МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ   
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования «Алтайский государственный университет»

Колледж АлтГУ

Отделение экономики и информационных технологий

**ОТЧЕТ**

по производственной (преддипломной) практике

Выполнил студент  
3 курса, К.105с11-3 группы

Малимонов Сергей Дмитриевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Руководитель:

Высшая к.к

Петракова Ольга Викторовна

Работа защищена:

"\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Барнаул 2024 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc166787461)

[1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ВЕБ-САЙТА 5](#_Toc166787462)

[1.1 Характеристика организации 5](#_Toc166787463)

[1.2 Веб-сайт: основные понятия, frontend и backend. 6](#_Toc166787464)

[1.3 PHP 10](#_Toc166787465)

[1.3.1 История создания PHP 10](#_Toc166787466)

[1.3.2 Основные аспекты и принцип работы PHP 11](#_Toc166787467)

[1.4 Архитектурный паттерн MVC 14](#_Toc166787468)

[2 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ВЕБ-САЙТА КРАЕВОГО ГОСУДАРСТВЕННОГО КАЗЕННОГО УЧРЕЖДЕНИЯ 17](#_Toc166787469)

[2.1 Настройка среды разработки 17](#_Toc166787470)

[2.2 Разработка макета 18](#_Toc166787471)

[2.3 Разработка веб-сайта 22](#_Toc166787472)

[2.3.1 Настройка веб-сервера Apache и реализация роутинга 22](#_Toc166787473)

[2.3.2 Верстка страниц 24](#_Toc166787474)

[2.3.3 Создание и внедрение базы данных в разработку 27](#_Toc166787475)

[2.3.4 Адаптация сайта под мобильные устройства 30](#_Toc166787476)

[ЗАКЛЮЧЕНИЕ 33](#_Toc166787477)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34](#_Toc166787478)

[ПРИЛОЖЕНИЕ А 36](#_Toc166787479)

[ПРИЛОЖЕНИЕ Б 37](#_Toc166787480)

[ПРИЛОЖЕНИЕ В 38](#_Toc166787481)

[ПРИЛОЖЕНИЕ Г 39](#_Toc166787482)

[ПРИЛОЖЕНИЕ Д 41](#_Toc166787483)

# ВВЕДЕНИЕ

Интернет в деловой и повседневной жизни человека стал столь же привычным, как телефон. Бесчисленное множество новых технологий прочно входит в наш быт, изменяя наш образ мысли и деятельности. Сегодня Интернет содержит миллионы сайтов, на которых размещены все виды информации. Люди получают доступ к этой информации с помощью интернет-технологий.

Сейчас сложно представить организацию без собственного веб-сайта. В первую очередь, сайт организации несет в себе информационный характер. Пользователь, зайдя на сайт, получит подробную информацию об организации. Так же, веб-сайт может служить платформой для обратной связи с клиентами, например, через онлайн-формы, чаты или электронную почту. Это позволяет оперативно реагировать на запросы и обращения, улучшая качество обслуживания.

Разработка сайта для краевого государственного казенного учреждения «Управление социальной защиты населения по Шелаболихинскому району» необходима, потому что в век информационных технологий у каждой организации должен быть свой сайт, на котором можно было бы узнать больше информации о ней.

Производственная (преддипломная) практика проходила в краевом государственном казенном учреждении «Управление социальной защиты населения по Шелаболихинскому району» с 18.04.2024 по 15.05.2024, в должности программиста.

Объектом исследования являются информационные технологии построения веб-сайта.

Предметом исследования является веб-сайт краевого государственного казенного учреждения «Управление социальной защиты населения по Шелаболихинскому району»

Основной целью прохождения производственной (преддипломной) практики является разработка веб-сайта для краевого государственного казенного учреждения «Управление социальной защиты населения по Шелаболихинскому району»

В ходе прохождения производственной (преддипломной) практики необходимо решить следующие задачи:

* изучить соответствующую литературу;
* исследовать методы разработки веб-сайтов;
* разработать макет веб-сайта и сверстать страницы по разработанному макету;
* реализовать фронтенд и бэкенд части веб-сайта;
* провести ряд тестов веб-сайта.

Отчет по производственной (преддипломной) практики состоит из введения, двух глав, заключения, списка использованных источников.

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ВЕБ-САЙТА

## Характеристика организации

Краевое государственное казенное учреждение «Управление социальной защиты населения по Шелаболихинскому району» начало работу 24 декабря 2007 года. Главной целью деятельности Управления социальной защиты населения по Шелаболихинскому району как самостоятельного структурного подразделения "Главалтайсоцзащиты" является реализация государственной политики в сфере социальной поддержки населения по Шелаболихинскому району.

Основные направления деятельности:

* государственная поддержка малоимущих граждан, семей, женщин, детей, пожилых граждан и инвалидов;
* организация социальных выплат населению.

Цели и задачи управления:

* социальная поддержка граждан в виде предоставления субсидии на оплату жилья и коммунальных услуг;
* осуществление доплаты к пенсиям государственным служащим и лицам, имеющим особые заслуги перед Российской Федерацией и Алтайским краем, финансируемой за счет средств краевого бюджета;
* организация работы по обеспечению соблюдения органами социальной защиты населения муниципальных образований Алтайского края законодательства о компенсационных выплатах гражданам, пострадавшим от воздействия радиации;
* осуществление государственной поддержки слабо защищенных групп населения, граждан, находящихся в трудной жизненной ситуации, семьи, женщин, детей, пожилых граждан и инвалидов в формах и порядке, установленных законодательством Российской Федерации, Алтайского края;
* организационно-методическое и правовое обеспечение органов социальной защиты населения муниципальных образований Алтайского края в области назначения, перерасчета и выплаты государственных пособий, субсидий, других социальных выплат населению.

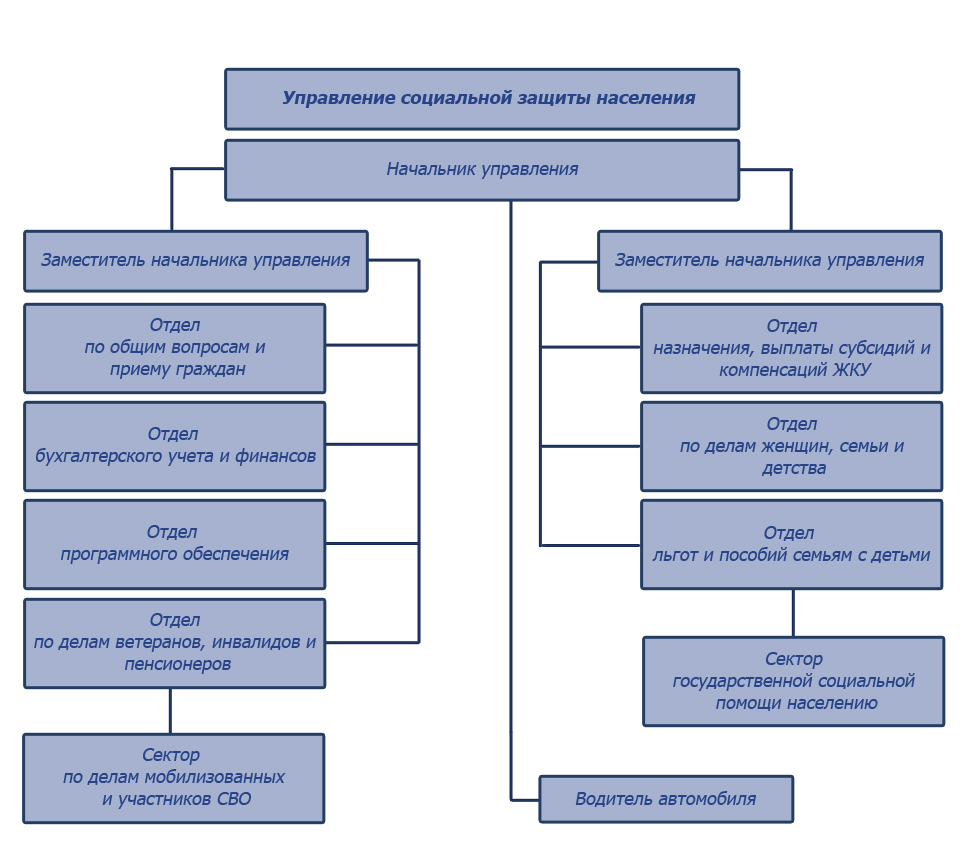
Структуру краевого государственного казенного учреждения «Управление социальной защиты населения по Шелаболихинскому району» можно увидеть на рисунке 1.

Рисунок 1 – Структура организации

1.2 Веб-сайт: основные понятия, frontend и backend.

Сайт – это интернет-ресурс, состоящий из одной, нескольких или множества виртуальных страниц. Все страницы связаны между собой ссылками и обычно объединены общей темой или задачей. По действующему законодательству, у сайта обязательно есть владелец: веб-ресурс зарегистрирован на физическое либо юридическое лицо.[11]

Компании рассказывают о себе, делятся контактами, собирают отзывы и продают свои товары и услуги. По сайту зачастую судят о фирме: удобный, информативный и «живой» сайт повышает градус доверия в глазах потенциального партнера или клиента.

