**СОДЕРЖАНИЕ**

Введение 4

1 Технический проект 7

1.1 Анализ предметной области 7

1.2 Постановка задачи 8

1.3 Требования к техническим средствам и программному обеспечению 9

1.4 Проектирование веб-сайта 10

1.5 Проектирование структуры базы данных 15

1.6 Выбор средств реализации 17

2 Рабочий проект 20

2.1 Логическая структура базы данных 20

2.2 Функциональные модули веб-сайта 24

2.3 Файловая структура веб-сайта 25

2.4 Работа с веб-сайтом 28

Заключение 32

Список использованных источников 33

Приложение А (справочное) Листинг кода для добавления отзывов 34

Приложение Б (справочное) Листинг кода для добавления новостей 36

**ВВЕДЕНИЕ**

Темой курсового проекта является разработка сайта для организации «ЦСМ» Адыгейский филиал. Целью данного курсового проекта будет являться написание сайта, позволяющего обеспечить эффективную коммуникацию с гражданами и предоставления им необходимой информацию.

Написание данного курсового проекта является целесообразным, потому что:

* сайт позволит получить необходимую информации о государственной организации, ее структуре, деятельности и контактной информации. Это позволит им получить актуальные данные без необходимости обращаться в организацию лично или по телефону;
* сайт позволит пользователям задавать вопросы, оставлять отзывы или подавать жалобы прямо через сайт. Это сократит время и усилия, необходимые для обращения в государственную организацию и позволит ускорить процесс получения ответов или решения проблем;
* сайт позволит пользователям ознакомиться с документами и материалами, законодательством, нормативными актами и другой информацией, связанной с работой государственной организации. Это поможет улучшить прозрачность и доступность информации для граждан.

Для достижения поставленной цели будут рассмотрены и решены следующие задачи:

* разработан удобный и интуитивно понятный интерфейс сайта;
* размещено на сайте полная и актуальная информация о деятельности организации, включая основные направления работы, проводимые мероприятия и достижения;
* предоставлен доступ к законодательству и нормативным документам, регулирующим деятельность организации;
* разработана страница «Услуги», где пользователи смогут ознакомиться со списком предоставляемых организацией услуг и процедурами их получения;
* создана новостная секции, где будут публиковаться актуальные новости, события и изменения в деятельности организации;
* размещена контактная информация для связи с организацией, включая телефоны, электронную почту и адреса офисов;
* создана база знаний или справочника, где пользователи смогут найти ответы на часто задаваемые вопросы и разъяснения по вопросам, связанных с деятельностью организации;
* обеспечена возможность онлайн-записи на прием или получение услуг организации через сайт;
* внедрена система обратной связи, чтобы пользователи могли оставлять отзывы, задавать вопросы или отправлять предложения по улучшению работы организации и ее сайта.

Объектом исследования будет организация «ЦСМ» Адыгейский филиал.

Предметом исследования будут являться средства реализации функциональности и контента сайта, а также удобного и интуитивно понятного интерфейса.

Практическая значимость проекта заключается в облегчении доступа граждан к информации и услугам организации «ЦСМ» Адыгейский филиал, повышении эффективности работы, а также в возможности расширения функционала и внедрении новых технологий для улучшения работы и удовлетворения потребностей граждан.

Основными пользователями данной программы будут граждане, сотрудники организации, уполномоченные органы и контролирующие организации, представители других государственных организации.

Теоретическими основами и методами решения поставленных задач будут методологии проектирования функциональности приложения, структуры базы данных (БД), особенности реализации программного кода с помощью конкретного языка программирования, принципы и алгоритмы применения инструментальных средств на различных этапах разработки программного продукта, а также знания принципов дизайна пользовательского интерфейса и требований к государственным сайтам.

# **1 ТЕХНИЧЕСКИЙ ПРОЕКТ**

## Анализ предметной области

Сегодня стандартизация является частью современной предпринимательской стратегии. Ее влияние и задачи охватывают все сферы общественной жизни. Так, стандарты на процессы и документы (управленческие, товаросопроводительные, технические) содержат те правила, которые должны знать и выполнять специалисты промышленности и торговли для заключения взаимовыгодных сделок, а приемы технического регулирования являются инструментами обеспечения не только безопасности, конкурентоспособности, но и эффективного партнерства изготовителя, заказчика и продавца на всех уровнях правления.

Стандартизация – деятельность по разработке (ведению), утверждению, изменению (актуализации), отмене, опубликованию и применению документов по стандартизации и иная деятельность, направленная на достижение упорядоченности в отношении объектов стандартизации. Изначально стандарты тесно связаны с торговлей. Свободное перемещение товаров и услуг по всему миру без торговых барьеров – конечно же горячее желание каждого производителя и поставщика. Однако без некоего технического регулирования, учитывающего интересы всех участников рыночных отношений, неизбежны трудности совместимости продуктов, процессов, отсутствие гарантий ответственности. Соглашения по формальной стандартизации облегчают жизнь покупателям и продавцам во всем мире. Сегодня рынки становятся глобальными, и цепочки поставок могут пересекать границы многих стран. B деле создания более благоприятных условий для международной торговли могут помочь международные соглашения и стандарты.

Разрабатываемый сайт будет содержать материал по основной информации о центре стандартизации, а также будут выполнены вышеперечисленные задачи.

## Постановка задачи

Необходимо разработать сайт для государственной организации, который будет предоставлять информацию о деятельности организации, услугах для граждан, новостях и других важных данных.

Данная программа должна будет содержать все необходимые сведения об организации «ЦСМ» Адыгейский филиал, иметь удобный и интуитивно понятный пользователю графический интерфейс.

Программа должна обеспечивать выполнение следующих функций:

* предоставление в главном окне основной информации об организации: наименование, цели и задачи;
* предоставление информации о структуре организации, включая подразделения, их функции и контактные данные;
* доступ к актуальным новостям и событиям;
* отображение информации о предоставляемых организацией услугах и возможности онлайн записи на прием;
* обеспечение возможности онлайн-заполнения и отправки различных форм и анкет;
* предоставление доступа к актуальным документам, приказам, положениям и другой документации, относящейся к работе организации;
* хранение материалов в БД.

Программный продукт должен включать в себя три части:

* дизайнерскую часть, которая обеспечит удобную работу с сайтом, обеспечит удобный и интуитивно понятный интерфейс для навигации по сайту;
* функциональную часть, в которой будет реализована возможность поиска информации на сайте, разработана система управления контентом, создана отдельная страница для публикации новостей и анонсов организации, разработана форма обратной связи, реализована интеграция с социальными сетями и возможность публикации материалов сайта в них;
* тестовую часть, в которой будет производиться тестирование и отладка сайта для обеспечения его стабильной работы и высокой производительности.

Входными данными для сайта будет информация о государственной организации, ее структуре, услугах, проектах, новостях, анонсах.

Выходными данными для сайта будут отображение этих данных в удобном и понятном для пользователя формате в виде текстовой информации, изображения, видео, документы для загрузки, ссылки на другие ресурсы и так далее.

Приложение будет состоять из страниц, на которых будет располагаться вся необходимая пользователю информация.

Способ решения поставленных задач зависит от выбора средств разработки.

Тестирование сайта будет проводиться путем выполнения различных сценариев использования сайта, чтобы убедиться в корректной работе всех функций и компонентов.

Блок тестирования будет проверяться путем запуска теста и эмуляций работы различных модулей программы отдельно для проверки их функциональности и взаимодействия с другими компонентами.

## Требования к техническим средствам и программному обеспечению

Для корректной работы разрабатываемого сайта необходимо, чтобы компьютер удовлетворял следующим минимальным требованиям:

* процессор LGA775 Intel Celeron Dual-Core E3300;
* объем свободной оперативной памяти DIMM DDR2 2GB;
* видеоадаптер PCI-E 1GB Sapphire Radeon HD 4650;
* 100 Мб свободного места на жестком диске;
* стандартная клавиатура с 101/109 клавишами;
* манипулятор типа «Мышь».

