веб-страниц, которые включают HTML-верстку без контента и PHP-код. Шаблоны Blade содержат язык разметки HTML, а также директивы Blade, которые позволяют легко выводить значения, создавать операторы «если», выполнять итерацию данных.

Валидация – проверка входящих данных, для определения и обработки ошибки. Логику валидации можно написать самостоятельно, Laravel содержит удобные правила валидации, которые задокументированы.

### **1.2.6 Выбор системы управления базой данных**

Также нам необходимо выбрать базу данных. Поскольку в качестве серверного фреймворка был выбран Laravel, то будем использовать MySQL, так как она автоматически включается в поставку PHP.

MySQL [9] – одна из наиболее используемых систем управления базами данных, которая работает с реляционными БД. Реляционными называют такие базы данных, таблицы которой связаны между собой.

Принцип работы MySQL – клиент-сервер. Клиент, то есть пользователь, отправляет со своего компьютера запрос, который обрабатывает сервер баз данных и предоставляет ответ. MySQL-сервер – это сервер, на котором хранится база данных.

Данная СУБД отличается хорошей скоростью работы, надежностью и гибкостью.

### **1.2.7 AJAX**

Для отправки данных на сервер с форм будем использовать AJAX.

AJAX [10] – технология обращения к серверу без перезагрузки страницы. Любой веб-сайт, использующий AJAX, может отправлять и извлекать данные с сервера без необходимости перезагрузки всей страницы.

Технология AJAX полезна для форм и кнопок, связанных с элементарными действиями: добавить в корзину, подписаться. В проекте отправка данных с форм происходит посредством AJAX.

Преимущества использования данной технологии в том, что действия происходят быстрее, а также это экономичнее, так как не все пользователи обладают безлимитным интернетом. Пользователь должен понимать, что происходит на странице в конкретный момент, поэтому разработчикам необходимо реализовать индикаторы загрузки или добавить текстовых сообщений, гласящих о том, что идёт обмен данными с сервером.

### **1.2.8 Docker**

Docker [11] – это платформа для разработки и запуска контейнерных приложений. С помощью контейнеров можно упаковать приложение со всеми его зависимостями в единый образ, который в последующем будет запускаться в изолированной среде.

Рассмотрим преимущества использования контейнеров docker.

Docker решает проблемы зависимостей и рабочего окружения. Например, разработчики создают сайт в системе разработки, там все настроено и сайт работает. Когда все готово, его нужно перенести в продуктивную среду. Если в этой среде будет не хватать какой-нибудь зависимости, то сайт не будет работать. Контейнеры позволяют избежать такой проблемы, так как содержат в себе все необходимое для запуска сайта. Разработчики создают приложение и упаковывают все зависимости и настройки в единый образ, который потом можно запускать на других системах.

Изоляция и безопасность. Приложения, запущенные в контейнере, не влияют на основную операционную систему, и при этом потребляют намного меньше ресурсов.

Возможность разворачивать и настраивать несколько контейнеров одновременно благодаря Docker-compose. Например, при разработке веб-приложения есть необходимость развернуть стек LAMP: Linux, Apache, MySQL, PHP. Каждый компонент – это отдельный контейнер, однако нам

необходимы все контейнеры вместе, а не отдельно взятый компонент. Docker-compose дает возможность развернуть и настроить все приложения одной командой.

В проекте docker используется для того, чтобы обеспечить переносимость веб-сайта при развертывании на сервер.

### **1.3 Выводы по разделу 1**

В первой главе выполнен обзор существующих аналогов разрабатываемого проекта. Были проанализированы существующие сайты медицинских учреждений. В ходе анализа сервисов были определены основные преимущества и недостатки рассмотренных программных продуктов, и по ним составлена обобщенная сравнительная характеристика представленных проектов, выделены общие возможности. По результатам сравнения были сформулированы требования к разрабатываемому проекту.

Также были рассмотрены и описаны средства разработки веб-сайта: HTML, CSS, PHP, фреймворк laravel для разработки с использованием архитектурной модели MVC и фреймворк bootstrap для адаптивной верстки сайта.

## **2 Проектирование**

Выполнение поставленных задач подразумевает разработку клиентской и серверной части веб-сайта медицинской клиники.

### **2.1 Клиентская часть веб-сайта**

Клиентская часть веб приложения – это графический интерфейс, то, что отображается на странице в браузере. Верстка была выполнена на основе макетов графического оформления веб-сайта.

Шаблоны графического оформления сайта содержат 14 макетов формата psd для основных разделов сайта.

Ниже представлена структура сайта.

- Акции.
- Клиника.
  - Специалисты.
  - Отзывы.
  - Вопрос-ответ.
  - Лицензии и сертификаты.
  - О клинике.
- Услуги и цены.
  - Лечение боли в суставах верхних конечностей.
  - Лечение боли в области головы и шеи.
  - Лечение боли в спине.
  - Лечение боли в суставах нижних конечностей.
- Врачи.
- Контакты.
- Запись на прием.

Графический интерфейс и структура веб-страницы описывается языком разметки HTML. Так как мы работаем с Laravel, то все страницы были написаны с помощью встроенного шаблонизатора blade. Оформление веб-страниц описывается таблицами стилей – CSS.

В проекте была реализована адаптивная верстка с использованием фреймворка bootstrap. Адаптивным дизайном называется способ вёрстки, когда ширина макета веб-страницы подстраивается под ширину устройства. Лучшим решением будет менять макет, чтобы он соответствовал устройству. Для этого в bootstrap введены ключевые слова, которые встречаются в именах классов:

- xs – для устройств  $\geq 576\text{px}$ ;
- sm – для устройств  $\geq 768\text{px}$ ;
- lg – для устройств  $\geq 992\text{px}$ ;
- xl – для устройств  $\geq 1200\text{px}$ .

Для добавления интерактивности графическому интерфейсу будем использовать скрипты JavaScript. Благодаря скрипту, написанном на JavaScript можно реализовать пользовательский интерфейс с богатыми возможностями – выпадающие списки, анимацию, слайдеры, реакцию на нажатие кнопок.

Подробнее реализация клиентской части рассмотрена в разделе 2.4.2.1.

## **2.2 Алгоритм взаимодействия сервера с клиентом**

В веб-сайтах используется концепция взаимодействия «клиент-сервер». Клиент и сервер физически представляют собой программы. Клиент – это то, с чем взаимодействует пользователь. Клиентом может являться браузер. Клиентский код отвечает за ту часть, что видит пользователь:

- структура веб-страницы;
- настройка стилей и внешнего вида страницы;

- обработка событий пользовательского взаимодействия (нажатие кнопок).

Взаимодействие браузера (клиента) с сервером происходит при помощи HTTP – протокола передачи гипертекста. Браузер отправляет на сервер HTTP-запрос после заполнения формы, при переходе на другую страницу сайта или при поиске.