Частные лица с помощью сайтов рассказывают о себе потенциальным работодателям или клиентам (если владелец сайта – предприниматель или фрилансер); ищут единомышленников для общения; делятся опытом и творчеством, чтобы получить обратную связь от аудитории.[15]

Существует множество различных типов веб-сайтов, каждый из которых имеет свою уникальную цель и функциональность. Можно выделить следующие:

1. Лендинг (Landing Page). Обычно содержит одну страницу с яркими элементами дизайна, кратким описанием предложения, призывом к действию и контактной информацией. Целью лендинга является привлечение внимания к определенному продукту, услуге или событию;
2. Корпоративный сайт. Обычно содержит несколько страниц, таких как "О нас", "Продукты/услуги", "Контакты", "Блог" и т.д. Дизайн оформлен в корпоративном стиле компании. Цель – представление компании, её продуктов, услуг, истории и контактной информации. Сайт, который будет разрабатываться, можно отнести к корпоративному;
3. Интернет-магазин. Включает в себя каталог товаров с описаниями, фотографиями, ценами и кнопками добавления в корзину. Также содержит функционал оформления заказа, оплаты, доставки и обратной связи Целью интернет – магазина является продажа товаров и услуг через Интернет;
4. Блог. Содержит статьи, заметки, обзоры, рецензии, мнения авторов и комментарии пользователей. Обычно упор делается на качественный и интересный контент Цель блога – публикация авторского контента, обмен мнениями и информацией;
5. Портфолио. Содержит портфолио автора с описанием выполненных работ, изображениями, ссылками на проекты и контактной информацией. Обычно упор делается на визуальное представление. Цель у такого сайта – демонстрация работ, проектов, навыков.[12]

У каждого сайта есть свое доменное имя. Это уникальное имя, используемое для идентификации веб-сайта в сети Интернет. Доменное имя состоит из:

1. домен нулевого уровня – точка, которая не отображается в строке браузера;
2. домен первого уровня – доменная зона. Например, национальные домены: .ru, .рф, .eu и другие;
3. домен второго уровня  – поддомен, который никогда не повторяется, та часть домена, находящаяся слева от точки, например, yandex.ru, где yandex – домен второго уровня;
4. домен третьего уровня – используется, чтобы создать более понятную структуру и выделить определенный блок на сайте, например, maps.yandex.ru, где maps – домен третьего уровня.[14]

Каждая страница интернет–ресурса содержит текст и/или изображения (видео). Содержимое преобразуется в веб-страницу с помощью языков разметки, стилей и/или программирования: HTML, CSS, JavaScript, PHP.

Говоря про веб-сайт, нельзя обойти стороной HTML – HyperText Markup Language и CSS – Cascading Style Sheets. HTML – стандартный язык разметки, который используется для создания структуры и содержания веб-страниц. Он состоит из набора тегов и атрибутов, которые определяют различные элементы на веб-странице. Иными словами, HTML создает иерархическую структуру веб-страницы, используя заголовки, абзацы, списки и таблицы.[5]

CSS – это язык стилей, который используется для оформления внешнего вида веб-страниц. CSS позволяет задавать различные стили для элементов HTML, делая веб-сайты более привлекательными и функциональными. CSS является мощным инструментом для стилизации веб-страниц и обеспечивает широкие возможности для создания красивого и функционального внешнего вида веб-сайтов.[1]

В целом, то, что видит пользователь, с чем он взаимодействует, называется frontend частью веб-сайта. Картинки, кнопки, анимации – примеры frontend части веб-сайта. Из этого следует, что HTML и CSS относятся к frontend. Но зачастую, одного HTML и CSS недостаточно. Дело в том, что ограничиваясь лишь этими средствами, не получится создать по-настоящему полноценный веб-сайт. На выходе получится статичный документ. Для того, чтобы «оживить» сайт, сделать из статичного документа динамичный веб-ресурс, используют язык программирования JavaScript.[2]

JavaScript является высокоуровневым интерпретируемым языком программирования, который применяется для создания динамических и интерактивных элементов на веб-страницах.[8] JavaScript расширяет возможности HTML и CSS, добавляя возможности для создания сложного поведения и обработки событий на веб-страницах. JavaScript позволяет привязывать обработчики событий к различным элементам на веб-странице, таким как клики мыши, наведение, отправка формы и другие действия. Это позволяет создавать интерактивные пользовательские интерфейсы и реагировать на действия пользователя. JavaScript является одним из основных инструментов для веб-разработки и играет важную роль в создании динамических и интерактивных веб-сайтов.[6]

Все то, что обрабатывается на стороне сервера и с чем пользователь не взаимодействует, называется backend частью веб-сайта. Отправка данных с форм, запросы к базе данных, вывод записей из базы данных на страницу – примеры backend`a. Для написания backend части веб-сайта можно использовать различные языки, например, Python, Go, Java и тому прочее. Но самым распространенным является PHP. Так как в разработке веб-сайта PHP будет доминирующим языком, поговорим о нем подробнее.[3]

1.3 PHP

1.3.1 История создания PHP

История PHP началась в 1994 году, когда датский программист Расмус Лердорф создал набор скриптов для управления его персональным сайтом. Изначально, эти скрипты были простыми инструментами для отслеживания посетителей и их личной информации. Они также позволяли Расмусу создавать динамические веб-страницы на своем сайте.

Со временем, Лердорф стал расширять функциональность своих скриптов, добавляя новые возможности, такие как работа с базами данных и формами. Он назвал свой проект "Personal Home Page Tools" (сокращенно PHP Tools), поскольку его целью было создание персональных веб-страниц. Постепенно, PHP стал все более популярным среди разработчиков веб-сайтов, и Лердорф решил открыть исходный код своего проекта для общественности.

В 1995 году была выпущена первая версия PHP (PHP/FI), и она быстро набрала популярность благодаря своей простоте и гибкости. В этой версии были добавлены новые возможности, такие как поддержка баз данных mSQL и MySQL, работа с файлами, а также возможность создания динамических HTML-страниц.

Следующие версии PHP были значительно усовершенствованы и расширены. В 1997 году вышла версия PHP 3, которая включала в себя существенные улучшения в языке, такие как поддержка объектно-ориентированного программирования и расширенные возможности для работы с базами данных.

В 2000 году вышла версия PHP 4, которая принесла еще больше новых возможностей, таких как улучшенная производительность, поддержка сессий, новые функции и многое другое. PHP 4 стала самой популярной версией PHP на тот момент и широко использовалась для создания веб-приложений и динамических веб-сайтов.

В 2004 году была выпущена версия PHP 5, которая включила в себя еще более значительные изменения и улучшения. Эта версия внедрила поддержку современных концепций, таких как полноценная поддержка объектно-ориентированного программирования, интерфейсы, пространства имен и многое другое.

Следующие версии PHP (PHP 7 и последующие) продолжили совершенствование языка, добавляя новые возможности, повышая производительность и безопасность, а также улучшая совместимость с современными технологиями.[4]

Актуальная версия PHP на данный момент – 8.3. Именно эта версия будет использоваться для написания веб-сайта.

* + 1. Основные аспекты и принцип работы PHP

Итак, PHP (Hypertext Preprocessor) – это широко используемый интерпретируемый язык программирования, который применяется для создания динамических веб-сайтов и веб-приложений. Он обладает мощными возможностями и широким набором функций для работы с базами данных, файлами, сетью и другими аспектами веб-разработки.

Исследования от 2023 года показывают, что PHP как язык серверного программирования применяется в 77,6% всех веб-сайтов в Интернете. При этом зачастую при их создании используются системы управления контентом (CMS), и многие из них тоже написаны на PHP. Например, Wordpress, на системе которого работают более 60 миллионов сайтов.

PHP является одним из наиболее популярных языков программирования для веб-разработки благодаря своей простоте, гибкости и обширной документации и сообществу разработчиков.

PHP используют практически все крупнейшие компании, выпускающие известные IT-продукты. К ним относятся VK, Wikipedia, Авито и многие другие.

Основные аспекты языка PHP:

1. Синтаксис и структура. PHP имеет синтаксис, сходный с языками программирования вроде C, что делает его относительно легким в освоении для разработчиков. Программы на PHP можно встраивать в HTML-код в виде специальных инструкций, которые начинаются с открывающего тега <?php и закрываются тегом ?>;
2. Встраивание в HTML. PHP предоставляет возможность встраивать код PHP непосредственно в HTML-страницы, что позволяет динамически генерировать контент на основе условий, данных из базы данных и других переменных;
3. Обработка форм. PHP широко используется для обработки данных, полученных из форм на веб-страницах. Он может получать данные из форм, проверять их на корректность, сохранять в базу данных и отправлять по электронной почте;
4. Работа с базами данных. PHP обеспечивает поддержку различных систем управления базами данных (СУБД), таких как MySQL, PostgreSQL, SQLite и другие. С помощью PHP можно выполнять запросы к базе данных, добавлять, изменять и удалять записи, а также обрабатывать полученные данные;
5. Создание сессий и куки. PHP предоставляет механизмы для работы с сессиями и куками (cookies), что позволяет создавать персонализированные и безопасные веб-сайты. Сессии позволяют хранить данные между запросами от одного пользователя, а куки - сохранять информацию на компьютере пользователя;
6. Генерация динамического контента. PHP позволяет генерировать динамический контент на основе различных факторов, таких как пользовательские запросы, данные из базы данных, параметры URL и другие;
7. Расширяемость. PHP поддерживает концепцию модульности и обеспечивает механизмы для расширения его функциональности с помощью подключаемых модулей (расширений) и библиотек.[13]

CSS и НТМL файлы, как и файлы изображений, сервер отправляет сразу клиенту (браузеру), независимо от содержания. В отличие от них PHP-файлы содержат код, который интерпретируется на сервере. Клиент (браузер) получает не сам PHP-код, а результат его исполнения, который, как правило, является чистым CSS и HTML. Пользователь не может увидеть исходный код программы.