Для разработки веб-сайта на компьютере должны быть установлены следующие программы:

* любой популярный на сегодняшний день браузер (Google Chrome, Yandex, Microsoft Edge или любой другой);
* Adobe Photoshop CC 2015;
* Corel Draw Pro X7;
* Brackets;
* интерпретатор Python;
* фреймворк Django;
* Java;
* Adobe Flash Player;
* Bootstrap 3;
* JetBrainsPyCharm 5.

Также для правильной работы скриптов программного продукта необходимо активное интернет-соединение и разрешение использования его в браузерах.

## Проектирование веб-сайта

Разрабатываемый сайт будет являться информационным ресурсом.

По стилю оформления и функциональности выделяют следующие типы сайтов:

* статические;
* динамические;
* флэш-сайты.

Статические сайты сделаны по классической технологии HTML. Такие веб-страницы написаны полностью на языке разметки HTML. Статическими они называются потому, что содержимое сайта (контент) всегда будет неизменным. Плюсом такого вида разработки является его простота и нетребовательность к аппаратным ресурсам. Основным минусом является отсутствие интерактивности. Кроме того, в таких сайтах сложнее изменить информацию и дизайн страниц. Они не удовлетворяют современным требованиям к сайтам и сейчас практически не используются.

Динамические сайты не существуют в неизменном виде на сервере, а формируются при помощи скриптов. Кроме того, пользователи могут посылать какие-либо запросы на сервер и получать необходимую им информацию, например, осуществлять поиск по товарам. Плюсами таких сайтов является интерактивность, большая функциональность и легкость изменения дизайна. Минусы динамических сайтов в том, что они требуют поддержку дополнительных технологий, создают большую нагрузку на сервер, а использование скриптов представляет большую потенциальную угрозу для безопасности.

Флеш-технология позволяют создавать очень красивые, со звуком и анимацией, динамичные сайты. Для их реализации требуются высококвалифицированные команды разработчиков. Плюсами таких сайтов является интерактивность и большая функциональность. Но из-за своей сложности и насыщенности мультимедийными элементами они занимают большой объем памяти, что значительно увеличивает время загрузки. По этой причине они не получили широкого распространения [9].

Исходя из рассмотренного выше, разрабатываться будет динамический сайт.

Проектирование и разработка сайта включают в себя следующие этапы:

* определение структурной схемы сайта – расположения контента, разделов и элементов навигации;
* разработка веб-дизайна – создание графических элементов макета сайта, стилей и элементов навигации;
* разработка модулей, БД и других элементов;
* разработка взаимодействия модулей и БД;
* тестирование и размещение сайта в сети Интернет.

Сайт будет представлять многостраничный LandingPage, что в переводе на русский означает «целевая страница». Особенностью данной структурной схемы является то, что основное содержание сайта располагается на одной странице, а дополнительные возможности вынесены на отдельные страницы.

Логическая структура сайта – это некий порядок, который определяет логическое расположение страниц сайта, а также отдельных его частей при отображении сайта на экране. Данный термин объединяет в себе два понятия, такие как внутренняя и внешняя структуры сайта.

Внутренняя структура сайта – это упорядоченное размещение страниц сайта относительно главной страницы. Она представлена на рисунке 1.

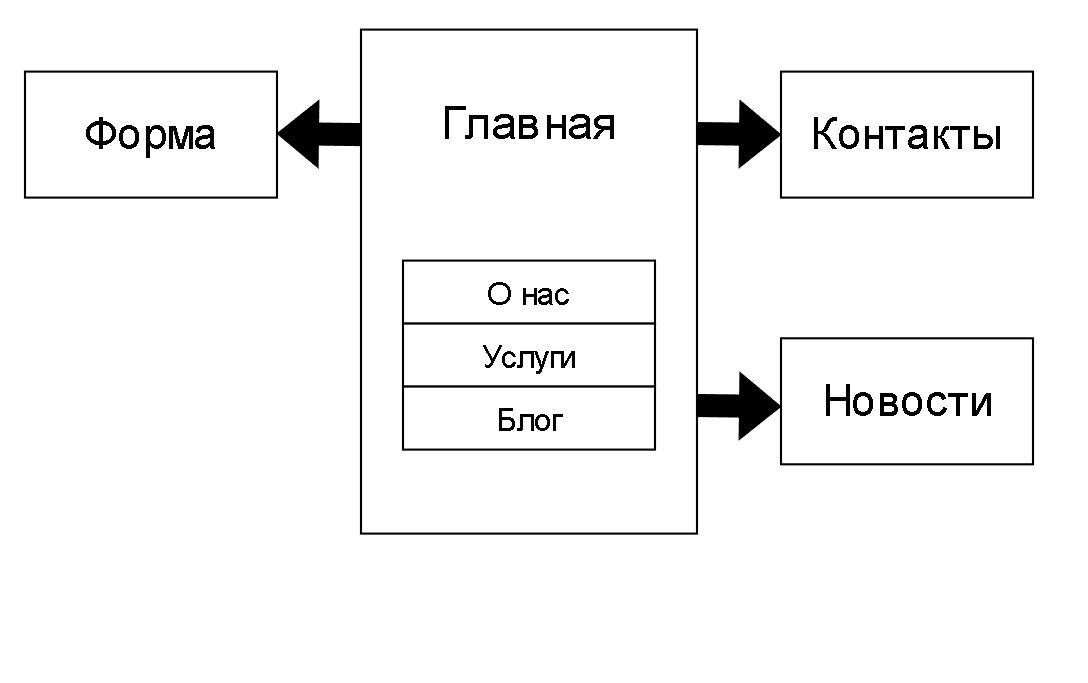


Рисунок 1 – Внутренняя структура веб-сайта

На главной странице сайта будут расположены следующие разделы:

* отображение информации об организации (О нас);
* отображение информации об услугах организации (Услуги);
* отображение новостей организации (Блог).

Внешняя структура сайта – это некая блок-схема сайта, которая определяет порядок размещения отдельных элементов сайта на его страницах. То есть, где должен располагаться основной текст, меню сайта, дополнительные информационные блоки и так далее. Она представлена на рисунке 2.



Рисунок 2 – Внешняя структура главной страницы

С помощью «Меню» пользователю будет предоставляться возможность перемещаться по разделам сайта, а также по его страницам

В шапке сайта будут располагаться краткое описание организации и форма для записи на консультацию.

Страница «О нас» будет содержать информацию о рабочем процессе.

На странице «Услуг» будет отображаться спектр услуг организации.

На странице «Блог» будут отображаться последние добавленные новости организации.

В подвале сайта будет располагаться необходимая дополнительная информация и различные элементы.

Для решения поставленных задач необходимы следующие программные модули:

* добавление фотографий для новостей организации;
* обратная связь;
* отображение информации об организации;
* добавление новостей об организации;
* отображение контактной информации.

Данные модули будут доступны только на главной странице.

На главной странице будет располагаться форма обратной связи для записи на консультацию.

На странице новостей будет доступен модуль добавления новостной информации.

Схема функционального взаимодействия модулей с внутренней структурой разрабатываемого сайта представлена на рисунке 3.