После обработки запроса, сервер отвечает веб-браузеру сообщением с HTTP-ответом. В ответе содержится код статуса HTTP-ответа, который показывает, выполнен ли запрос успешно. После того как HTML-страница возвращена, она отображается браузером.

Схематично клиент-серверное взаимодействие изображено на рисунке 2.1.

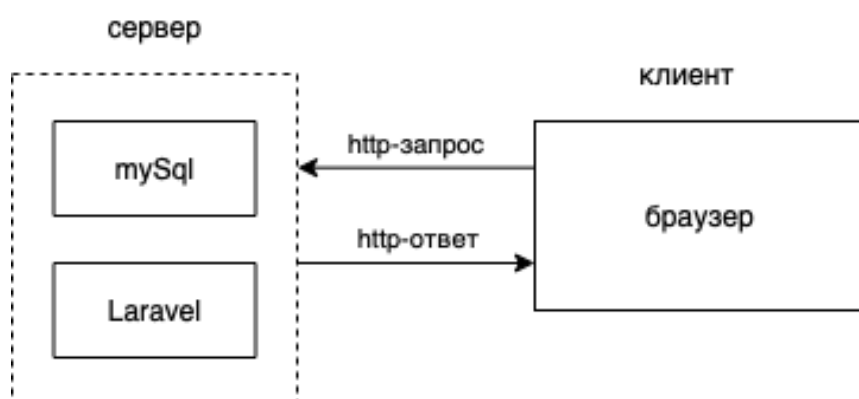


Рисунок 2.1 – Клиент-серверное взаимодействие

### 2.3 Динамические и статические сайты

В веб-разработке существуют статические сайты и динамические.

Статические сайты состоят из неизменяемых страниц, внешний вид и наполнение которых одинаково для всех посетителей. При переходе на статический сайт в браузере сервер сразу предоставляет готовый HTML-документ в том виде, в котором он был создан.

Динамические сайты содержат страницы, которые собираются заново по каждому новому запросу, а не размещены на сервере в готовом виде. В базе данных хранится структурированная информация, из которой собираются страницы, а не отдельные HTML-файлы.

Динамический сайт выигрывает у статического, когда речь идет о большом количестве страниц. Можно эффективно хранить информацию в базе данных, наполнение которой легко расширить и отредактировать. Использование HTML-шаблонов позволяет очень легко изменить структуру сайта, так как делать это нужно только в одном шаблоне, а не через десятки статических страниц.

Наш веб-сайт является динамическим, а в базе данных хранится вся динамическая информация сайта, такая как прайс, услуги, информация о врачах, отзывы и пункты меню. При переходе на страницу с врачами клиники, сервер получает GET-запрос, обращается к базе данных за информацией и затем создаёт HTML-страницу для ответа, вставляя полученные данные в HTML-шаблон.

## **2.4 Архитектура веб-сайта**

### **2.4.1 Модель MVC**

Фреймворк Laravel написан с использованием шаблона проектирования MVC, согласно которому можно разделить приложение на три отдельные составляющие: модель, представление и контроллер.

Рассмотрим подробнее этот шаблон проектирования.

Модель – это данные и правила, применяющиеся для работы с данными. Модель отвечает за методы работы с ними и структуру программы, а также реагирует на команды из контроллера и выдает информацию. Данные затем передаются в представление.



Для управления GET или POST запросами от пользователя, используется контроллер. Он определяет, каким образом будет реагировать приложение на действия пользователя.

Представление – это наши HTML-шаблоны, которые отображаются в браузере после успешной обработки запроса. Если запрос некорректный, то попадаем на страницу ошибки 404. Задача компонента – отобразить данные, полученные от модели, на уровне пользовательского интерфейса и определить внешний вид веб-сайта.

Подводя итог, можно сказать, что контроллер выступает обработчиком входящих запросов, модель получает из базы данных информацию, необходимую для выполнения конкретных запросов, а представление определяет результат запроса, который получает пользователь.

Схематично архитектура MVC представлена на рисунке 2.2.

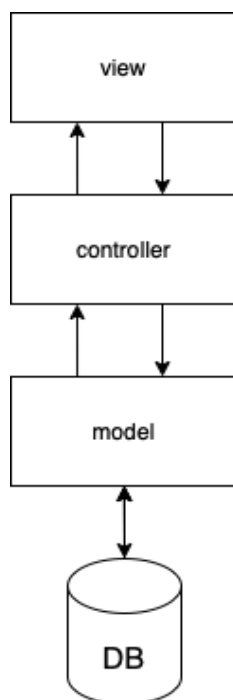


Рисунок 2.2 – MVC-архитектура

## 2.4.2 Модули сайта

Проект разделен на несколько важных частей – представления, модели, контроллеры, и маршруты.

### 2.4.2.1 Представления

Представления в нашем проекте – это HTML-шаблоны. Они размещены в папке `resources/view`. В Laravel есть встроенный шаблонизатор `blade`, который упрощает и оптимизирует работу.

Можно создать шаблон, поместив файл с расширением `.blade.php` в каталог `resources/views` приложения. Данное расширение сообщает фреймворку, что файл содержит шаблон `blade`. Шаблоны `blade` содержат HTML, а также директивы `blade`, которые позволяют легко выводить значения, создавать операторы «если», выполнять итерацию данных.

Шаблоны могут быть возвращены из маршрутов или контроллера с помощью глобального помощника `view`, как показано на рисунке 2.3.

```
Route::get('/', function () {  
    return view('greeting', ['name' => 'Finn']);  
});
```

Рисунок 2.3 – Использование `view`

В нашем проекте мы используем именованное отслеживание url-адресов (рис. 2.4).

```
Route::get('/doctors', 'DoctorController@index')->name('doctor.index');
```

Рисунок 2.4 – Именованное отслеживание

С помощью функции `name` мы назначаем имя url-обработчика. Это удобно тем, что, если в последующем мы захотим поменять url-адрес страницы, нам необходимо будет просто изменить его в файле с маршрутами, а не во всех шаблонах, где он используется, что значительно упрощает разработку и экономит время.

При переходе на страницу со специалистами клиники у нас вызывается функция `index` из контроллера `DoctorController`.

На рисунке 2.5 представлена функция отображения врачей, из которой возвращается шаблон `doctor.index` с помощью `view`.

```
public function index()
{
    $doctors = Doctor::where('active', true)
        ->orderBy('sort', 'asc')
        ->get();

    foreach ($doctors as $doctor) {
        if ($doctor->image)
            $doctor->image = getResizeImage($doctor->image, ['w' => 270, 'h' => null, 'method' => 'resize']);
    }

    return view('doctor.index', ['doctors' => $doctors]);
}
```

Рисунок 2.5 – Функция отображения врачей

#### 2.4.2.2 Модели

Модель содержит всю бизнес-логику сервиса или, другими словами, то, как наш сервер взаимодействует с базой данных. Любая таблица БД имеет соответствующую «Модель», которая применяется для взаимодействия с этой таблицей (добавление, удаление записей). В моделях мы прописываем все поля нашей таблицы.

