Федеральное государственное образовательное бюджетное учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

(наименование структурного подразделения)

Дипломный проект

Тема Разработка веб-сайта охранного предприятия

(наименование)

Студент (ка) Гладков Артемий Ильич

(фамилия, имя, отчество полностью)

Учебная группа 4ПКС-120

Специальность 09.02.03 «Программирование в компьютерных системах»

(код и наименование специальности)

|  |  |  |
| --- | --- | --- |
| Руководитель  дипломного проекта |  | И.В. Сибирев |
|  | (подпись) | (инициалы, фамилия) |
| Председатель предметной (цикловой) комиссии |  |  |
|  | (подпись) | (инициалы, фамилия) |

Москва – 2024г.

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc168002582)

[Глава 1. Предпроектное исследование 6](#_Toc168002583)

[1.1. Описание предметной области 6](#_Toc168002584)

[1.2. Сравнительный анализ программ-аналогов 7](#_Toc168002585)

[1.3. Постановка задачи 8](#_Toc168002586)

[1.4. Характеристика инструментальных средств разработки 9](#_Toc168002587)

[Глава 2. Проектирование и реализация сайта охранного предприятия 12](#_Toc168002588)

[2.1. Проектирование веб-сайта 12](#_Toc168002589)

[2.2. Проектирование программного обеспечения 15](#_Toc168002590)

[2.3. Разработка программного обеспечения 17](#_Toc168002591)

[2.4. Отладка и тестирование программного обеспечения 37](#_Toc168002592)

[2.5. Руководство по использованию программы 42](#_Toc168002593)

[ЗАКЛЮЧЕНИЕ 44](#_Toc168002594)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ 46](#_Toc168002595)

[ПРИЛОЖЕНИЕ 49](#_Toc168002596)

# **ВВЕДЕНИЕ**

В современной модели российской экономики частная охранная деятельность играет важную роль в обеспечении личной и имущественной безопасности граждан и предприятий. Развитие информационных технологий и автоматизация процессов в целом стали неотъемлемой частью экономики России, что привело также к внедрению различных веб-компонентов и созданию сайтов, предназначенных для управления бизнес-моделями в области частной охраны и рекламы охранных услуг.

Целью дипломного проекта является разработка веб-приложения «Охранное предприятие» с использованием современных веб-технологий, таких как HTML, CSS, JavaScript. Веб-сайт предназначен для эффективного управления различными сферами деятельности охранного предприятия, включая информацию о компании, рекламу услуг, наличие вакансий, условия приёма на работу и размер оплаты труда, контактные данные, а также обратную связь.

Кроме перечисленных выше инструментальных средств используются следующие программные средства: Microsoft Visual Code, MySQL, PlantUML, Draw.io, Microsoft Word и Microsoft PowerPoint.

Результатом проекта будет создание функционального веб-сайта частного охранного предприятия, способствующего улучшению управления и достижения высокой эффективности работы охранного бизнеса, а также повышающего уровень безопасности объектов охраны (клиентов).

Важно отметить, что проект не ограничен лишь описанием процесса разработки веб-сайта «визитки». Также будут даны рекомендации и инструкции по использованию созданного веб-сайта и поддержанию его актуальности. Веб-сайт послужит практическим примером успешного внедрения современных информационных технологий.

Дипломный проект включает в себя решение следующих задач:

1. анализ предметной области, включающий изучение особенностей охранного бизнеса, требований клиентов и тенденций рынка в области безопасности;
2. сравнительный анализ аналогов, где проводится исследование существующих веб-сайтов частных охранных организаций, выявление их преимуществ и недостатков;
3. постановка задачи индивидуального проекта, включающая определение основных целей и функциональных требований к веб-сайту частной охранной организации;
4. характеристика инструментальных средств, где происходит выбор сред разработки для создания эффективного и надёжного веб-сайта;
5. построение диаграмм и схем, включающее разработка структурных и функциональных диаграмм, отображающих взаимосвязи между компонентами сайта охранного предприятия;
6. разработка базы данных для хранения информации о логине и пароле администратора, а также списка условий;
7. реализация функциональных возможностей сайта, включая создание функциональных условий для вывода пользователю информации о компании, её контактных данных, а также других сведений, необходимых для получения представления о деятельности частной охранной организации;
8. разработка веб-сайта с написанием кода, обеспечивающего удобный интерфейс и эффективное взаимодействие с пользователями;
9. оформление пояснительной записки и подготовка презентации, включающее подготовку документации, описывающей процесс разработки, и создание презентации для защиты дипломного проекта.

Объектом исследования является охранное предприятие как организация, специализирующаяся на предоставлении охранных услуг организациям и частным лицам.

Предмет исследования: веб-технологии по проектированию и разработке веб-сайта «визитки» для частного охранного предприятия (организации).

Методика исследования включает в себя анализ предметной области охранного бизнеса, выявление требований к веб-сайту, анализ текущих процессов безопасности и выделение узких мест и проблем. Далее предполагается разработка рекомендаций по оптимизации бизнес-процессов в охранном предприятии и тестирование предложенных изменений.