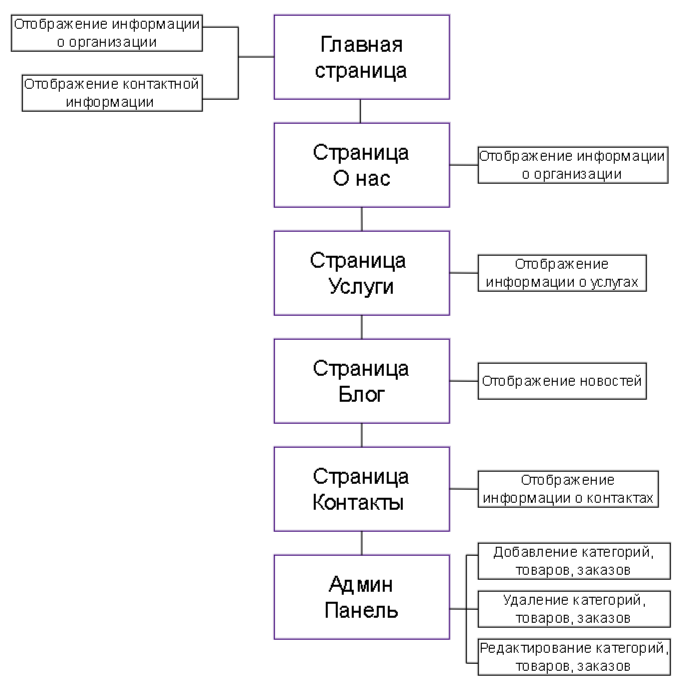
****

Рисунок 3 – Функциональные модули веб-приложения

## Проектирование структуры базы данных

Для реализации полнофункционального сайта необходимо использование БД.

В ходе исследования предметной области был выявлен набор сущностей, информацию о которых необходимо хранить в БД:

* запись на консультацию;
* различные новости и мероприятия организации;
* форма обратная связи.

На рисунке 4 представлена концептуальная модель БД «CSM».

Каждая из представленных сущностей хранит независимые данные, которые будут использоваться отдельными модулями проектируемой системы. Исходя из этого, каждая сущность будет преобразована в отдельную БД, состоящую из одной таблицы.

При взаимодействии с БД будет использоваться технология клиент-сервер. В основе этой концепции лежит идея о том, что помимо хранения файлов БД, центральный процессор должен выполнять основную часть обработки данных. В отличие от других технологий, технология клиент-сервер после обработки запроса выдает пользователю только ту информацию, которая ему необходима, а не весь набор данных. Таким образом, предотвращается перегрузка сети серверами.

Кроме того, можно выделить следующие преимущества клиент-серверной системы:

* позволяет объединить клиентов с разными аппаратными платформами и операционными системами;
* существует возможность распределения функций вычислительной системы между несколькими независимыми компьютерами в сети. Это позволяет упростить обслуживание вычислительной системы. В частности, замена, ремонт, модернизация или перемещение сервера не затрагивают клиентов;
* все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

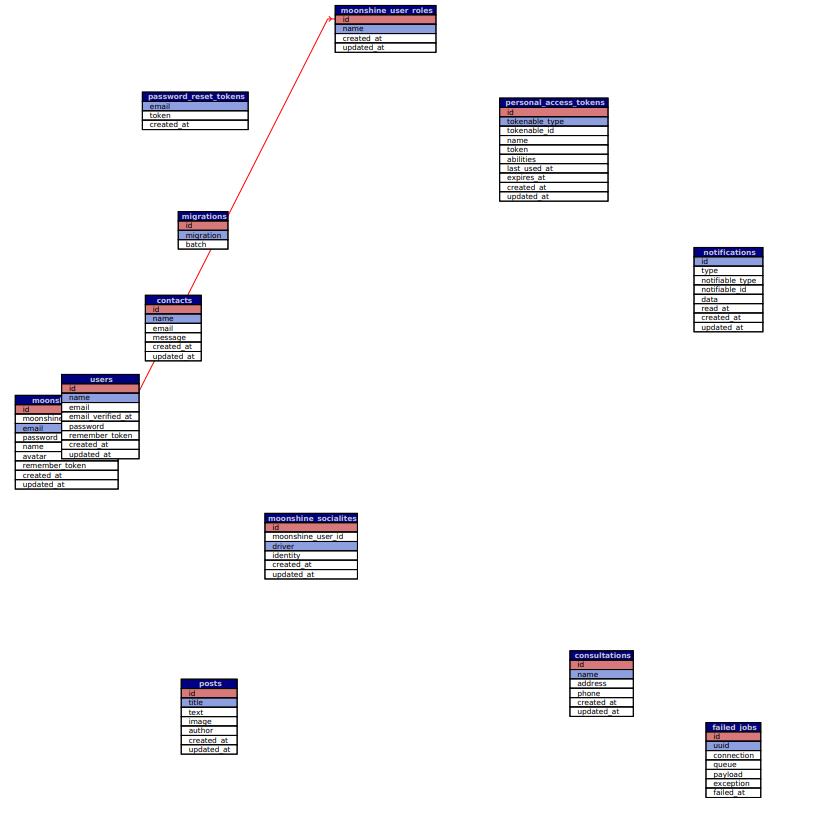


Рисунок 4 – Концептуальная модель базы данных «ЦСМ»

## Выбор средств реализации

Существуют определенные **технологии создания сайтов**, которыми необходимо отлично владеть, чтобы создать хороший интерактивный сайт. Рассмотрим наиболее популярные современные средства создания сайтов.

PHP (Hypertext Preprocessor) – это скриптовый язык программирования, преимущественно используемый для веб-разработки. Он часто встроен непосредственно в HTML-код, и его основное применение – создание динамических веб-сайтов и веб-приложений. PHP работает на серверной стороне, генерируя контент «на лету», что позволяет создавать интерактивные сайты с возможностью взаимодействия с БД, обработки форм, сессий и многого другого.

JavaScript – это высокоуровневый, интерпретируемый язык программирования, который преимущественно используется для создания динамических и интерактивных веб-страниц.

JavaScript относится к динамическим языкам программирования и выполняется на стороне клиента (в браузере пользователя). Он позволяет изменять содержимое веб-страницы без необходимости перезагрузки всей страницы. Также он обеспечивает взаимодействие с пользователем (например, обработку кликов, наведения мыши, отправки форм). JavaScript позволяет проверять данные, введенные в формы, до их отправки на сервер.

С помощью технологии AJAX, основанной на JavaScript, можно отправлять и получать данные от сервера без перезагрузки страницы. При этом он тесно интегрирован с HTML и CSS, что позволяет создавать динамические, стильные и интерактивные веб-приложения.

Существует множество библиотек и фреймворков, таких как jQuery, React, Angular и Vue.js, которые расширяют возможности JavaScript и упрощают процесс разработки.

Большинство современных веб-браузеров поддерживают JavaScript. Также существуют инструменты и методы для обеспечения совместимости с различными браузерами и устройствами. Однако важно учитывать аспекты безопасности при разработке с использованием JavaScript, так как неправильное использование может привести к уязвимостям, таким как межсайтовый скриптинг (XSS).

JavaScript является одним из самых популярных и широко используемых языков программирования. Благодаря активному сообществу разработчиков постоянно появляются новые инструменты, библиотеки и решения для улучшения производительности и функциональности веб-приложений.

Язык HTML (HyperText Markup Language) – это основной язык разметки, используемый для создания веб-страниц. Он предоставляет структуру и семантику контента, позволяя определить, какие элементы на странице являются заголовками, параграфами, ссылками, изображениями и так далее. HTML используется для описания содержимого веб-страницы, а не её визуального представления. Визуальное оформление страницы определяется с помощью CSS (Cascading Style Sheets).

HTML состоит из различных элементов, таких как теги, атрибуты и текстовое содержимое. Теги используются для определения начала и конца элемента, например <p> для параграфа или <img> для изображения. Атрибуты предоставляют дополнительную информацию об элементе, такую как его идентификатор, класс, ссылку на изображение и так далее. Текстовое содержимое вставляется между открывающим и закрывающим тегами элемента.