Для генерации новой модели воспользуемся Artisan-командой: `php artisan make:model`. Эта команда расположит новый класс модели в каталог `app/Models`.

Также нам необходимы файлы миграций, которые описывают определенную таблицу БД. Чтобы создать миграции, можно дописать к вышесказанной команде флаг -m. Они появятся в папке database/migrations.

Миграции – это что-то вроде контроля версий базы данных. Они позволяют команде определять схемы базы данных приложения и совместно использовать их определение. Класс миграции содержит два метода: up и down. Метод up используется для добавления новых таблиц или столбцов в базу данных, тогда как метод down должен отменять операции, выполняемые методом up.

### 2.4.2.3 Контроллеры

В контроллерах находится основная логика и методы всего проекта. По умолчанию контроллеры хранятся в каталоге app/Http/Controllers.

Контроллеры могут объединить связанную логику обработки запросов в один класс. Рассмотрим класс FaqController. Он получает информацию с базы данных от модели, передает в шаблон и возвращает собранную html-страницу с часто задаваемыми вопросами и ответами на них. Код данного класса представлен на рисунке 2.6.

```
public function __invoke()
{
    $faqs = Faq::where('active', 1)
        ->orderBy('sort', 'asc')
        ->get();

    return view('faq.index', ['faqs' => $faqs]);
}
```

Рисунок 2.6 – Функция класса FaqController

Если действие контроллера является особенно сложным, для него можно создать новый класс контроллера. Для этого необходимо определить один

метод `__invoke` в контроллере. При регистрации маршрутов для контроллеров одиночного действия указывать метод контроллера не нужно. Вместо этого можно просто передать маршрутизатору имя контроллера.

#### **2.4.2.4 Маршрутизация**

В данном фреймворке существуют маршруты. Все маршруты Laravel назначены в файлах маршрутов, которые размещены в каталоге `routes`. Эти файлы автоматически загружаются фреймворком. В файле `routes/web.php` определены маршруты для веб-интерфейса.

В Laravel маршрутизация предназначена для направления запроса к соответствующему контроллеру.

### **2.5 Проектирование базы данных**

По итогу анализа технического задания и требований к функционалу были определены следующие сущности базы данных:

- сертификаты;
- врачи;
- вопрос-ответ;
- акции;
- отзывы;
- услуги;
- типы меню;
- пункты меню;
- прайс;
- категории прайса;
- роли;
- пользователи.

В третьей главе приведена ег-диаграмма основных сущностей и описаны их связи.

## **2.6 Панель управления сайтом**

Панель управления сайтом – это инструмент для управления веб-ресурсом, редактирования внешнего вида сайта и его наполнения, добавления новых страниц и удаления старых. Эта часть доступна только администратору. Посетители сайта видят свою часть сайта, администратор видит еще и панель управления.

Возможности, которые дает админ-панель:

- возможность непрерывного развития сайта, поскольку изменение текстовой или графической информации на сайте осуществляется с помощью интуитивно-понятного редактора, по своим возможностям и графическому оформлению, напоминающего программу Microsoft Word;
- редактирование информационного наполнения администратор сайта может вести самостоятельно без обращения к веб-мастерам;
- многопользовательское администрирование предоставляет возможность управления сайтом одновременно нескольким пользователям;
- расширение функциональности путем добавления новых модулей;
- возможность управления поступающими отзывами, их публикация, редактирование или удаление.

Основные элементы панели управления сайтом:

- Окно для ввода логина и пароля, чтобы обеспечить доступ к админ-панели.
- Главная страница со всеми компонентами, которые дают возможность править текстовый контент, описание услуг, цены, изображения, файлы и другое содержание сайта.

- Инструменты редактирования. Они позволяют вносить на сайт новые элементы или изменять уже существующие.

Администрирование сайта происходит удаленно – из любого удобного места с помощью стандартного web-браузера.

Установки дополнительного программного обеспечения для работы не требуется. После внесения изменений, публикация на сайте происходит автоматически, таким образом можно сразу увидеть полученный результат.

## **2.7 Интеграция с CRM-системой Битрикс24**

CRM-система Битрикс24 – это комплекс инструментов, который подходит для нужд малого бизнеса, для эффективной и слаженной работы сотрудников.

Интеграция форм на сайте с Битрикс24 позволяет передавать заявки пользователей не только на почту администратору, но и в Битрикс24, в раздел «Сделки». Сделка это один из основных элементов CRM, процесс продажи товара или услуги клиенту. В CRM-системах весь цикл продажи представлен в виде воронки продаж, где каждая воронка разбита на определенное количество стадий.

После интеграции сайт выглядит и работает также, как и раньше, но теперь данные с формы обратной связи и записи на приём напрямую попадают в CRM. Менеджеры могут работать с клиентами в одном окне Битрикс24 и не искать заявки на почте.

После успешной отправки формы, приложение создаёт на портале сделку с данными, отправленными пользователем. Далее с этой сделкой можно работать.

Приведем пример воронки продаж для медицинской клиники. Она может состоять из следующих стадий:

- Получение заявки – потенциальный клиент заполняет форму обратной связи на сайте, далее его контактные данные передаются в CRM.

- Выявление потребности – менеджер определяет потребности клиента, цель обращения.

- Коммерческое предложение – после сбора данных менеджер формирует коммерческое предложение, предлагает и рассказывает об услугах клиники.

- Оформление записи на прием.

В процессе обслуживания клиента и оказания услуг, в CRM также можно добавлять необходимую информацию и комментарии.

Далее приводится описание списка полей формы записи на прием и их тип.

- Имя – строка. Текстовое поле, обязательное к заполнению.

- Телефон – строка, поле обязательное к заполнению.

- Что беспокоит – список, поле обязательное к заполнению.

- Желаемая дата записи – дата/время, поле обязательное к заполнению.

- Врач – список, поле необязательное к заполнению.

## **2.8 Оптимизация работы сайта**

Для оптимизации работы веб-сайта используется кэширование.

Для решения некоторых задач по извлечению или обработке данных, выполняемых веб-сайтом, может потребоваться много ресурсов центрального процессора или большое количество времени. Для ускорения процесса данные кешируют на некоторое время, чтобы их можно было быстро извлечь при последующих запросах тех же данных. Laravel по умолчанию использует драйвер кэширования файлов, который хранит кэшированные объекты в файловой системе сервера [12].

Компиляция шаблонов во время запроса отрицательно влияет на производительность, поэтому Laravel содержит команду для предварительной компиляции всех шаблонов, используемых приложением. Для повышения



производительности можно выполнить эту команду как часть процесса развертывания: `php artisan view:cache`.

## **2.9 Выводы по разделу 2**

Во втором разделе пояснительной записки был описан клиент-серверный подход и алгоритм взаимодействия сервера с клиентом. Мы рассмотрели детально архитектуру веб-сайта, в частности такой подход к проектированию сайта как MVC. Подробно описали модули сайта – представление, модели, контроллеры и маршруты, чтобы лучше понять логику работы веб-сайта. Была рассмотрена структура панели управления сайтом и ее возможности. Описали интеграцию форм на сайте с CRM-системой Битрикс24.