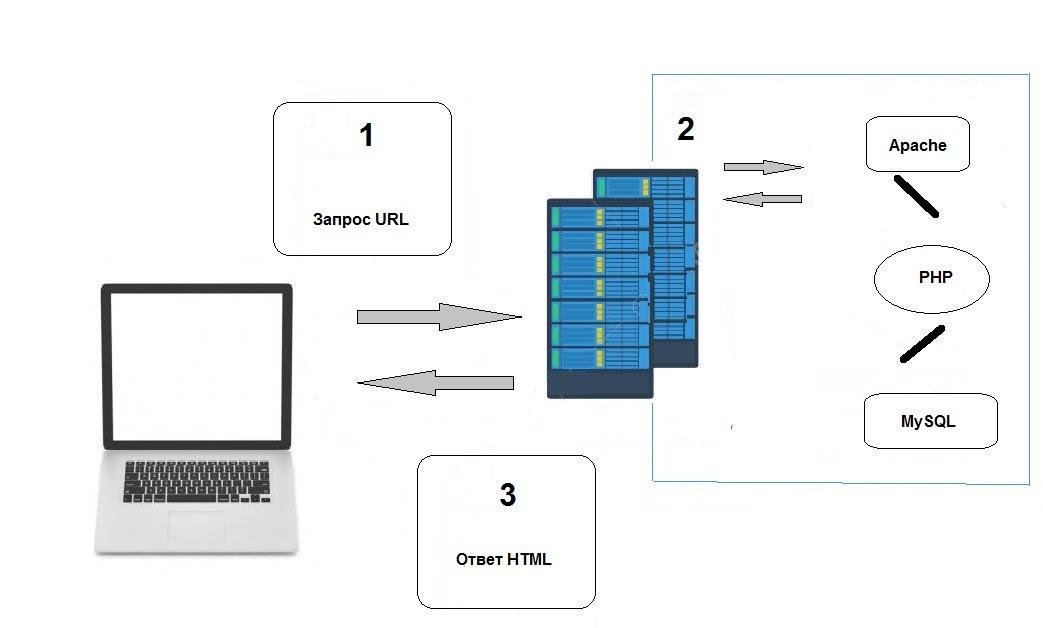
PHP хранит состояние, созданное в текущий момент времени. В отличие от классических компилируемых языков программирования, которые работают все время от запуска до остановки программы, PHP-скрипт начинает выполняться с момента обращения веб-сервера и заканчивает (забывает все, что знал), когда возвращает сгенерированный пакет данных обратно веб-серверу. Схема взаимодействия PHP с веб-сервером представлена на рисунке 2.[7]

Рисунок 2 – Схема взаимодействия PHP с веб-сервером

1. Входящее соединение принимает и анализирует веб-сервер. Происходит интерпретация и формируется ответ;
2. Как правило, результат работы PHP-интерпретатора (программы, выполняющей PHP-код) – HTML-код. На этом этапе PHP-скрипт преобразуется в статический HTML-код, который может читать браузер. В процессе работы PHP может использовать MySQL – свободную систему управления базами данных (БД). Она выполняет обмен информацией с сервером базы данных на согласованном языке (SQL). Сервер базы данных и веб-сервер образуют тандем, читающий и записывающий информацию и данные, предоставляя их посетителям сайта;
3. БД отправляет данные PHP-скрипту, который расставляет их по нужным местам страницы или пакета данных и формирует окончательный ответ. Ответ отправляется обратно веб-серверу, который и возвращает клиенту страницу или пакет данных.

На сегодняшний день PHP остается одним из самых популярных языков программирования для веб-разработки, и его экосистема продолжает активно развиваться и расширяться благодаря активному сообществу разработчиков и различным проектам с открытым исходным кодом.[9]

Стоит отметить, что на PHP написаны Laravel и Symphony – самые популярные фреймворки в современной веб-разработке, которые зарекомендовали себя как надежные платформы для масштабируемых веб-приложений. Эти фреймворки базируются на паттерне проектирования MVC(Model-View-Controller). При разработке сайта будет написан небольшой фреймворк, использующий паттерн MVC.

1.4 Архитектурный паттерн MVC

Концепцию MVC предложил в конце 1970-х годов сотрудник Xerox Трюгве Реекскауг. Она была реализована в языке программирования Smalltalk-80. Окончательную версию шаблона опубликовали только 10 лет спустя в журнале Technology Object. Концепция стала популярна с появлением быстро развертываемых фреймворков и интерактивных веб-приложений. Но что представляет из себя MVC?

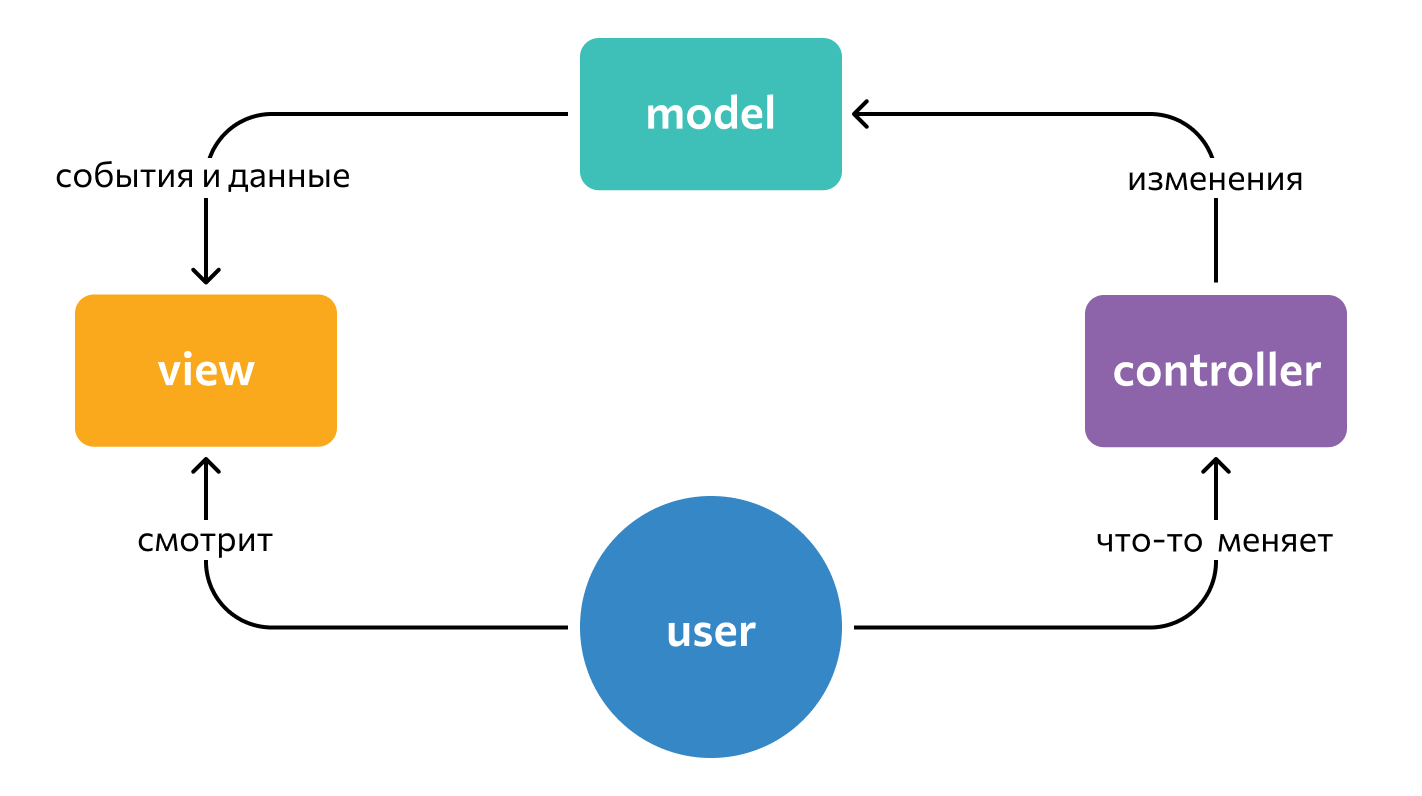
Паттерн проектирования MVC (Model-View-Controller) – это архитектурный шаблон, который используется для разделения компонентов приложения на три основные части: Model (Модель), View (Представление) и Controller (Контроллер). Этот подход способствует улучшению организации кода, облегчает его сопровождение и расширение, а также повышает его переиспользуемость. Логическая схема MVC представлена на рисунке 3.