HTML также поддерживает использование различных типов элементов, таких как формы для сбора информации от пользователей, таблицы для представления данных, списки для структурирования информации и многие другие.

Основной целью HTML является создание структурированного контента, который может быть интерпретирован браузерами и отображен в виде веб-страницы. В современном веб-разработке HTML часто комбинируется с другими технологиями, такими как CSS для стилизации и JavaScript для добавления интерактивности на страницу.

В фреймворке Laravel миграции (migrations) представляют собой способ организации БД и изменения её схемы с использованием чистого PHP вместо SQL. Миграции обеспечивают версионность БД, что делает процесс разработки и развертывания более удобным и надежным.

Основные преимущества использования миграций в Laravel:

* позволяют управлять изменениями структуры БД, сохраняя каждое изменение в коде. это упрощает совместную работу над проектом и переход между версиями БД;
* обеспечивают портабельность схемы БД, что упрощает перенос приложения между различными окружениями (например, с локального сервера на продакшен-сервер).
* можно легко идемпотентно применять изменения БД на различных этапах разработки, начиная от создания таблиц и заканчивая их изменением или удалением.

# **2 РАБОЧИЙ ПРОЕКТ**

## 2.1 Логическая структура базы данных

На основе полученной в техническом проекте концептуальной модели БД и после приведения таблиц к третьей нормальной форме была получена логическая модель БД (таблицы 1-12).

Таблица 1 – Таблица «Консультации» (consultations)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark157) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [name](#_bookmark158) | varchar(255) |  | Нет |  |
| [address](#_bookmark159) | text |  | Нет |  |
| [phone](#_bookmark161) | varchar(255) |  | Нет |  |
| [created\_at](#_bookmark163) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark165) | timestamp |  | Да | По умолчанию NULL |

Таблица 2 – Таблица «Контакты» (contacts)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark111) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [name](#_bookmark112) | varchar(255) |  | Нет |  |
| [email](#_bookmark113) | varchar(255) |  | Нет |  |
| [message](#_bookmark114) | varchar(255) |  | Нет |  |
| [created\_at](#_bookmark115) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark116) | timestamp |  | Да | По умолчанию NULL |

Таблица 3 – Таблица «Миграции» (Migrarions)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark118) | int(10) | UNSIGNED | Нет | auto\_increment |
| [migration](#_bookmark119) | varchar(255) |  | Нет |  |
| [batch](#_bookmark120) | int(11) |  | Нет |  |

Таблица 4 – Таблица «Неудачные задачи» (failed\_jobs)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark169) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [uuid](#_bookmark171) | varchar(255) |  | Нет |  |
| [connection](#_bookmark173) | text |  | Нет |  |
| [queue](#_bookmark175) | text |  | Нет |  |
| [payload](#_bookmark176) | longtext |  | Нет |  |
| [exception](#_bookmark177) | longtext |  | Нет |  |
| [failed\_at](#_bookmark178) | timestamp |  | Нет | current\_tim estamp() |

Таблица 5 – Таблица «Авторизация через социальные сети» (moonshine\_socialites)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark147) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [moonshine\_user\_id](#_bookmark156) | bigint(20) | UNSIGNED | Нет |  |
| [driver](#_bookmark159) | varchar(255) |  | Нет |  |
| [identity](#_bookmark163) | varchar(255) |  | Нет |  |
| [created\_at](#_bookmark168) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark173) | timestamp |  | Да | По умолчанию NULL |

Таблица 6 – Таблица «Уведомления» ([notifications](#_bookmark148))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | По умолч |
| [id](#_bookmark157) | char(36) |  | Нет |  |
| [type](#_bookmark160) | varchar(255) |  | Нет |  |
| [notifiable\_type](#_bookmark165) | varchar(255) |  | Нет |  |
| [notifiable\_id](#_bookmark170) | bigint(20) | UNSIGNED | Нет |  |
| [data](#_bookmark175) | text |  | Нет |  |
| [read\_at](#_bookmark178) | timestamp |  | Да | По умолчанию NULL |
| [created\_at](#_bookmark181) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark184) | timestamp |  | Да | По умолчанию NULL |

Таблица 7 – Таблица «Пользователи» ([moonshine\_users](#_bookmark209))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark210) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [moonshine\_user\_role\_id](#_bookmark211) | bigint(20) | UNSIGNED | Нет |  |
| [email](#_bookmark212) | varchar(190) |  | Нет |  |
| [password](#_bookmark213) | varchar(60) |  | Нет |  |
| [name](#_bookmark221) | varchar(255) |  | Нет |  |
| [avatar](#_bookmark225) | varchar(255) |  | Да | По умолчанию NULL |
| [remember\_toke n](#_bookmark229) | varchar(100) |  | Да | По умолчанию NULL |
| [created\_at](#_bookmark233) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark236) | timestamp |  | Да | По умолчанию NULL |

Таблица 8 – Таблица «Роли пользователей» (moonshine\_user\_roles)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark187) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [name](#_bookmark188) | varchar(255) |  | Нет |  |
| [created\_at](#_bookmark189) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark190) | timestamp |  | Да | По умолчанию NULL |

Таблица 9 – Таблица «Пользователи» (users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark167) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [name](#_bookmark172) | varchar(255) |  | Нет |  |
| [email](#_bookmark176) | varchar(255) |  | Нет |  |
| [email\_verified\_at](#_bookmark179) | timestamp |  | Да | По умолчанию NULL |
| [password](#_bookmark182) | varchar(255) |  | Нет |  |
| [remember\_toke n](#_bookmark185) | varchar(100) |  | Да | По умолчанию NULL |
| [created\_at](#_bookmark192) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark193) | timestamp |  | Да | По умолчанию NULL |

Таблица 10 – Таблица «Токен восстановления пароля» (password\_reset\_tokens)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [email](#_bookmark223) | varchar(255) |  | Нет |  |
| [token](#_bookmark227) | varchar(255) |  | Нет |  |
| [created\_at](#_bookmark231) | timestamp |  | Да | По умолчанию NULL |

Таблица 11 – Таблица «Заказы» ([orders](#_bookmark148) )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark215) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [first\_name](#_bookmark216) | varchar(255) |  | Нет |  |
| [last\_name](#_bookmark217) | varchar(255) |  | Нет |  |
| [email](#_bookmark218) | varchar(255) |  | Нет |  |
| [street\_address](#_bookmark222) | varchar(255) |  | Нет |  |
| [city](#_bookmark226) | varchar(255) |  | Нет |  |
| [region](#_bookmark230) | varchar(255) |  | Нет |  |
| [postal\_code](#_bookmark234) | varchar(255) |  | Нет |  |
| [phone](#_bookmark237) | varchar(255) |  | Нет |  |
| [product\_id](#_bookmark239) | bigint(20) | UNSIGNED | Да | По умолчанию NULL |
| [summ](#_bookmark240) | int(11) |  | Нет |  |
| [created\_at](#_bookmark241) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark242) | timestamp |  | Да | По умолчанию NULL |

Таблица 12 – Таблица «Сообщение» (posts)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Столбец | Тип | Атрибуты | Null | Дополнительно |
| [id](#_bookmark162) | bigint(20) | UNSIGNED | Нет | auto\_increment |
| [title](#_bookmark164) | varchar(255) |  | Нет |  |
| [text](#_bookmark166) | text |  | Нет |  |
| [image](#_bookmark168) | varchar(255) |  | Нет |  |
| [author](#_bookmark170) | varchar(255) |  | Да | По умолчанию NULL |
| [created\_at](#_bookmark172) | timestamp |  | Да | По умолчанию NULL |
| [updated\_at](#_bookmark174) | timestamp |  | Да | По умолчанию NULL |

Физическая модель была реализована с помощью миграций Laravel, они хранятся в виде упорядоченных файлов, которые выполняются один за другим с помощью команды php artisan migrate. При выполнении этой команды выполняется запрос в БД, который создаёт таблицы с указанными полями в файле.