### 3 Программная реализация

В процессе программной реализации были решены следующие задачи:

- выполнена адаптивная верстка страниц в соответствии с макетами;
- реализована серверная часть сайта с использованием фреймворка Laravel и СУБД MySQL;
- разработана панель управления сайтом;
- проведено информационное наполнение сайта;
- интеграция форм на сайте с Битрикс24;
- развертывание сайта на веб-сервере.

Далее мы рассмотрим подробнее основные разделы сайта.

#### 3.1 Главная страница сайта

При переходе на сайт отображается главная страница медицинской клиники «Боли нет». Главная страница сайта представлена на рисунке 3.1 и 3.2.



Рисунок 3.1 – Главная страница



Рисунок 3.2 – Главная страница - блок с преимуществами

В шапке сайта на каждой странице по требованию медицинских организаций присутствует ссылка на лицензию.

У посетителя сайта есть возможность перейти на любую заинтересовавшую его страницу из меню. Переход по пунктам и подпунктам меню осуществляется благодаря файлу web.php и маршрутам, прописанных в нём. На главной странице, как и на всех последующих, есть кнопка записи на прием, и кнопка «перезвонить» для обратной связи.

Формы для отправки данных представлены на рисунке 3.3.

КЛИНИКА  
БОЛИ НЕТ

**Обратная связь**

Имя \*

Телефон \*

☒ Нажимая кнопку «Перезвоните мне», я даю свое согласие на обработку моих персональных данных, в соответствии с Федеральным законом от 27.07.2006 года №152-ФЗ «О персональных данных», на условиях и для целей, определенных в Согласии на обработку персональных данных \*

Перезвоните мне

защита от спама reCAPTCHA  
Конфиденциальность - Условия использования

Заряжено Битрикс 24

Имя \*

Телефон \*

Что беспокоит? \*

Желаемая дата записи? \*

Врач \*

☒ Нажимая кнопку «Отправить», я даю свое согласие на обработку моих персональных данных, в соответствии с Федеральным законом от 27.07.2006 года №152-ФЗ «О персональных данных», на условиях и для целей, определенных в Согласии на обработку персональных данных \*

Отправить

защита от спама reCAPTCHA  
Конфиденциальность - Условия использования

Рисунок 3.3 – Формы записи на прием и обратной связи

В формах есть защита от спама – reCAPTCHA v3. Данная reCAPTCHA является «невидимой» – от пользователя не требуется выполнять никакие действия, система сама анализирует действия посетителей сайта, такие как движения мыши или взаимодействия со страницами. Производится оценка действий пользователя по шкале от нуля до единицы. Чем ближе результат к единице, тем выше вероятность того, что посетитель – человек. Таким образом, менеджер будет получать только те заявки, которые заполняет человек, а не компьютерная программа.

Данные с формы отправляются на почту в виде уведомления. При нажатии на кнопку записаться вызывается контроллер FormController с функцией callback. В данной функции реализована отправка уведомлений на почту.

В Laravel каждое уведомление представлено единым классом (которое хранится в директории `app/Notifications`). Каждый класс уведомления содержит метод построения сообщений (например, `toMail` или `toDatabase`), который конвертирует уведомление в сообщение, оптимизированное для конкретного канала.

Уведомления отправляются с помощью `Notification`. Это полезно, когда нужно отправить уведомление нескольким уведомляемым объектам, таким как коллекция пользователей. Чтобы отправлять уведомления, передаём все уведомляемые объекты и экземпляр уведомления методу `send` (рис. 3.4).

```
public function callback(CallbackRequest $request)
{
    $users = User::whereIn('id', [2,3])->get();
    Notification::send($users, new Callback($request->all()));
}
```

Рисунок 3.4 – Функция отправки данных с формы

Для валидации данных, введенных в формы, мы будем создавать запрос формы. `Form Request` – это класс, где мы описываем правила валидации для входящих данных. Новый класс запроса формы будет помещен в каталог `app/Http/Requests` приложения.

Наш запрос формы имеет метод `rules`, который содержит правила валидации, которые должны применяться к данным запроса (рис. 3.5).

```
public function rules()
{
    return [
        'name' => 'required|min:3|max:255',
        'phone' => 'required|min:3|max:255',
    ];
}
```

Рисунок 3.5 – Правила валидации

### 3.2 Версия сайта для слабовидящих

На сайте предусмотрена версия для слабовидящих.

Крупный шрифт, отключение изображений, смена цветовой палитры сайта – все это может сделать веб-ресурс доступным для людей с проблемами со зрением. Такая версия сайта была реализована с помощью плагина bvi – button visually impaired.

Button visually impaired [13] – это плагин, который позволяет адаптировать версию сайта для слабовидящих людей. Панель на сайте для слабовидящих позволяет изменять цветовую гамму сайта, масштабировать шрифт, а синтезатор речи озвучит вслух изменения настроек. Таким образом, можно модифицировать сайт так, чтобы он удовлетворял потребности людей с ограниченными возможностями.

С 1 января 2016 года действует поправка к Федеральному закону, согласно которой владельцы сайтов обязаны предоставить версию сайта для слабовидящих, чтобы избежать дискриминации в отношении инвалидов по зрению.

Согласно законам Российской Федерации, каждый интернет-ресурс должен иметь версию для слабовидящих, чтобы не нарушать права инвалидов. Тем не менее, контроль за соблюдением этих норм осуществляется только по отношению к:

- органам муниципальной и государственной власти;
- медицинским учреждениям;
- учебным заведениям;
- учреждениям культуры и искусства;
- социальным учреждениям.

Сайты вышесказанных учреждений обязаны иметь версию для слабовидящих, соответствующую госту Р 52872-2019 «Требования доступности Интернет-ресурсов для инвалидов по зрению».

Ниже представлена главная страница версии сайта для слабовидящих (рис. 3.7). Были полностью отключены картинки, цветовая гамма сайта – белый текст на черном фоне.