Рисунок 3 – Логическая схема MVC

1. Модель (Model). Модель отвечает за управление данными приложения, их обработку и бизнес-логику. Модель представляет собой абстракцию данных, которая может быть использована в приложении. Модель не зависит от других компонентов паттерна MVC и обычно включает в себя классы и методы для доступа к базе данных, взаимодействия с внешними сервисами и выполнения бизнес-логики. Основной принцип модели – ее независимость от представления и контроллера. Она должна быть способной работать независимо от того, как данные будут отображены и каким образом будут обрабатываться пользовательские запросы;
2. Представление (View). Представление отвечает за отображение данных пользователю. Оно представляет собой интерфейс, через который пользователи могут взаимодействовать с приложением. Представление получает данные из модели и форматирует их для отображения. Это может быть HTML-страница, JSON-ответ для API, графический интерфейс и тому прочее. Одним из ключевых принципов представления является отделение логики отображения данных от логики приложения. Представление не должно содержать бизнес-логику или прямого взаимодействия с моделью;
3. Контроллер (Controller). Контроллер отвечает за обработку пользовательских запросов, взаимодействие с моделью для получения необходимых данных и передачу этих данных в представление для отображения. Контроллер принимает входящий запрос от пользователя, определяет, какая операция должна быть выполнена, и вызывает соответствующие методы модели для обработки запроса. Контроллер также определяет, какое представление должно быть отображено пользователю, и передает данные представлению для отображения. Ключевым принципом контроллера является его независимость от специфики представления и модели. Он должен быть способен обрабатывать запросы независимо от того, какие данные будут использованы для отображения и каким образом данные будут обработаны.

Применение паттерна MVC позволяет создавать чистый, гибкий и модульный код, который легко поддерживать и расширять. MVC также способствует повышению переиспользуемости компонентов приложения и облегчает разработку сложных систем. Благодаря отделению логики данных от логики отображения и управления пользовательским взаимодействием, MVC становится мощным инструментом для создания современных веб-приложений.[10]

2 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ВЕБ-САЙТА КРАЕВОГО ГОСУДАРСТВЕННОГО КАЗЕННОГО УЧРЕЖДЕНИЯ

2.1 Настройка среды разработки

Прежде чем начать разработку сайта, нужно настроить среду разработки. Так как было решено использовать язык PHP, который является серверным, нужен локальный сервер, чтобы PHP скрипты могли исполняться. Для разработки отлично подойдет OpenServer – мощная портативная программная среда, созданная специально для веб-разработчиков, включающая в себя тщательно подобранный набор серверного программного обеспечения.

После того, как OpenServer был установлен на компьютер, нужно выбрать необходимые модули для разработки: http-сервер, версию php, предпочтительную базу данных. Выбор пал на веб-сервер Apache 2.4, PHP 8.1 и MySql 8.0, как показано на рисунке 4.

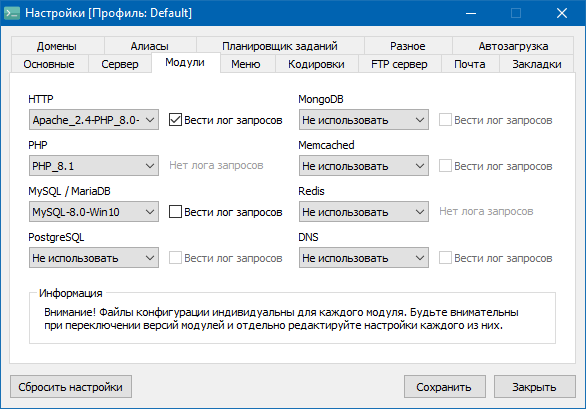


Рисунок 4 – Настройки OpenServer

В качестве редактора кода был выбран Visual Studio Code – удобный редактор для разработки веб-приложений.

Было решено сразу разбить проект на директивы и создать соответствующие папки:

* app – в этой папке находятся все модели, контроллеры и виды;
* config – в этой папке находятся конфигурационные файлы: массив с роутами, данные для подключения к базе данных и константы для удобного подключения файлов;
* public\_html – в этой папке находится front controller, принимающий запросы от пользователей, а также JavaScript файлы, CSS стили, картинки и прочее;
* router – в этой папке находится router;
* vendor – в этой папке находятся написанные функции и класс для работы с базой данных.

2.2 Разработка макета

Разрабатываемый сайт должен состоять из четырех основных страниц: главная страница, страница с новостями, страница с документами и страница с формой обратной связи. Исходя из корпоративного цвета, основная цветовая палитра – синий, белый и холодный белый.

В качестве средства разработки выбор пал на Figma – онлайн-сервис для разработки интерфейсов.

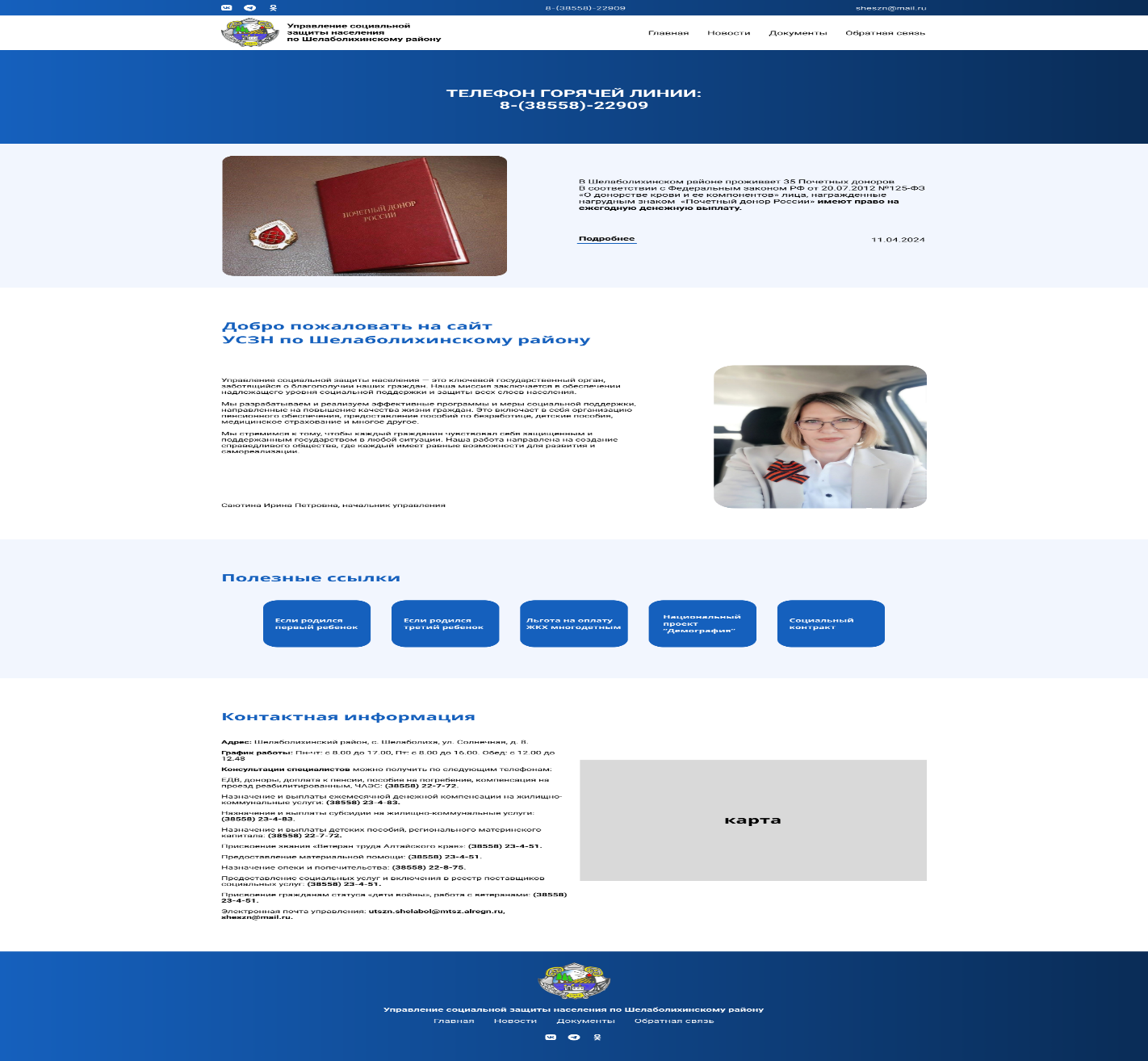
Первым был сделан дизайн главной страницы, как показано на рисунке 5.

Рисунок 5 – Дизайн главной страницы сайта

На главной странице должно быть 5 основных секций: номер телефона, последняя опубликованная новость, приветственное слово от начальника организации, полезные ссылки, ведущие на главный сайт управления по Алтайскому краю и секция с контактной информацией.