## 2.2 Функциональные модули веб-сайта

На основе технического проекта был разработан веб-сайт, предоставляющий информацию об организации «ЦСМ» Адыгейский филиал. Данный сайт состоит их трех страниц, содержащих всю необходимую функциональность.

При разработке сайта использовались следующие системные модули языка программирования Python:

* os.py – предоставляет функций для работы с операционной системой;
* sys.py – обеспечивает доступ к некоторым переменным и функциям, взаимодействующим с интерпретатором Python.

Также были использованы следующие атрибуты модулей:

* render\_to\_response (из модуля shortcuts.py) – отправка результата функции в шаблон Django для последующей обработки;
* models (из модуля db.py) – создание таблиц в базе данных;
* admin (из модуля contrib.py) – отображение административной страницы Django, а также регистрация компонентов на этой странице;
* url (из модуля conf.urls.py) – используется для создания URL-адресов сайта;
* include (из модуля conf.urls) – функция, которая принимает полный путь импорта Python к другому модулю URLconf, которые должны быть «включены» в этом месте.

Для подключения этих модулей используются инструкции import и from. Отличия заключается в том, что инструкция import импортирует весь модуль целиком, таким образом могут подключиться лишние атрибуты модуля, которые не будут использованы в программе, а инструкция from позволяет подключить определенные атрибуты модуля.

При программировании сайта с помощью Python и Django использовались обе эти инструкции. Например, чтобы импортировать атрибут модуля url необходимо выполнить следующую инструкцию: from django.conf.urls import url. А если необходимо импортировать весь модуль, следует выполнить инструкцию: import os.

## 2.3 Файловая структура веб-сайта

Файловая структура сайта состоит из HTML-, CSS-, JavaScrip- и Python-файлов.

Так как был использован фреймворк Django, то в корневом каталоге homeland располагаются приложения отдельных модулей сайта.

Приложения создаются через командную строку. При их создании внутри приложения создаются следующие каталоги и файлы:

* каталог \_\_pycache\_\_ – каталог, содержащий скомпилированные Python-файлы, находящиеся в каталоге с приложением; они значительно ускоряют работу сайта, так как нет необходимости заново компилировать исходный код Python в код «понятный» компьютеру;
* каталог migrations – использует миграции для переноса изменений в моделях, такие как добавление поля, удаление модели и так далее, на структуру БД;
* файл \_\_init\_.py – необходим для того, чтобы Python трактовал эти каталоги как содержащие пакеты. Это сделано во избежание нечаянного сокрытия правомерных модулей, встречающихся в дальнейшем по пути поиска, каталогами с часто используемыми именами;
* файл tests.py – используется для автоматического тестирования отдельных модулей;
* файл admin.py – файл для настройки административной страницы. Используется для добавления новых компонентов на страницу, программной настройки их отображения и так далее;
* файл models.py – позволяет создавать таблицы в БД. Для этого необходимо создать новый класс и прописать необходимые поля. Так же используя встроенный класс Meta можно изменить название таблицы в базе данных и произвести другие манипуляции с таблицей;
* файл urls.py – позволяет создаватьURL-адреса. При создании URL-адреса в приложении создается подуровень URL-адреса, перенаправленного с основного файла urls.py;
* файл views.py – здесь располагаются все функции обработки какого-либо модуля, после чего результат отправляется в шаблон Django. [8]

Рассмотрим более подробно корневую папку сайта:

* каталог «feedback» − приложение для отправки отзывов на страницу «Отзывы»;
* каталог «news» – приложение для добавления новостей на страницу «Новости»;
* каталог «index\_page» – приложения для добавления информации об организации, изменения контактной информации, добавления сотрудников, отправки вопроса через форму «Связь с нами»;
* каталог «static» – каталог для хранения CSS-, JS-файлов, логотипов и кнопок;
* каталог «media» – каталог для хранения фото- и видеоматериалов;
* каталог «templates» – каталог для хранения HTML-страниц сайта;
* каталог «homeland» – приложение с настройками сайта, URL- адресами и взаимодействием между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером;
* файл db – файловая БД.

Файловая структура корневого каталога изображена на рисунке 5.

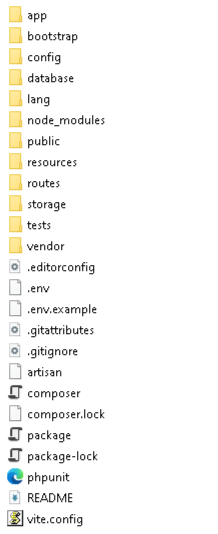


Рисунок 5 – Файловая структура корневого каталога

Рассмотрим файловую структуру шаблона веб-сайта:

Файловая структура сайта представлена на рисунке 6.

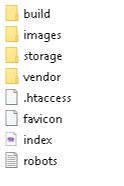


Рисунок 6 – Файловая структура веб-сайта

## 2.4 Работа с веб-сайтом

Сайт располагается на домене adigeyacsm.ru. Для того чтобы войти на административную страницу сайта интернет-магазина запчастей, необходимо перейти по адресу https://adigeyacsm.ru/admin и ввести свои учетные данные. После чего будет доступна административная панель сайта, где администратор сможет редактировать, удалять и добавлять необходимую ему информацию. Страница административного режима представлена на рисунке 7.