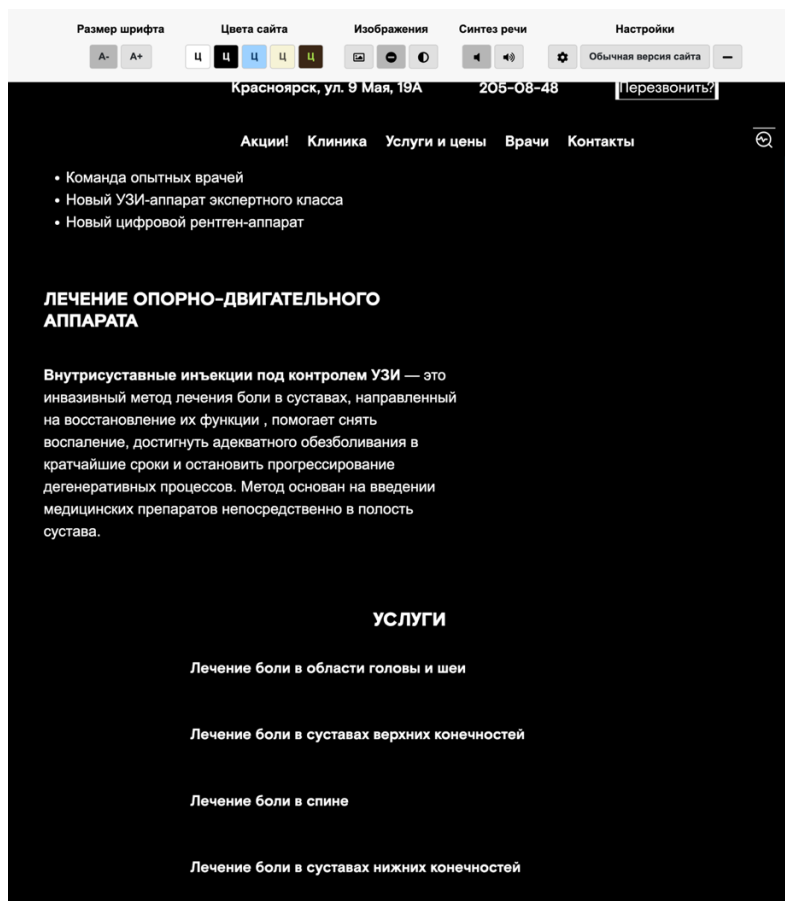


Рисунок 3.7 - Версия сайта для слабовидящих

### 3.3 Раздел «Врачи»

Страница со специалистами клиники содержит список врачей с кратким описанием, фотографией и возможностью записаться по телефону или оставить отзыв (рис. 3.8).



Рисунок 3.8 – Блок с врачами

При отправке отзыва данные отправляются на почту по тому же алгоритму что был описан выше. Помимо этого, в функции реализовано добавление данных в БД.

### 3.4 Раздел «Отзывы»

Страница с отзывами содержит блоки с именем и фамилией пациента, которые при нажатии на них разворачиваются – таким образом можно увидеть текст отзыва (рис. 3.9).



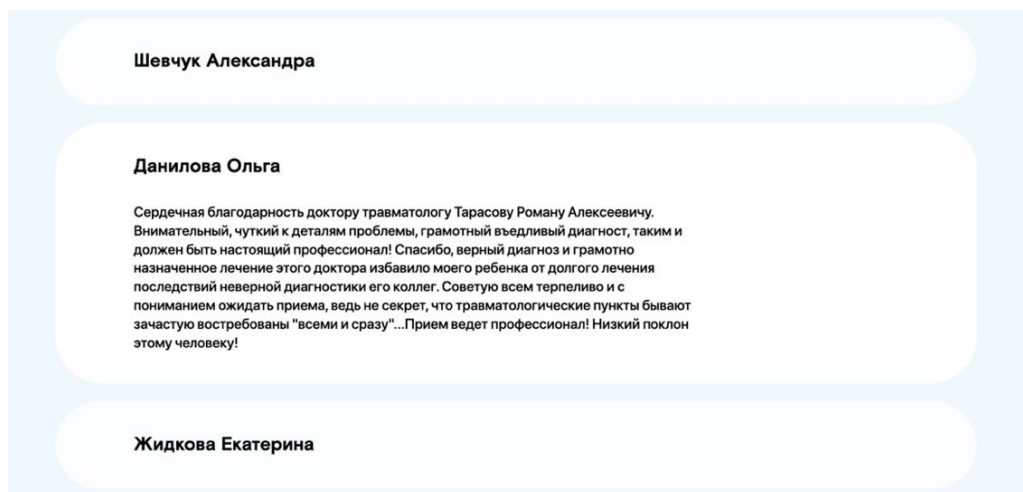


Рисунок 3.9 – Отзывы клиентов

Как было сказано выше, данные из формы при написании отзыва отправляются на почту и в базу данных. Данные в базу данных сохраняются с атрибутом `active`, который равен 0. Таким образом, отзыв не активен и не появляется на странице автоматически. После прохождения модерации, администратор размещает отзыв на сайте с помощью панели управления. Каждый отзыв проходит проверку на соответствие правилам размещения: уникальность, отсутствие ненормативной лексики и соответствие содержанию сайта.

### 3.5 Раздел «Контакты»

Страница с контактами содержит интерактивную Яндекс-карту (рис. 3.10). На сайт она была добавлена с помощью конструктора карт.

Конструктор карт – это визуальный редактор, с помощью которого можно создать собственную карту, используя метки, линии или многоугольники.

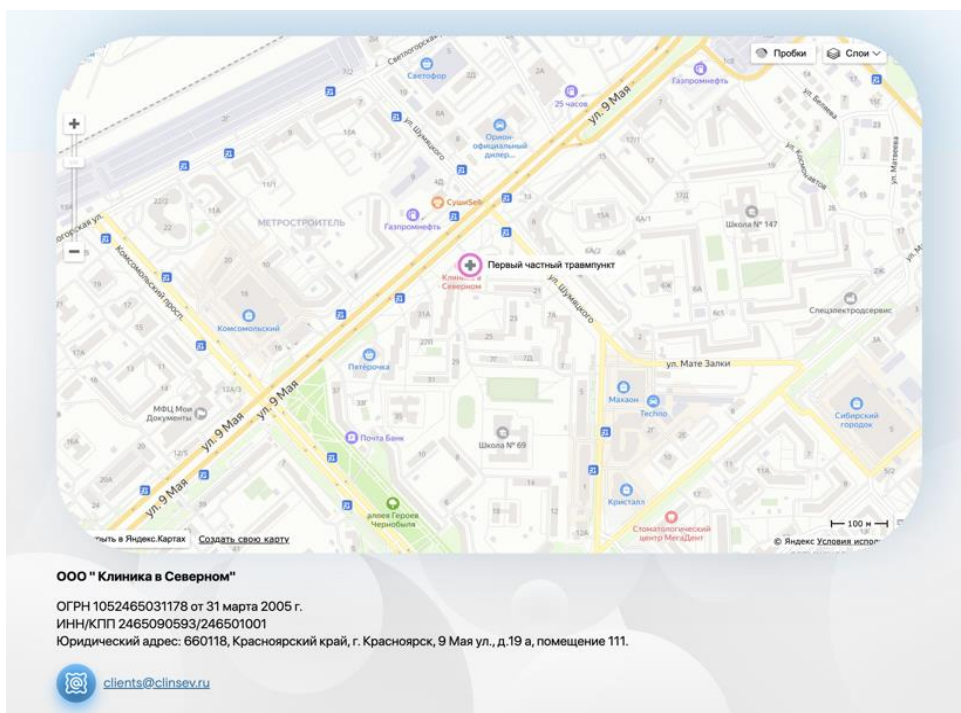


Рисунок 3.10 – Раздел «Контакты»

### 3.6 Услуги

Ниже представлен один из разделов пункта меню «Услуги и цены» – лечение боли в спине (рис. 3.11).