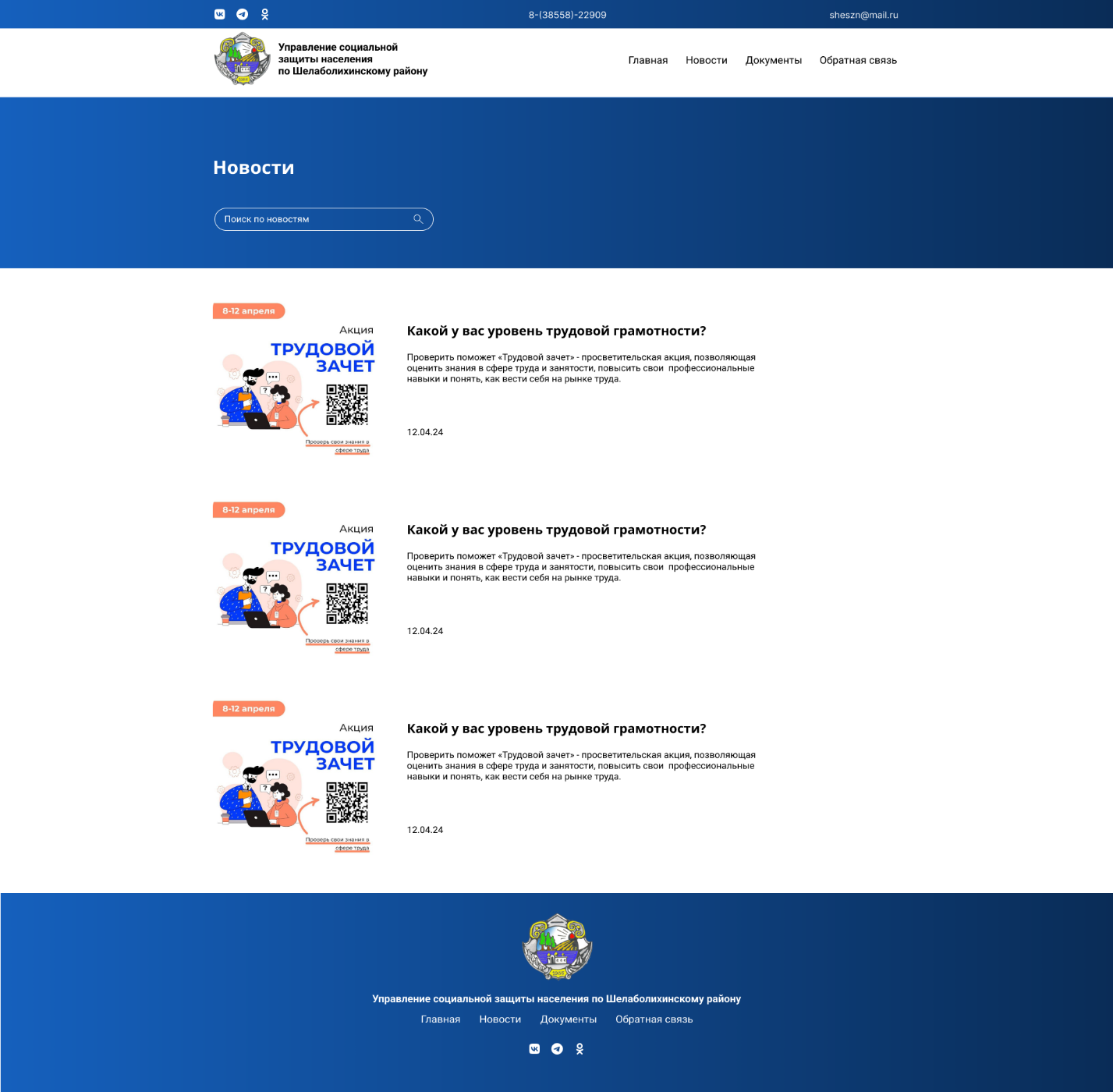
После, был сделан дизайн страницы с новостями, на который выводятся все новости, а также имеется поле для поиска по новостям, как представлено на рисунке 6.

Рисунок 6 – Дизайн страницы с новостями

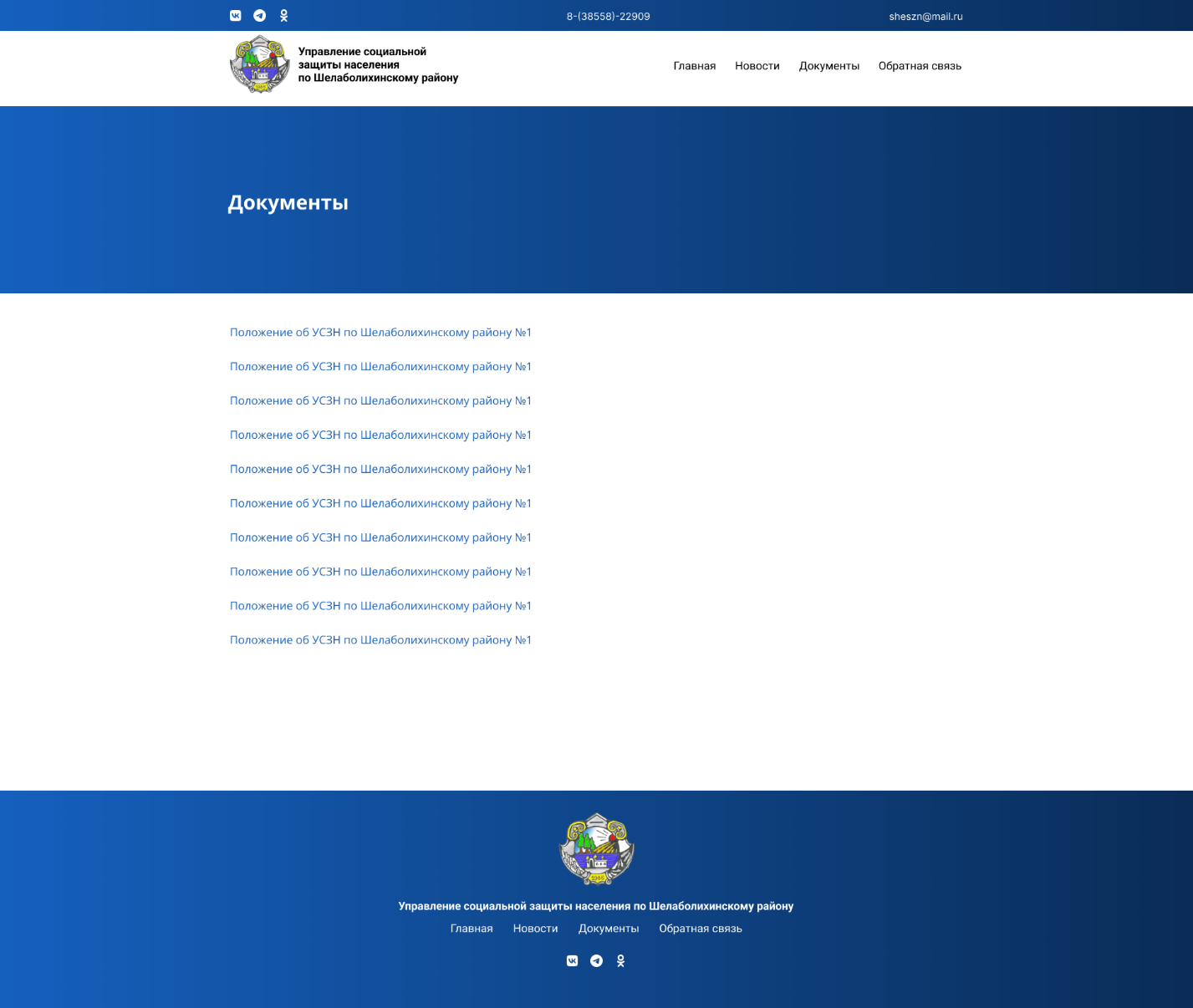
Далее, был сделан дизайн страницы с документами. На этой странице должны выводится все соответствующие организации документы, с которыми может ознакомиться любой желающий, как представлено на рисунке 7.

Рисунок 7 – Дизайн страницы с документами

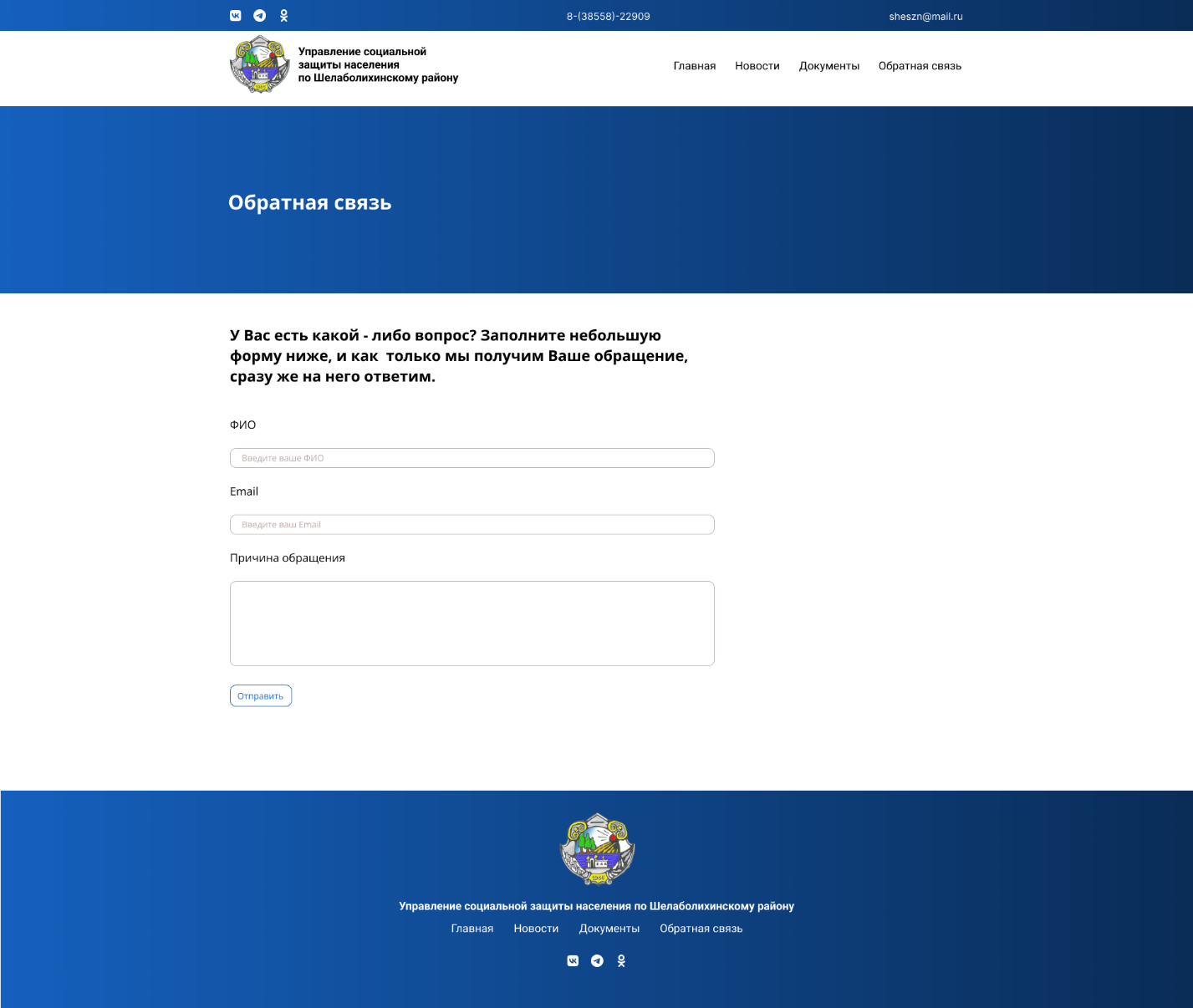
Завершающей страницей стала «обратная связь», на которой находится форма, заполнив которую, пользователь отправит свое обращение в базу данных, как показано на рисунке 8.