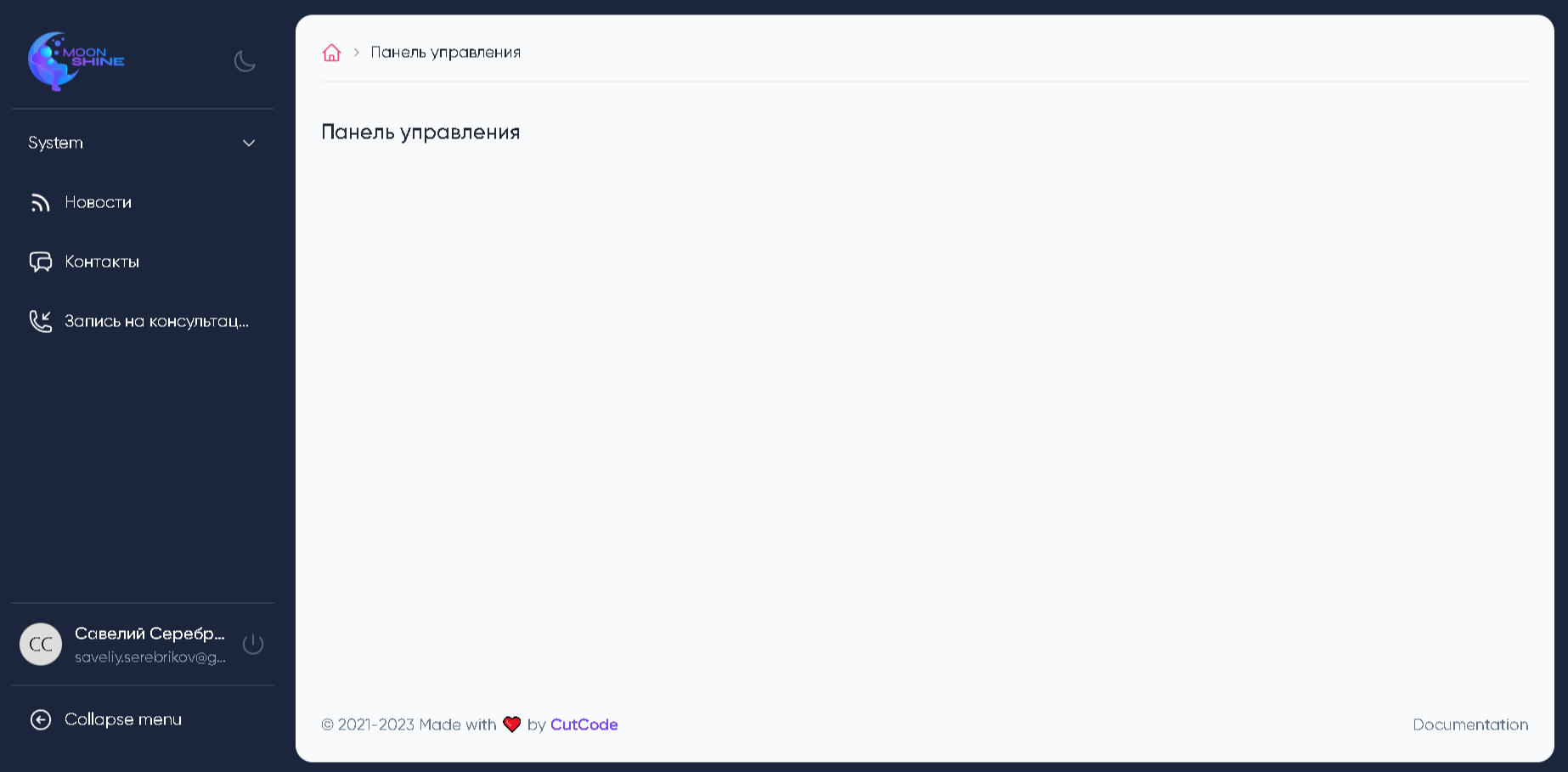


Рисунок 7 – Страница административного режима

Для того чтобы войти на страницу сайта «Адыгейский ЦСМ», необходимо перейти по адресу https://adigeyacsm.ru.

В верхней части страницы расположено меню, с помощью которого можно переходить по страница (рисунок 8).

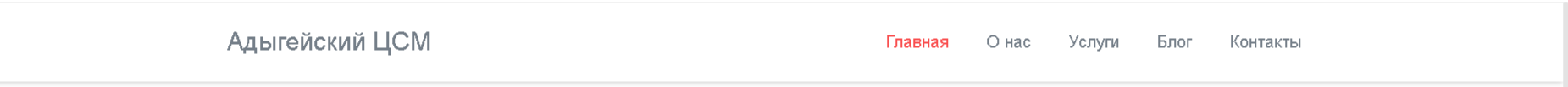


Рисунок 8 – Меню сайта

На странице «О нас» отображается краткая информация об организации (рисунки 9-10).



Рисунок 9 – О нас

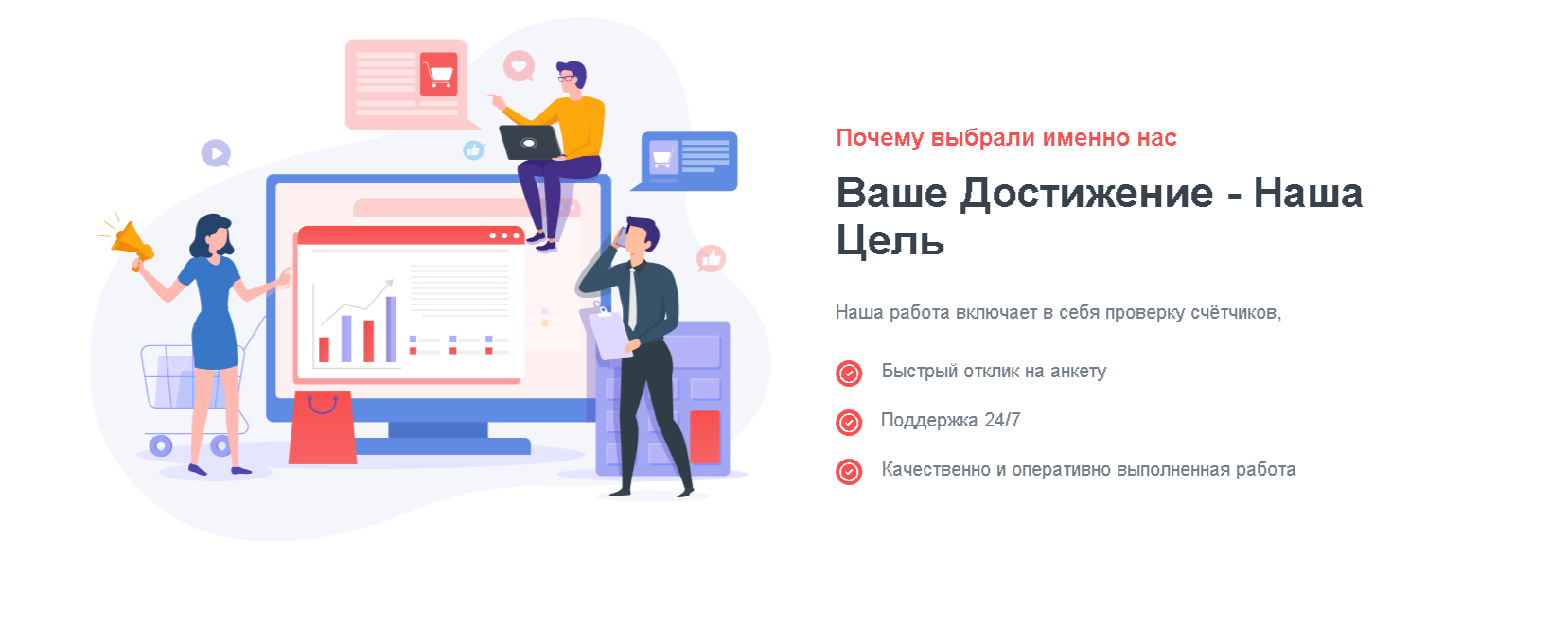


Рисунок 10 – Наши достижения

На странице «Услуг» отображаются список всех услуг, которые выполняет организация. (рисунок 11).