Данный раздел содержит общую информацию о лечении боли в спине, и ссылку на страницу «Почему возникают боли в спине и пояснице?».

В виде слайдера с картинками описана процедура избавления от боли (рис. 3.12). Слайдер выполнен с помощью плагина swiper-bundle. Swiper – это современный бесплатный сенсорный слайдер. Он предназначен для использования в мобильных веб-сайтах.



Рисунок 3.11 – Страница с услугами



Рисунок 3.12 – Слайдер на странице услуг

### 3.7 Панель управления

В этом разделе описаны возможности панели управления и представлен ее интерфейс (рис. 3.13).

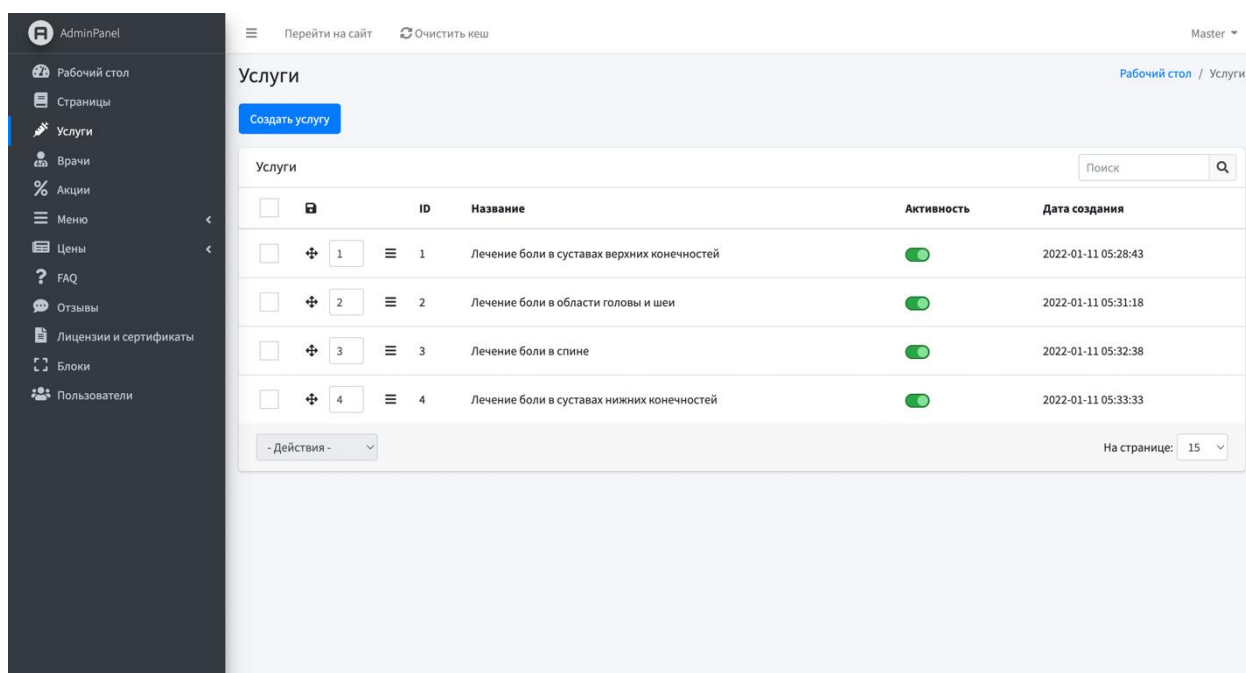


Рисунок 3.13 – Панель управления

В левой части интерфейса вне зависимости от того, на какой именно странице находимся, присутствует панель меню. Если меню находится в свёрнутом виде, при нажатии на знак стрелочки появится всплывающий список подразделов. Если раздел не содержит подразделов, то при нажатии на иконку попадём в соответствующий раздел.

Рассмотрим возможности админ-панели.

Мы можем открыть любой отзыв, меню, запись о враче или услуге, лицензии и сертификаты, вопрос-ответ и акции в режиме редактирования – изменить данные, убрать/добавить запись или отсортировать их. Нам доступна информация о дате создания и внесения изменений. Также из панели управления можно заменить фотографию, выбрав другую с вашего устройства. В каждом разделе админ-панели есть поиск по данным.

Панель управления была сделана с помощью Adminlte. Adminlte – адаптивный шаблон, бесплатная расширенная панель управления. В шаблоне присутствуют все необходимые элементы, такие как: заголовки, списки, таблицы, формы, меню, иконки. Верстка выполнена на высоком уровне, а

также внедрены все необходимые js-элементы. Для того чтобы использовать ее возможности, необходимо подключить файлы css и js к проекту.

Из панели управления можно создать нового пользователя (рис. 3.14). Для него необходимо выбрать роль. Были заведены такие роли:

- Мастер.
- Администратор.
- Менеджер.
- Редактор.

Рисунок 3.14 – Создание нового пользователя

Каждой из ролей доступен определенный функционал. Мастеру доступны все возможности панели управления. Администратор может администрировать инфоблоки, но не имеет прав менять глобальные настройки сайта, например изменять папки и файлы. Менеджер может добавлять новую информацию на сайт. Он обрабатывает отзывы и публикует их на сайт. Редактор может только редактировать сайт, но не имеет возможности удалять разделы.

### 3.8 Структура базы данных

Ниже представлена ER-диаграмма основных сущностей БД (рис. 3.15).