Для осуществления дипломного проекта будут использованы научные статьи, учебники по технологии разработки программного обеспечения, официальные документы в области безопасности, а также стандарты и ГОСТы, касающиеся информационных технологий и безопасности.

Для обеспечения безопасности доступа к данным на создаваемом веб-сайте будет реализована аутентификация только для администратора с использованием логина и хешированного пароля. Это обеспечит безопасность доступа к данным, предотвращая несанкционированный доступ.

Создание веб-сайта для охранного предприятия также позволит повысить информационную доступность и привлекательность компании. Это обеспечит более качественное представление условий и контактов для потенциальных клиентов, повысит привлекательность и статус компании.

# **Глава 1. Предпроектное исследование**

# **1.1. Описание предметной области**

В сфере обеспечения общественной безопасности охранное предприятие играет существенную роль. Разработка веб-сайта для такой компании направлена на эффективное взаимодействие с клиентами, предоставляя информацию о предлагаемых условиях и других аспектах, обеспечивая удобство и надежность обслуживания.

Основные аспекты предметной области:

* услуги и методы обеспечения безопасности: подробное описание услуг, которые предоставляет охранная компания, включая охрану объектов, физическую защиту, видеонаблюдение, техническое обеспечение безопасности и другие методы. Также предоставляется информация о квалификации сотрудников, используемых технологиях и методиках работы;
* продукты и оборудование: описание оборудования и технических средств, которые используются для обеспечения безопасности объектов. Включает информацию о доступных продуктах, их технических характеристиках и возможностях применения;
* клиентский сервис и поддержка: разработка функционала для связи с клиентами, включая формы обратной связи, онлайн-консультации и поддержку. Гарантируется оперативное реагирование на запросы и вопросы клиентов, а также предоставление квалифицированных консультаций;
* контактная информация и обратная связь: обеспечение доступа к контактным данным охранной компании, включая адреса офисов, телефоны и электронную почту. Разработка удобных форм обратной связи для оперативного общения с клиентами;
* адаптивный дизайн и безопасность: разработка веб-сайта с адаптивным дизайном, который обеспечивает удобство использования на различных устройствах. Гарантируется высокий уровень безопасности при обработке персональных данных клиентов и конфиденциальной информации.

# **1.2. Сравнительный анализ программ-аналогов**

Для проведения сравнительного анализа охранного предприятия рассмотрим несколько ведущих компаний в сфере охраны и сравним их по ряду критериев, указанных в таблице 1.

Таблица 1. Сравнительный анализ компаний в сфере охраны

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | ЧОП «ФРОНТ-СБ» | ЧОП "ЛЮБЕР" | ЧОО "ПРИОРИТЕТ" |
| Квалификация персонала | - Охранники имеют лицензии и проходят регулярное обучение по программам безопасности;  - Работники службы поддержки клиентов обладают профессиональными навыками и могут оперативно реагировать на запросы клиентов. | - Охранники имеют лицензии и проходят регулярное обучение по программам безопасности;  - Работники службы поддержки клиентов обладают профессиональными навыками и могут оперативно реагировать на запросы клиентов. | - Охранники имеют лицензии и проходят регулярное обучение по программам безопасности;  - Работники службы поддержки клиентов обладают профессиональными навыками и могут оперативно реагировать на запросы клиентов. |
| Технические средства | - Обеспечивает высокие стандарты технического обеспечения безопасности. | - Имеет системы контроля доступа и пожарной сигнализации, проходящие регулярное обслуживание.;  - Все технические средства регулярно проходят техническое обслуживание. | - Использует современные системы видеонаблюдения с функциями обнаружения движения и распознавания лиц;  - Все технические средства регулярно проходят техническое обслуживание. |
| Уровень сервиса | - Оказывается круглосуточная поддержка клиентов по вопросам безопасности;  - Клиенты могут получить консультацию и помощь в любое время суток. | - Компания предоставляет гибкие условия обслуживания объектов с учетом специфики бизнеса клиента;  - Клиенты могут получить консультацию и помощь в любое время суток. | - Оказывается индивидуальный подход к каждому клиенту, учитывая его потребности и особенности объекта;  - Клиенты могут получить консультацию и помощь в любое время суток. |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | ЧОП «ФРОНТ-СБ» | ЧОП "ЛЮБЕР" | ЧОО "ПРИОРИТЕТ" |
| Ценовая политика | - Компания предлагает конкурентоспособные цены на услуги безопасности с учетом объема и продолжительности обслуживания;  - Для постоянных клиентов действуют специальные условия и скидки. | - Ценообразование компании прозрачно и соответствует рыночным стандартам;  - Дополнительные услуги и сервисы клиенты оплачивают отдельно. | - Цены на услуги охраны устанавливаются индивидуально для каждого клиента с учетом его потребностей и бюджета;  - Для постоянных клиентов действуют специальные условия и скидки. |

Исходя из проведенного сравнительного анализа, можно сделать следующие выводы:

1. каждая из компаний предлагает высококвалифицированный персонал и использование современных технических средств для обеспечения безопасности клиентов[14][15][16];
2. уровень сервиса во всех трех компаниях высокий, с учетом индивидуального подхода к каждому клиенту и оперативной поддержки;
3. ценовая политика компаний различается, но все они предлагают конкурентоспособные цены и гибкие условия для клиентов;
4. компании стремятся обеспечить надежность и безопасность своих услуг, что делает их привлекательными для клиентов.