Рисунок 8 – Дизайн страницы обратной связи

После того, как макет полностью завершен, можно приступать к разработке.

2.3 Разработка веб-сайта

2.3.1 Настройка веб-сервера Apache и реализация роутинга

Начав разработку, было принято решение использовать так называемую «единую точку входа» - концепцию, при которой все запросы от пользователя приходят на один файл – так называемый «front controller» и уже непосредственно в нем обрабатываются, и в зависимости от запроса отображается нужная страница, либо же, страница с ошибкой.

Чтобы реализовать данную концепцию, необходимо перенастроить конфигурационный файл сервера Apache – .htaccess. В случае разрабатываемого веб-сайта будет два таких файла – один в корне проекта, а другой – в папке public\_html.

Код первого файла находится в приложении А.

Данный код включает необходимый модуль rewrite, а после задает правило перезаписи – перенаправляет все запросы в папку public\_html, сохраняя оригинальный путь с помощью переменной $1.

Код файла, находящегося в папке public\_html находится в приложении Б.

Данный код делает две важные вещи: защищает от того, чтобы пользователь не мог открыть какой либо существующий файл в нашем проекте через URL, а так же перенаправляет все запросы пользователя во фронт-контроллер index.php

Убедившись, что все работает, можно приступать к написанию небольшого роутера для обработки запросов от пользователей.

Принцип работы роутера следующий: в переменную записывается запрос от пользователя, а после идет проверка, существует ли такой запрос, записанный в массиве, и если такой запрос нашелся – подключается соответствующий ему контроллер, или же вызывается написанная функция abort, которая в свою очередь отображает страницу 404 и устанавливает соответствующий код ответа сервера.

Например, по запросу /news будет подключен соответствующий контроллер, который в свою очередь отобразит пользователю страницу с новостями.

Код роутера находится в приложении В.

2.3.2 Верстка страниц

После того, как роутер был написан и протестирован, начался процесс верстки всех страниц. Для ускорения стилизации был использован CSS фреймворк Bootstrap.

Под версткой подразумевается реализация макета веб-сайта посредством HTML и CSS разметки, не касаясь при этом какой либо backend-логики.

В процессе верстки было принято решение выносить повторяющиеся элементы на страницах в модули – к таковым относятся header, footer и секция с названием страницы – и позже подключать их на страницы с помощью php функции require\_once.

В результате, была сверстана главная страница, как представлено на рисунке 9.



Рисунок 9 – Главная страница сайта

Как можно заметить, главная страница в полной мере соответствует заранее нарисованому макету

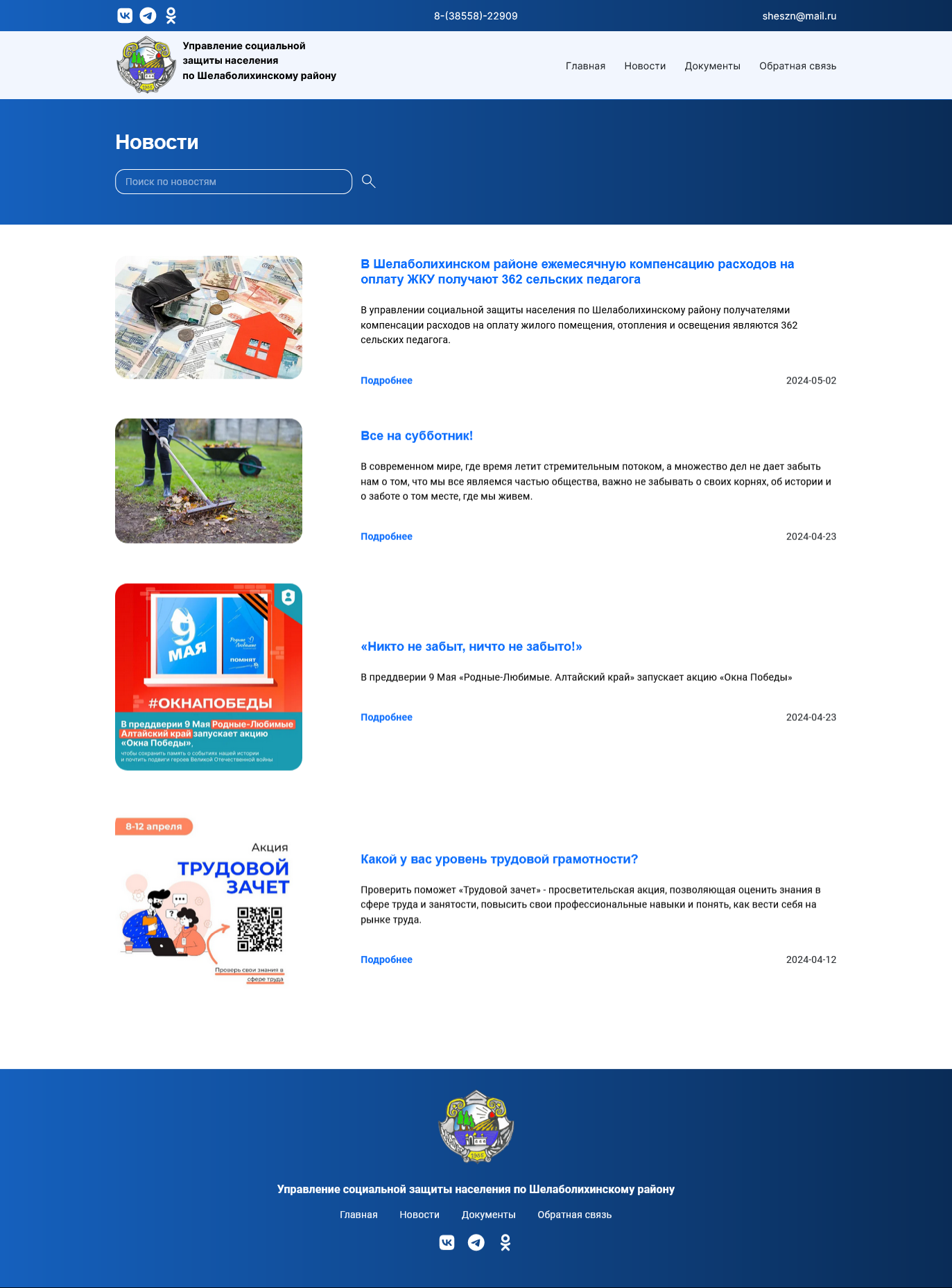
После, была сверстана страница с новостями, как показано на рисунке 10. 

Рисунок 10 – Страница с новостями

Также, была сверстана страница для просмотра новости, как представлено на рисунке 11.