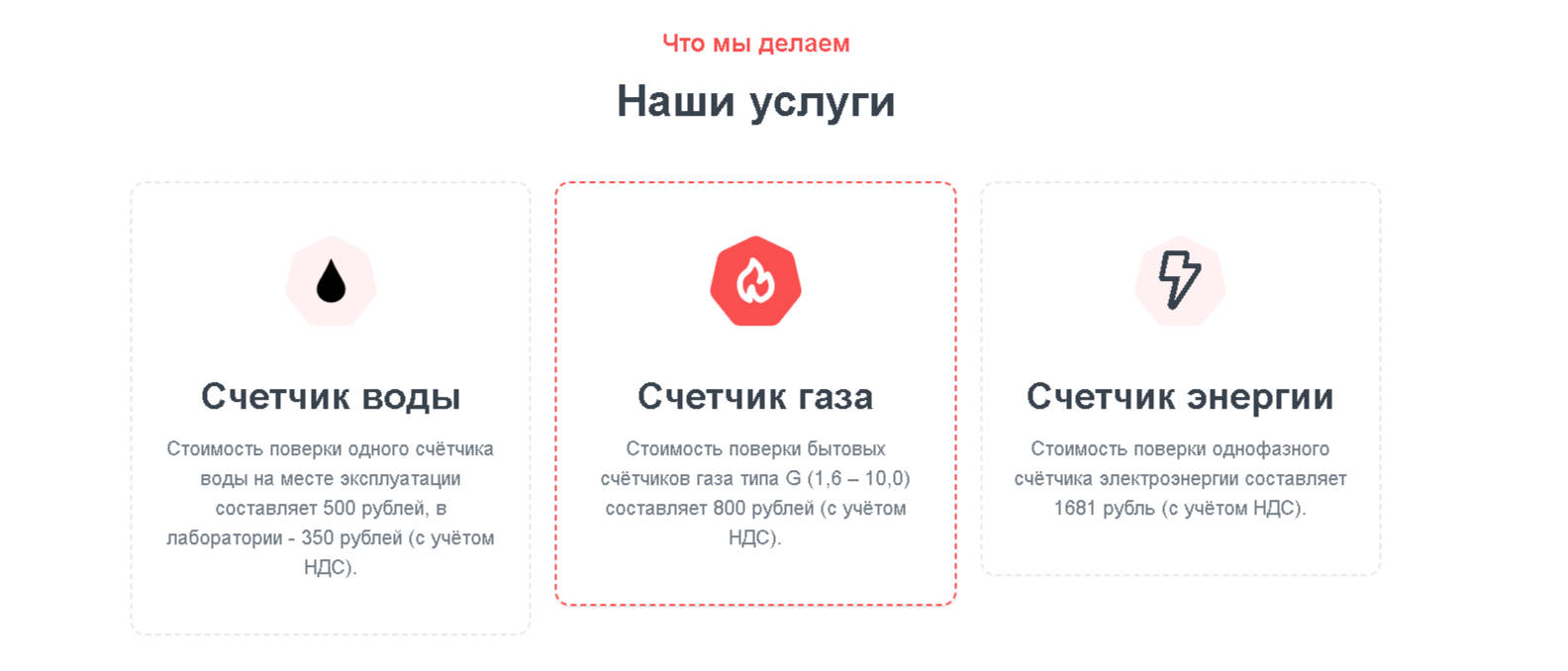


Рисунок 11 – Наши услуги

В самом внизу главной страницы будет располагаться форма обратной связи (рисунок 12).

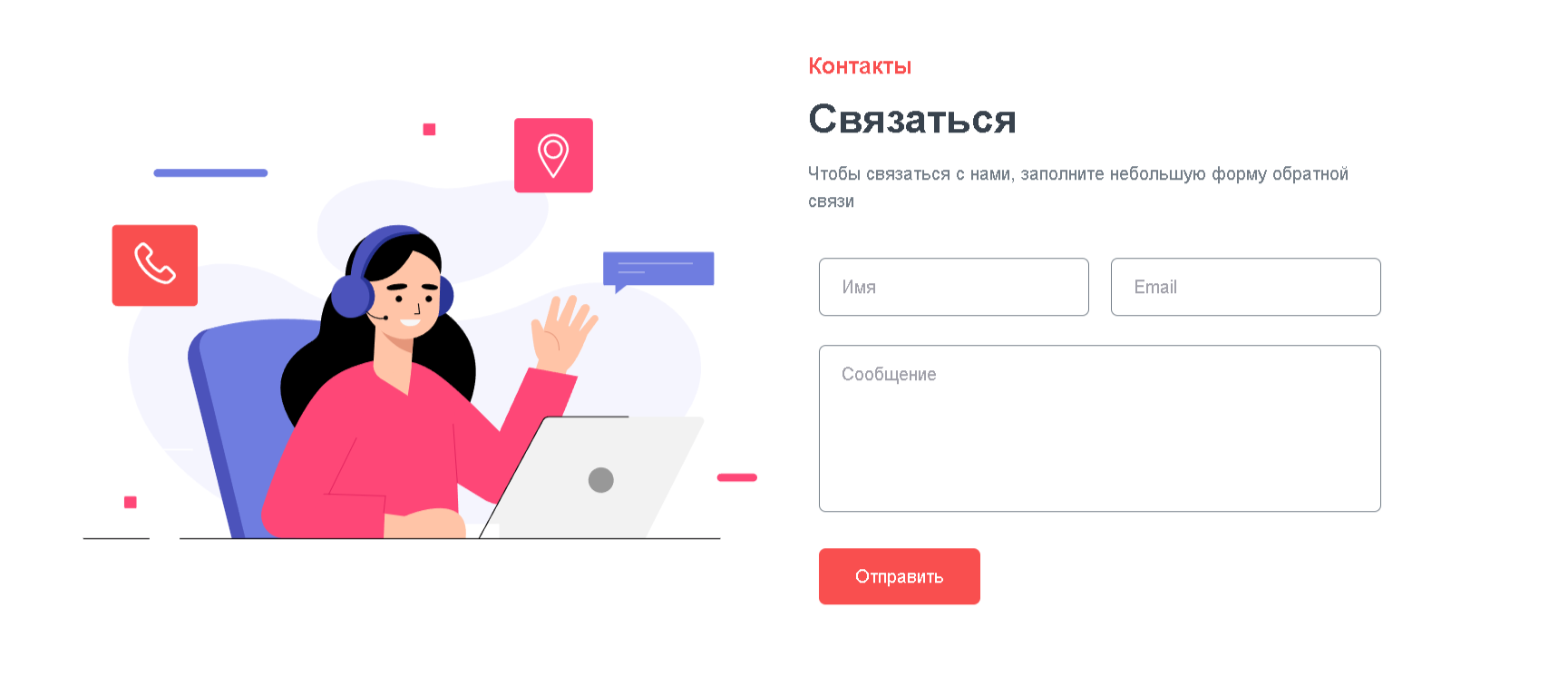


Рисунок 12 – Форма обратной связи

Для получения подробной контактной информации организации присутствует страница «Контакты» (рисунок 13).