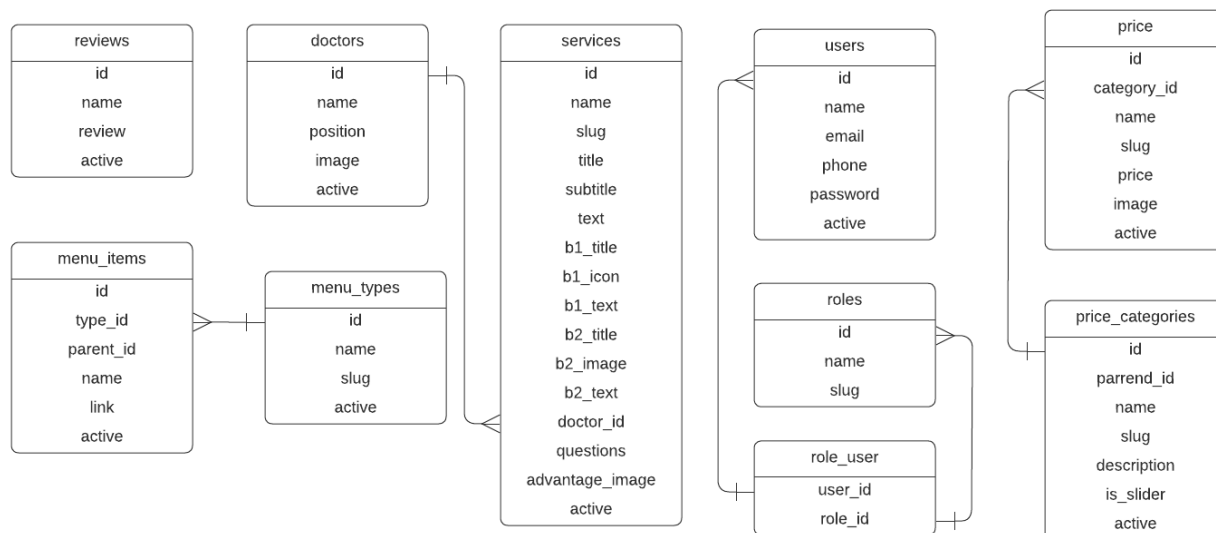


Рисунок 3.15 – ER-диаграмма

ER-диаграмма (схема «сущность-связь») – это разновидность блок-схемы, визуальное представление базы данных, на которой показано, как связаны элементы (люди, объекты, концепции) внутри.

Некоторые таблицы каким-то образом связаны с другими таблицами БД.

Рассмотрим связи, встречающиеся в нашем проекте.

- menu\_types и menu\_items – связь «один ко многим». Один тип меню может соответствовать нескольким пунктам меню.
- doctors и services – один ко многим. Мы будем получать запись врача, связанную с данной услугой.
- Многие ко многим. Такая связь относится к пользователю, который имеет много ролей, где роли относятся ко многим пользователям. Например, несколько пользователей могут иметь роль «admin». Для определения такой связи создаем три таблицы: users, roles, и role\_user. Таблица role\_user является промежуточным звеном, которое связывает

пользователей и роли. Поскольку роль может принадлежать многим пользователям, нельзя просто разместить столбец `user_id` в таблице `role`. Это означало бы, что роль могла принадлежать только одному пользователю. Для обеспечения поддержки ролей, назначаемых нескольким пользователям, необходима таблица `role_user`.

- `price` и `price_categories` – связь «один ко многим». Одна категория может иметь несколько пунктов с ценами.

### **3.9 Интеграция форм на сайте с Битрикс24**

В проекте реализована интеграция форм с CRM-системой Битрикс24 для контроля записей пациентов.

Чтобы провести интеграцию форм на сайте с Битрикс24 необходимо в CRM-системе создать формы с нужным списком полей, выбрать сценарий работы. Можно выбрать способ отсылки данных: ручной или автоматический. В ручном режиме отсылки данные сначала обрабатываются, а после передаются менеджером в CRM. При автоматическом режиме результаты заполнения сразу передаются в CRM.

В результате полученный JavaScript код необходимо разместить в проекте на месте, где должна быть форма.

Заполнив форму на сайте, пользователь нажимает на кнопку «Отправить». Менеджер компании получает сообщение со всей введенной пользователем информацией и приступает к выполнению своих обязанностей – помогает получить полезную информацию, оформляет запись на прием.

Эта функция значительно упрощает работу сотрудников и облегчает процесс взаимодействия клиентов с компанией.

На картинке ниже можно увидеть, как в Битрикс24 выглядит заявка, поступающая с сайта (рис 3.16).

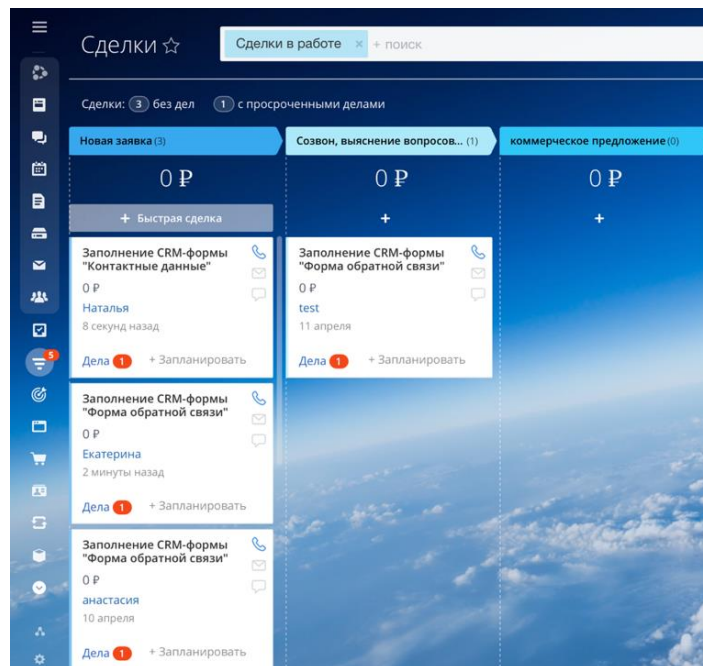


Рисунок 3.16 – Список сделок

Детальное описание заявки представлено на рисунке 3.17.

Функционал Битрикс24 позволяет добавлять комментарии, перезванивать прям из системы или отправлять смс в мессенджеры.

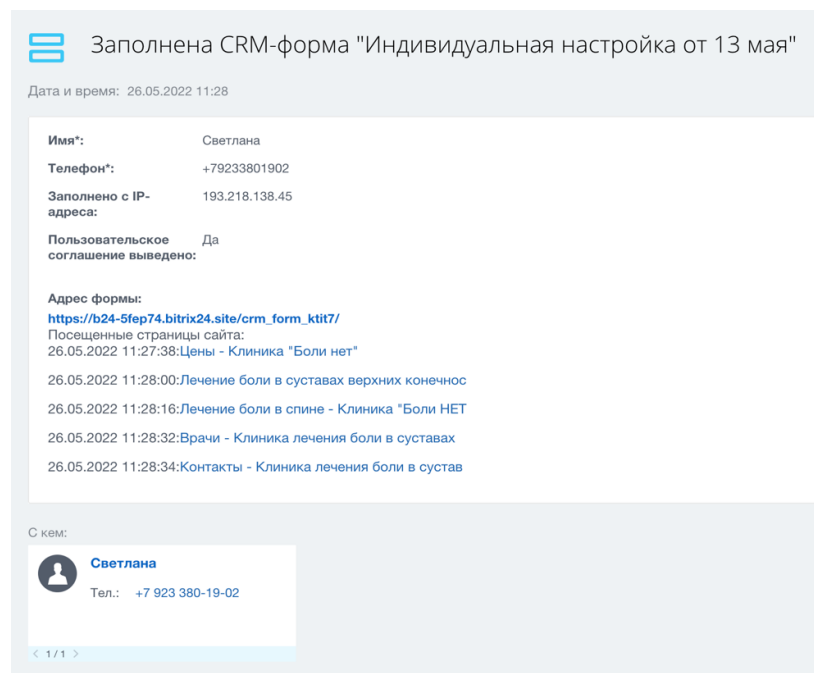


Рисунок 3.17 – Заявка в детальном представлении



### **3.10 Выводы по разделу 3**

В третьем разделе рассмотрена программная реализация веб-сайта. Более детально описаны главная страница сайта, версия сайта для слабовидящих, страница «Врачи», «Отзывы» и «Контакты». Описан интерфейс и возможности панели управления, а также представлена структура базы данных. В пункте «Интеграция форм на сайте с Битрикс24» можно ознакомиться с инструкцией по выполнению интеграции форм и увидеть, как в Битрикс24 выглядят заявки, поступающие с сайта.

## ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была изучена предметная область и существующие на данный момент аналоги.

После изучения аналогов был сформулирован ряд требований, предъявляемый к веб-сайту медицинского центра. На основе сформированных требований были определены технологии разработки. Адаптивная верстка сайта была выполнена с помощью фреймворка Bootstrap, а серверная часть разработана на языке программирования PHP версии 8.1 с использованием MySQL. В качестве серверного фреймворка был выбран Laravel, который предназначен для разработки с использованием архитектурной модели MVC. Такой выбор обусловлен тем, что Laravel это мощный PHP-фреймворк с неограниченными возможностями, а веб-сайты на Laravel обеспечивают высокую производительность и безопасность.

Проект Laravel был создан с помощью docker, в контейнере, который не мешает установленному на локальном компьютере программному обеспечению или его конфигурации. Docker обеспечивает удобное развертывание приложения.

В результате был разработан и протестирован веб-сайт, который позволяет просматривать информацию о клинике и врачах, о предоставляемых услугах и ценах, ознакомиться с часто задаваемыми вопросами, оставить отзыв или заявку на обратный звонок. Веб-сайт имеет панель управления, благодаря которой можно удобно и легко администрировать сайт. Реализована интеграция веб-сайта с CRM-системой Битрикс24 для управления поступающими заявками. Главной функцией сайта является возможность записаться на прием онлайн, оставив заявку через форму.

## **СПИСОК СОКРАЩЕНИЙ**

БД – База данных

JSON – JavaScript Object Notation

JS – JavaScript

CSS – Cascading Style Sheets

AJAX – Asynchronous Javascript and XML

СУБД – Система управления базами данных

ОС – Операционная система

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Электронный фонд правовых и нормативно-технических документов. ГОСТ Р 52872-2019» [Электронный ресурс] – Режим доступа: <https://docs.cntd.ru/document/1200167693>.
2. Медицинский центр «СМ-Клиника» [Электронный ресурс] – Режим доступа: <https://www.smclinic.ru>.
3. Медицинский центр Лорри [Электронный ресурс] – Режим доступа: <https://lorri-med.ru/price>.
4. Медицинский центр iNEO [Электронный ресурс] – Режим доступа: <https://ineo24.ru>.
5. HTML Academy [Электронный ресурс] – Режим доступа: <https://htmlacademy.ru/tutorial/php/basics>.
6. SkillFactory.блог. Framework. [Электронный ресурс] – Режим доступа: <https://blog.skillfactory.ru/glossary/framework/>
7. ИТШЕФ. Bootstrap. [Электронный ресурс] – Режим доступа: <https://itchief.ru/bootstrap/introduction>.
8. SkillFactory.блог. Laravel. [Электронный ресурс] – Режим доступа: <https://blog.skillfactory.ru/glossary/laravel/>
9. HelpReg. Базы данных. [Электронный ресурс] – Режим доступа: <https://help.reg.ru/hc/ru/articles/4408047763217-Что-такое-MySQL>
10. JAVASCRIPT.RU. Введение в AJAX. [Электронный ресурс] – Режим доступа: <https://learn.javascript.ru/ajax-intro>
11. Selectel сообщество. Что такое Docker: для чего он нужен и где используется. [Электронный ресурс] – Режим доступа: <https://selectel.ru/blog/what-is-docker/>
12. Laravel Russian Community. Кеш приложения. [Электронный ресурс] – Режим доступа: <https://laravel.su/docs/8.x/cache>
13. Button Visually impaired. [Электронный ресурс] – Режим доступа: <https://bvi.isvek.ru>.

## ПРИЛОЖЕНИЕ А

### Исходный код функции отображения раздела «Услуги»

```
<?php

namespace App\Http\Controllers;

use App\Models\Doctor;
use App\Models\Promotion;
use App\Models\Service;
use Illuminate\Http\Request;

class ServiceController extends Controller
{
    public function __invoke($slug)
    {
        $service = Service::where('slug', $slug)
            ->where('active', 1)
            ->firstOrFail();

        $doctors = Doctor::whereIn('id', $service->doctor_id->get());

        $promotion = Promotion::where('active', true)
            ->orderBy('created_at', 'desc')
            ->first();

        if ($promotion) {
            if ($promotion->image)
                $promotion->image = getResizeImage($promotion->image, ['w' => 470, 'h' =>
null, 'method' => 'resize']);
        }

        $composition = [
            1 => ['point' => 'header_arm_active..png', 'image' => 'header_arm..png'],
            2 => ['point' => '', 'image' => 'neck_skeleton.png'],
            3 => ['point' => '', 'image' => 'back_skeleton.png'],
            4 => ['point' => 'header_knee_active..png', 'image' => 'header_knee..png'],
        ];

        $appointment = [
            1 => ['default' => 'arm..png', 'hidden' => 'arm-active..png'],
            2 => ['default' => 'leg..png', 'hidden' => 'leg-active..png'],
            3 => ['default' => 'ankle..png', 'hidden' => 'ankle-active..png'],
            4 => ['default' => 'knee..png', 'hidden' => 'knee-active..png'],
        ];

        $slider = [
            1 => ['service_arm_1.jpg', 'service_arm_2.jpg', 'service_arm_3.png'],
            2 => ['service_neck_1.jpg', 'service_neck_2.jpg', 'service_neck_3.jpg'],
        ];
```

```

        3 => ['blokada1.jpg', 'blokada2.jpg', 'blokada3.jpg'],
        4 => ['procedure-slide-1.jpg', 'procedure-slide-2.jpg', 'procedure-slide-3.jpg'],
    ];

    return view('service.show', [
        'service' => $service,
        'promotion' => $promotion,
        'composition' => $composition,
        'appointment' => $appointment,
        'slider' => $slider,
        'doctors' => $doctors
    ]);
}
}

```



Министерство науки и высшего образования РФ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

 О.В. Непомнящий

подпись

инициалы, фамилия

«20» 06 2022 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника  
код и наименование специальности

Веб-сайт медицинского центра  
тема

Руководитель

 14.06.2022

подпись, дата

доцент,

канд. техн. наук

должность, ученая степень

М. С. Медведев

инициалы, фамилия

Выпускник

Маш- 14.06.2022

подпись, дата

Н. В. Машковская

инициалы, фамилия

Нормоконтролер

 14.06.2022

подпись, дата

доцент,

канд. техн. наук

должность, ученая степень

М. С. Медведев

инициалы, фамилия

Красноярск 2022