Рисунок 11 – Страница с новостью

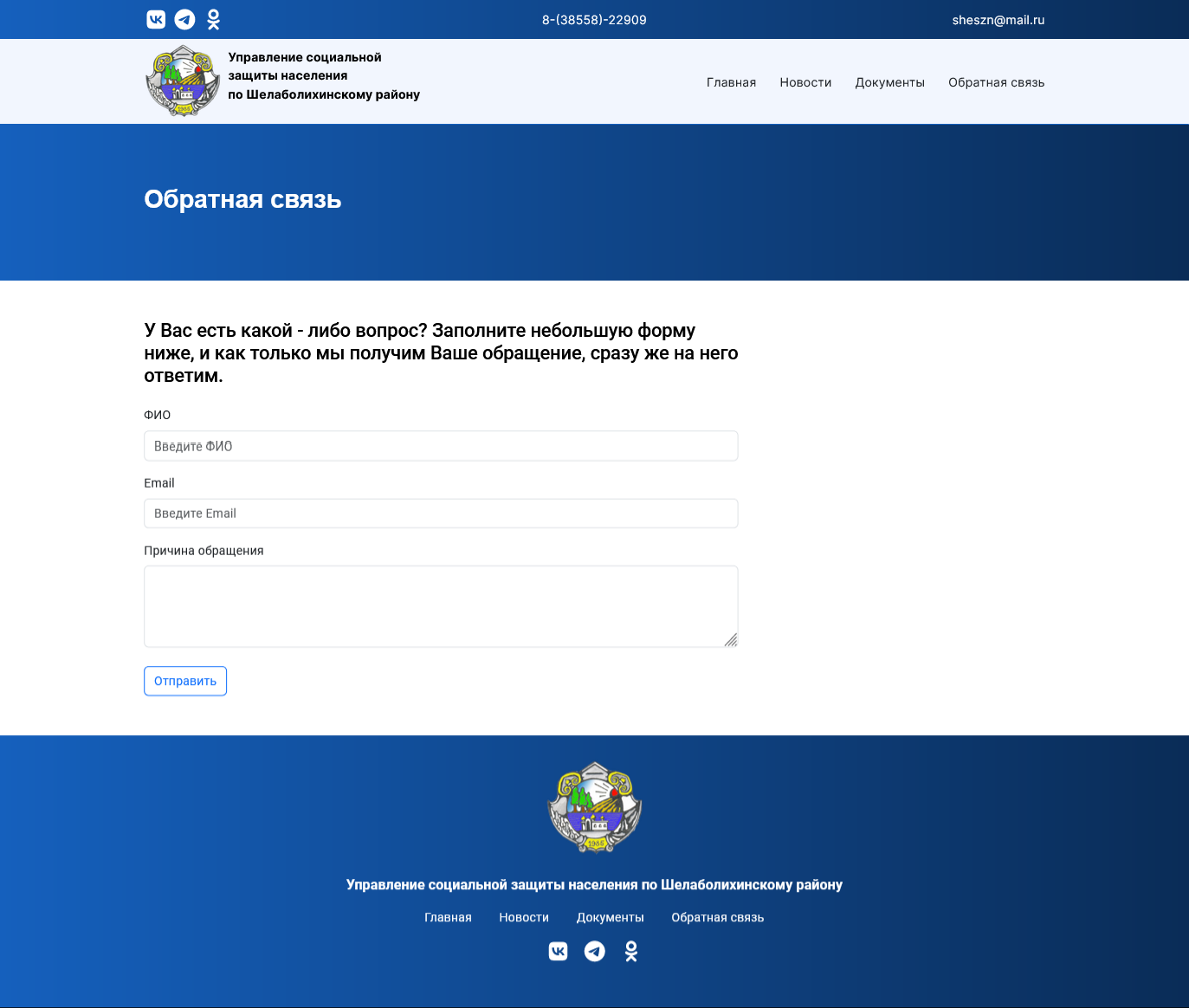
Далее была сверстана страница обратной связи, как представлено на рисунке 12.

Рисунок 12 – Страница обратной связи

После того, как были сверстаны все страницы, начался процесс написания логики веб-сайта.

2.3.3 Создание и внедрение базы данных в разработку

Для вывода большого количества информации – новостей, документов, а также для сохранения обращений от пользователей нужно использовать базу данных.

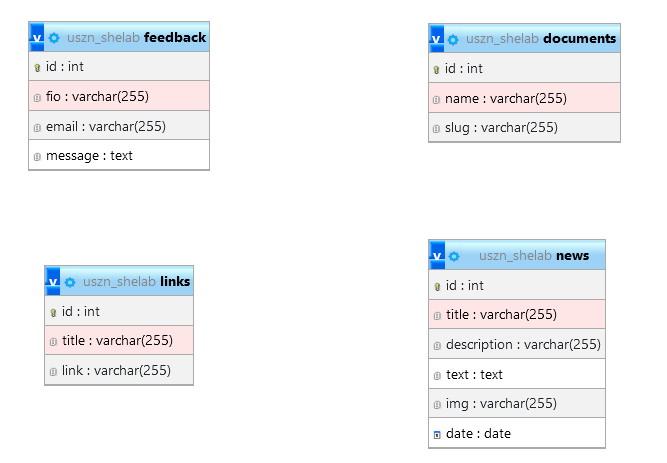
Была создана база данных uszn\_shelab, в соответствии с рисунком 13.

Рисунок 13 – Структура базы данных

В таблице feedback будут находится все обращения от пользователей, в таблице documents – все документы, в таблице links – ссылки введущие на главный сайт управления по Алтайскому краю, а в таблице news – новости.

После создания базы данных и наполнения таблиц записями, нужно подключить базу данных к сайту. Для этого был написан класс Db, содержащий в себе конструктор с подключением к базе данных через PDO, а также методы query, findAll, find, findOrFail.

Код класса Db представлен в приложении Г.

После подключения базы данных к сайту, пора приступать к выводу данных из таблиц.

Первыми были выведены новости. Последовательно вызывая метод query и findAll, в переменную $news были записаны все записи из таблицы в виде ассоциативного массива. Далее, используя цикл foreach, все новости были выведены на странице новостей. Стоит сказать, что почти вся информация, которая выводится на сайте, была выведена используя написанные методы и цикл foreach.

Чтобы прочитать новость полностью, был реализован следующий функционал: нажимая на название новости, происходит переход по адресу news/post с get параметром id, в котором содержится id новости. После, из базы данных выводится вся информация о новости, у которой id совпадает с id из get – запроса. Например, переходя на адрес news/post?id=6, на страницу выведется новость с id, равняющемся шести. Если же новости с указаным id не существует, сайт выдаст страницу 404. Реализовано это было с помощью метода findOrFail.

Далее, с помощью метода query и find, на главной странице была выведена последняя загруженная в базу данных новость.

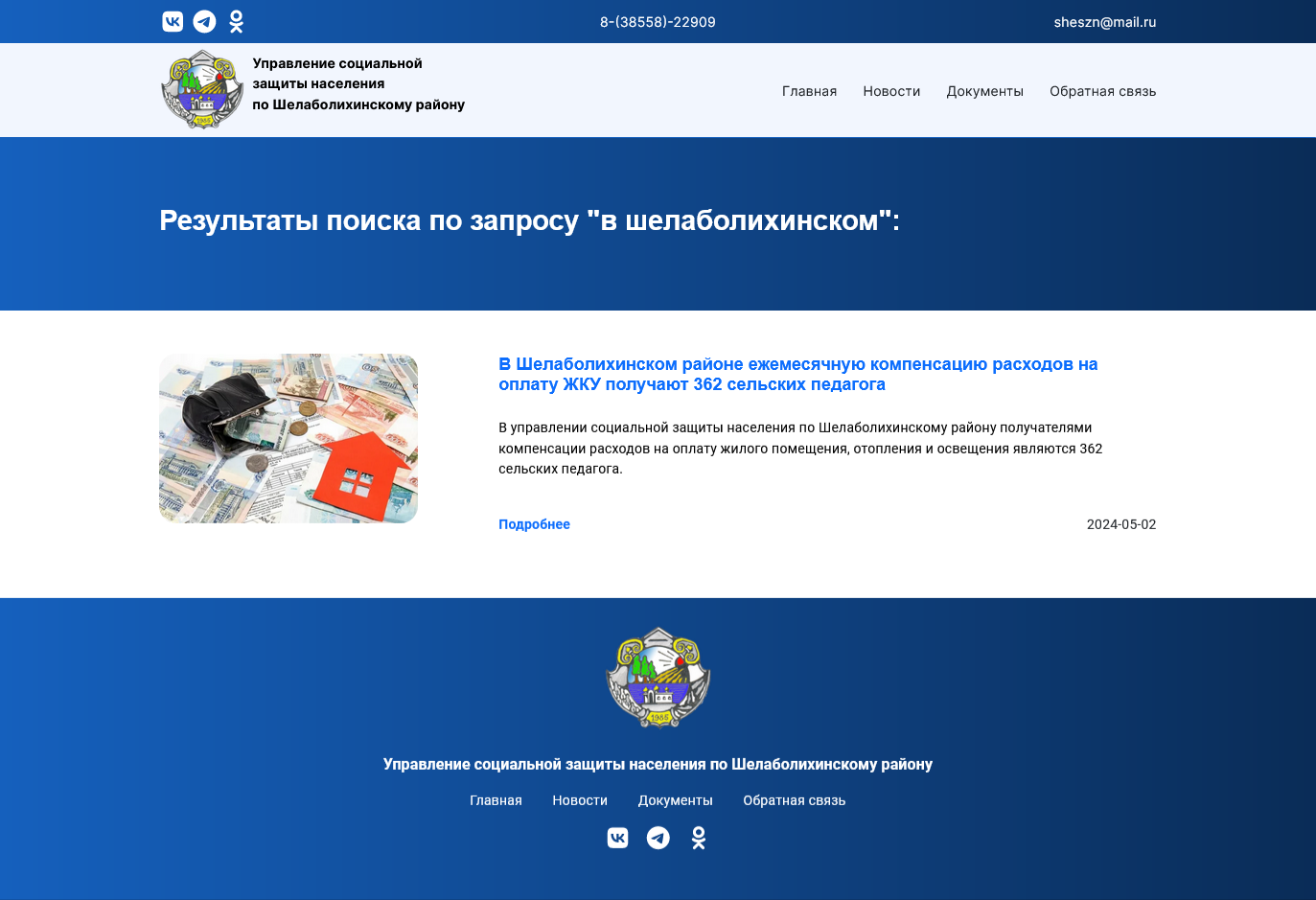
После, используя те же методы, что и для вывода новостей, а также SQL оператора LIKE был реализован функционал поиска, как показано на рисунке 14.