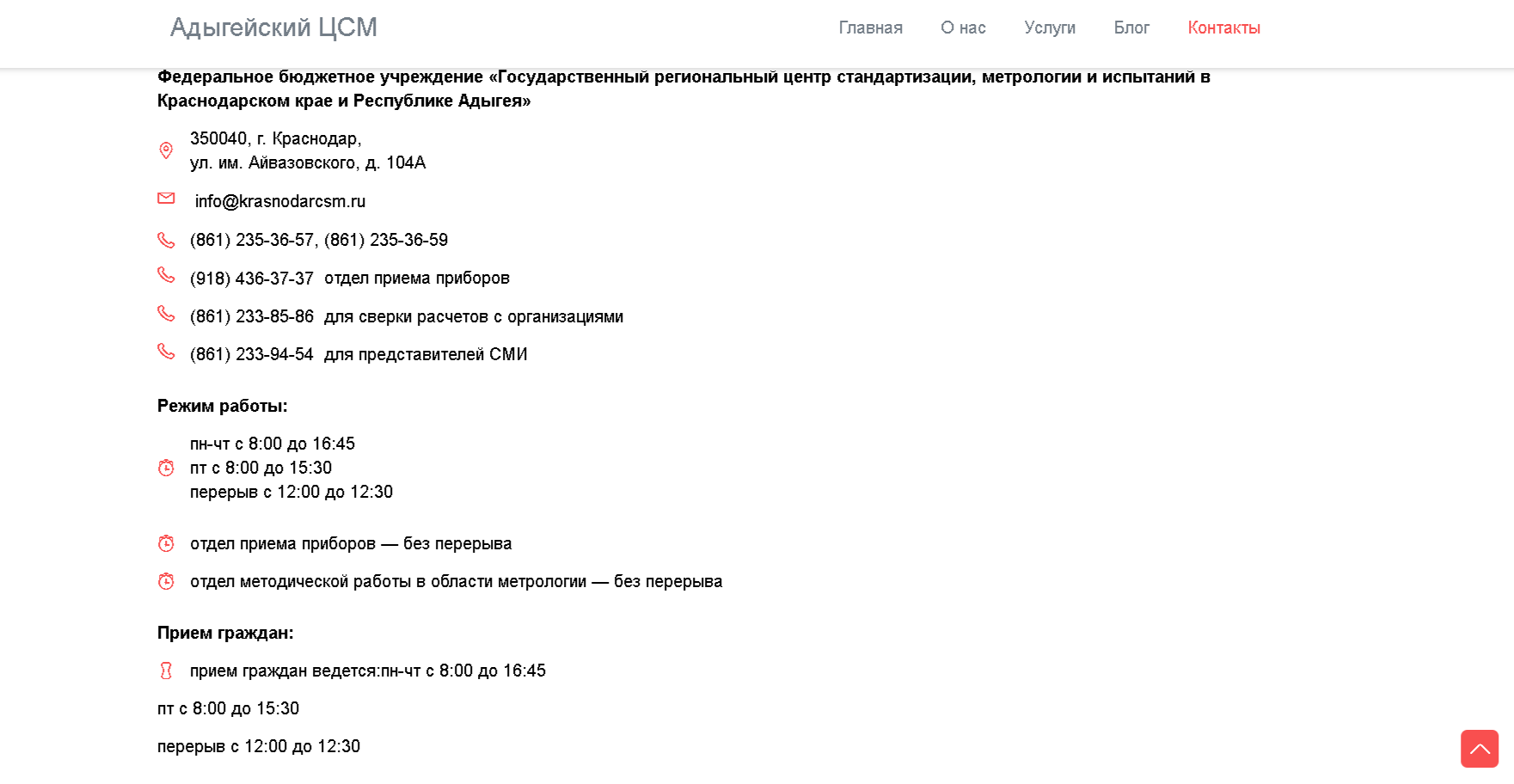


Рисунок 13 – Контакты

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы на тему «Разработка сайта организации ЦСМ Адыгейский филиал» был проведен многосторонний анализ, планирование и реализация проекта, направленного на создание современного и функционального веб-ресурса. В этом процессе акцентировалось внимание на современных технологических требованиях, пользовательских потребностях и безопасности.

Данный сайт ориентирован на пользователей любой возрастной группы, желающие узнать более подробную информацию об организации «ЦСМ» Адыгейский филиал либо записаться на прием.

Исходя из основных принципов веб-дизайна и разработки, была разработана концепция сайта, включающая в себя пользовательский интерфейс, структуру информации, навигацию и функциональные возможности. Инструменты и технологии, такие как HTML, CSS, JavaScript, а также современные фреймворки, были применены для создания адаптивного и отзывчивого веб-ресурса, который оптимизирован для различных устройств и браузеров.

Особое внимание уделялось вопросам безопасности: были применены методы защиты от вредоносных атак, утечек данных и других потенциальных угроз, что стало важным фактором успешной реализации проекта.

Кроме того, проект включал в себя этапы тестирования, оптимизации и анализа пользовательского опыта, что позволило улучшить функциональность, удобство использования и эффективность сайта.

В заключении можно подчеркнуть, что данная курсовая работа стала не только практическим применением теоретических знаний, полученных в процессе обучения, но и ценным опытом в области веб-разработки.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. [«Отзывчивый дизайн на HTML5 и CSS3 для любых устройств»](https://codelibrary.info/books/html/otzyvchivyj-dizajn-na-html5-i-css3)// Бен Фрэйн, 2022. 336 с.
2. Албахари Б., Албахари Дж. С# 7.0. Справочник. Полное описание языка. М.: Вильямс, 2019. 1024 с.
3. [«Шпаргалки для начинающего верстальщика HTML/CSS»](https://codelibrary.info/books/html/shpargalki-dlya-verstalshchika)// Елена Эберт, 2021. 103с.
4. [«Bootstrap и CSS-препроцессор Sass. Самое необходимое»](https://codelibrary.info/books/css/bootstrap-i-css-preprotsessor-sass)// Николай Прохоренок, 2021. 497 с.
5. [«HTML5 для профессионалов»](https://codelibrary.info/books/html/html5-dlya-professionalov)// Питер Лабберс
6. [«CSS для профи»](https://codelibrary.info/books/css/css-dlya-profi)// Кит Грант, 2019, 498 с.
7. [«PHP и MySQL. 25 уроков для начинающих»](https://codelibrary.info/books/php/php-i-mysql-25-urokov-dlya-nachinayushchikh)// Владимир Дронов с, 2021. 434.
8. [«Самоучитель PHP 7 + код»](https://codelibrary.info/books/php/samouchitel-php-7)// Максим Кузнецов, Игорь Симдянов, 2019, 450.
9. [«HTML5 + CSS3. Основы современного WEB-дизайна»](https://codelibrary.info/books/html/html5-css3-osnovy-sovremennogo-web-dizajna)// Кириченко А.В., Хрусталев А.А, 2021, 354.
10. [«Современный CSS с Tailwind»](https://codelibrary.info/books/css/sovremennyj-css-s-tailwind)// Ноэль Раппин (Noel Rappin)
11. [«Создаем динамические веб-сайты на PHP»](https://codelibrary.info/books/php/sozdaem-dinamicheskie-web-sajty-na-php)// Кевин Татро, Питер Макинтайр, 2021 ,544.
12. [«JavaScript. Дополнительные уроки для начинающих»](https://codelibrary.info/books/javascript/javascript-dopolnitelnye-uroki)// Владимир Дронов, 2023 ,354.

# **ПРИЛОЖЕНИЕ А (справочное)**

**Листинг кода для добавления отзывов**

# Файл models.py для создания таблицы «Отзыв о сайте или кружке» (Feedback)

from django.db import models

from imagekit.processors import ResizeToFill

from imagekit.models import ProcessedImageField

class Feedback(models.Model):

class Meta:

db\_table = 'feedback'

verbose\_name = 'Отзывы'

verbose\_name\_plural = 'Отзывы'

name = models.CharField(max\_length=150, verbose\_name='ФИО')

email = models.EmailField(verbose\_name='Email')

photo = ProcessedImageField(upload\_to='feedback',

processors=[ResizeToFill(80, 80)],

options={'quality': 100})

message = models.TextField(verbose\_name='Комментарий')

date = models.DateTimeField(auto\_now\_add=True, verbose\_name='Время и дата')

def \_\_str\_\_(self):

return self.name

# Файл views.py для обработки полученных данных с формы и передача их в шаблон HTML

from django.shortcuts import render

from .models import Feedback

from .forms import FormFeedback

from django.http import HttpResponseRedirect

def add\_feedback(request):

enable = '0'

if 'pause' in request.session:

enable = '2'

request.session.flush()

if request.method == 'POST':

form = FormFeedback(request.POST, request.FILES)

if form.is\_valid():

form.save()

request.session.set\_expiry(60)

request.session['pause'] = True

return HttpResponseRedirect('/feedback/')

else:

return render(request, 'feedback/add\_feedback.html',

{'form': form,

'add\_comments': Feedback.objects.all().order\_by('-id'),

'errors': '1',

})

else:

form = FormFeedback()

return render(request, 'feedback/add\_feedback.html',

{'form': form,

'add\_comments': Feedback.objects.all().order\_by('-id'),

'errors': enable,

})

# **ПРИЛОЖЕНИЕ Б (справочное)**

**Листинг кода для добавления новостей**

# Файл models.py для создания таблицы «Новость» (News)

from django.db import models

from imagekit.processors import ResizeToFill

from imagekit.models import ProcessedImageField

class Add\_news(models.Model):

class Meta:

db\_table = 'news'

verbose\_name = 'Новости'

verbose\_name\_plural = 'Новости'

title = models.CharField(max\_length=150, verbose\_name='Заголовок')

date = models.DateField(verbose\_name='Дата')

text = models.TextField(verbose\_name='Текст новости')

def \_\_str\_\_(self):

return self.title

# Файл views.py для передачи новости в шаблон HTML

from django.shortcuts import render

from .models import Add\_news

def add\_news(request):

return render(request, 'news/add\_news.html', {

'all\_news': Add\_news.objects.all().order\_by('-id'),

})

def single\_news(request, article\_id):

return render(request, 'news/show\_single\_news.html', {

'single\_news': Add\_news.objects.get(pk = article\_id)

})