Рисунок 14 – Страница с результатами поиска

После, были выведены полезеные ссылки на главной странице, как показано на рисунке 15. 

Рисунок 15 – Секция с полезными ссылками

Далее, были выведены документы. Документы были реализованы ссылками, при нажатии на которые – будут открываться соответствующие документы.

После был реализован функционал отправки заявок от пользователей. Отправка данных реализована с помощью метода POST. Когда пользователь заполняет форму и отправляет ее, данные приходят на ту же страницу, в свою очередь в моделе происходит проверка, если страница получена методом POST – данные валидируются и записываются в базу данных, а пользователю выводится сообщение об успешной отправке.

Код записи обращения в таблицу представлен в приложении Д.

2.3.4 Адаптация сайта под мобильные устройства

Несмотря на то, что Bootstrap предоставляет адаптивную сетку и стили, некоторые элементы пришлось адаптировать вручную, с помощью медиа запросов и JavaScript. Например, создание бургер меню.

Если ширина экрана будет меньше пятиста пикселей, в header`е скроется меню и название сайта, но появится кнопка бургер меню, при нажатии на которую появится адаптированное меню, в сответствии с рисунком 16 и рисунком 17.

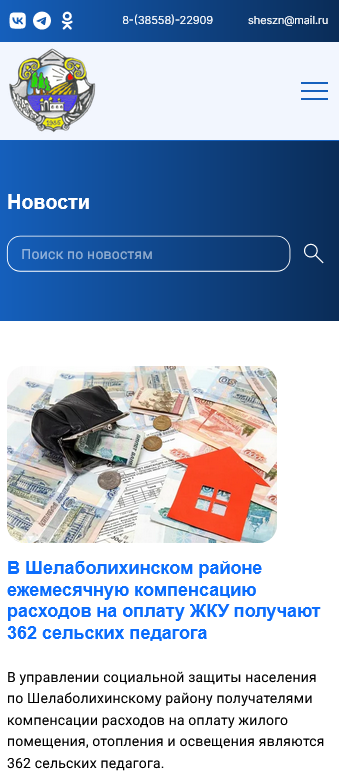


Рисунок 16 – Адаптированая страница для мобильных устройств

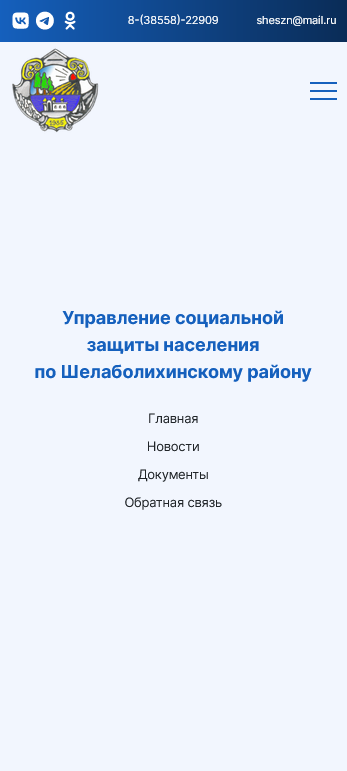


Рисунок 17 – Адаптированое меню для мобильных устройств

Код для бургер меню находится в приложении Е.

ЗАКЛЮЧЕНИЕ

В результате прохождения производственной (преддипломной) практики цель была достигнута, все задачи выполнены в полном объеме.

Привычную жизнь сложно представить без сети интернет. Всемирная паутина плотно вошла в повседневную реальность, ежедневно интернет посещают более четверти населения Земли. Интернет оказывает огромное влияние на различные аспекты нашей жизни, проникая в различные сферы деятельности и становясь неотъемлемой частью современного общества.

Создание качественного веб-сайта – залог успеха любого предприятия. Важно, чтобы он был интуитивно понятен для пользователей, удобен, был «отражением» организации.

В процессе выполнения задач практики был спроектирован и разработан веб-сайт краевого государственного казенного учреждения «Управление социальной защиты населения по Шелаболихинскому району». Сайт состоит из нескольких страниц, предоставляющих пользователю информацию об организации, новости, документы и возможность обратной связи.

В заключение практики был разработан сайт, было выполнено тестирование сайта, корректности внешнего вида, а также проверена адаптивность сайта к мобильным устройствам.

Итогом прохождения производственной (преддипломной) практики стало развитие навыков самостоятельной работы в соответствии с профессиональными стандартами, а также глубокое освоение современных методов и технологий проектирования веб-сайтов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Грант, К. CSS для профи / К. Грант. – Санкт-Петербург: Питер, 2021. – 496 с.
2. Дакетт, Д. HTML и CSS. Разработка и создание веб-сайтов / Д. Дакетт. – Москва: Эксмо, 2019. – 480 с.
3. Дакетт, Д. PHP и MYSQL. Серверная веб-разработка / Д. Дакетт. – Москва: Эксмо, 2023. – 688 с.
4. Томсон, В. Разработка веб-приложений с помощью PHP и MySql / В.Томсон. – М.: Диалектика, 2017. – 768 с.
5. Минник, К. HTML5 и CSS3 для чайников / Крис Минник, Эд. Титтел. – М.: Диалектика, 2019. – 400 c.
6. Роббинс, Дж. Веб-дизайн для начинающих. HTML, CSS, JavaScript и веб-графика / Дж. Роббинс. – Санкт-Петербург: BHV-СПб, 2021. – 956 с.
7. Шкрыль, А.А. PHP – это просто. Программируем для Web-сайта / А.А.Шкрыль. – СПб.: БХВ-Петербург, 2007. – 367 с.
8. Стоян, С. JavaScript. Шаблоны / С. Стоян. – Москва: Символ-Плюс, 2011. – 272 с.
9. Татро, К., Макинтайр, П. Создаём динамические веб-сайты на PHP / К. Татро, П. Макинтайр. – Санкт-Петербург: Питер, 2021. – 544 с.
10. Ташков, П.А. Веб-мастеринг на 100 %: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка / П.А. Ташков. – СПб.: Питер, 2014. – 659 c.
11. "Веб-сайт в интернете" [сайт]. URL: https://semantica.in/blog/chto–takoe–veb–sajt.html (дата обращения 01.05.2024)
12. "Разбираемся в типах сайтов. Полный список" [сайт]. – URL: https://yanakhodkina.com/razbiraemsya-v-tipax-sajtov-polnyj-spisok/ (дата обращения: 02.05.2024)
13. "Руководство по PHP" [сайт]. URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 05.05.2024)
14. "Структура сайта: понятие, типы, алгоритм создания" [сайт]. – URL: https://surgay.ru/blog/struktura–sayta/ (дата обращения: 07.05.2024)
15. "Что такое сайт" [сайт]. – URL: https://www.alkosto.ru/blog/chto–takoe–sajt/?utm\_referrer=https%3A%2F%2Fyandex.ru%2F (дата обращения: 09.05.2024)

ПРИЛОЖЕНИЕ А

RewriteEngine on

RewriteRule (.\*) public\_html/$1

ПРИЛОЖЕНИЕ Б

RewriteEngine on

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule (.\*) index.php?$1 [L,QSA]

ПРИЛОЖЕНИЕ В

<?php

require\_once CONFIG . '\routes.php';

$uri = parse\_url($\_SERVER['REQUEST\_URI'], PHP\_URL\_PATH); if(array\_key\_exists($uri, $routes))

{

if(file\_exists(CONTROLLERS . "/{$routes[$uri]}"))

{

require\_once CONTROLLERS . "/{$routes[$uri]}";

}

else

{

abort();

}

}

else

{

abort(404);

}

ПРИЛОЖЕНИЕ Г

<?php

class Db

{

private $connect;

private $stmt;

public function \_\_construct(array $db\_config)

{

$dsn = "mysql:host={$db\_config['host']};dbname={$db\_config['dbname']}";

try

{

$this-> connect = new PDO ($dsn, $db\_config['user'], $db\_config['password']);

}

catch(PDOException $e)

{

die("Ошибка подключения к базе данных: {$e -> getMessage()}");

}

}

public function query($sql, $params = [])

{

$this -> stmt = $this -> connect -> prepare($sql);

$this -> stmt -> execute($params);

return $this;

}

public function findAll()

{

return $this -> stmt -> fetchAll();

}

public function find()

{

return $this -> stmt -> fetch();

}

public function findOrFail()

{

$result = $this -> find();

if(!$result)

{

abort();

}

return $result;

}

}

ПРИЛОЖЕНИЕ Д

<?php

if($\_SERVER['REQUEST\_METHOD'] == "POST")

{

$fillable = ['fio', 'email', 'message'];

$data = validate($fillable);

foreach($data as $k => $v)

{

$data[$k] = trim($v);

$data[$k] = strip\_tags($v);

}

$result = $db->query("INSERT INTO `feedback` (fio, email, message) VALUES(?, ?, ?)", [$data['fio'],$data['email'],$data['message']]);

if($result)

{

$info = $\_SESSION['msg'] = "Ваше обращение успешно отправлено";

}

}