# **1.3. Постановка задачи**

Для программной реализации необходимо определиться с функциональными и эксплуатационными требованиями, входными и выходными данными веб-сайта:

Входные данные:

1. информация о предприятии:

* название компании;
* объявление об имеющихся вакансиях;
* описание условий;
* контактные данные (телефоны и адреса);
* фотографии, демонстрирующие деятельность компании.

Выходные данные:

1. интерфейс сайта:

* отображение информации о предприятии, структурированной и удобной для навигации;
* представление контактных данных в доступной форме.

Функциональные требования:

1. отображение информации:

* предоставление пользователю информации о предприятии и его условиях;
* возможность просмотра фотографий и видео;
* отображение контактных данных.

1. форма обратной связи:

* наличие формы обратной связи для связи с предприятием;
* возможность отправки сообщений клиентами.

Эксплуатационные требования:

1. адаптивность:

* корректное отображение сайта на различных устройствах (планшеты, смартфоны, ПК);
* адаптация дизайна и контента под разные разрешения экранов.

1. безопасность:

* применение мер по защите от кибератак и взлома.

1. тестирование и отладка:

* проведение тестирования функционала сайта перед запуском;
* отладка и устранение обнаруженных ошибок.

1. поддержка и обновление:

* обновление информации на сайте по мере необходимости.

# **1.4. Характеристика инструментальных средств разработки**

В процессе работы над проектом использованы следующие инструментальные средства:

1. HTML (Hyper Text Markup Language):

HTML позволяет создать структуру веб-страницы и определить расположение контента. Этот язык позволяет организовать информацию на странице с помощью различных тегов. [20]

1. CSS (CascadingStyleSheets):

CSS применяется для стилизации веб-страницы и придания ей визуального оформления. С помощью языка CSS формируется дизайн веб-сайта – выбирается цвет текста и фона, шрифт, размеры отступов, рамки и прочее. [21]

1. JavaScript:

JavaScript добавляет веб-странице элемент интерактивности и обработки действий пользователя. Язык программирования JavaScript позволяет создавать динамические элементы, управлять анимациями, валидировать формы, отправлять запросы на сервер и многое другое.

1. PHP:

PHP (Hypertext Preprocessor) - это язык программирования, который часто применяется при создании динамических веб-сайтов и разработке веб-приложений. [22]

1. MySQL:

MySQL представляет собой систему управления базами данных (СУБД) с реляционной моделью данных, которая применяется для хранения и управления информацией в веб-приложениях. MySQL обеспечивает эффективное хранение информации, а также поддерживает операции CRUD (Create, Read, Update, Delete) для работы с данными.

1. Microsoft Visual Code

Microsoft Visual Code - интегрированная среда разработки (IDE), позволяющая работать с HTML, CSS, JavaScript и другими языками программирования, а также управлять проектами.

1. Open Server Panel

Open Server Panel - это инструмент для локальной разработки веб-приложений, который обеспечивает удобный интерфейс для управления сервером и настройкой окружения разработки.

1. PhpMyAdmin

PhpMyAdmin - это веб-приложение, предоставляющее удобный интерфейс для администрирования баз данных MySQL. Оно упрощает выполнение разнообразных операций с данными, включая создание таблиц, выполнение запросов и многое другое.

1. описание инструментов для построения диаграмм и схем:

PlantUML и Draw.io – онлайн сервисы для разработки спецификаций. [17][18]

1. описание программных продуктов для оформления отчета:

Microsoft Word – это текстовый редактор, который позволяет создавать, редактировать и форматировать документы.

Microsoft PowerPoint – это инструмент для создания и презентации слайд-шоу и презентаций. [24]

# **Глава 2. Проектирование и реализация сайта охранного предприятия**

# **2.1. Проектирование веб-сайта**

В данном разделе представлен процесс проектирования веб-сайта охранного предприятия с помощью диаграмм, которые помогут наглядно представить структуру и функциональность проекта.

На рисунке 1 представлена диаграмма «сущность-связь» для разработки базы данных для сайта охранного предприятия.



Рисунок 1. Диаграмма «сущность-связь»

Рассмотрим сущности, которые будут использоваться для выполнения хранения информации в БД:

* About – для хранения описания;
* Advantages – для хранения данных о преимуществах;
* Application – для хранения заявок клиентов;
* Contact – для хранения данных контактной информации;
* Header – для хранения данных из шапки сайта;
* License – для хранения данных об лицензии;
* Services – для хранения данных об условиях;
* Slider – для хранения данных о деятельности компании в фотографиях;
* User – для хранения данных администратора;

На рисунке 2 представлена диаграмма вариантов использования для визуализации, как различные пользователи могут взаимодействовать с приложением. [18]

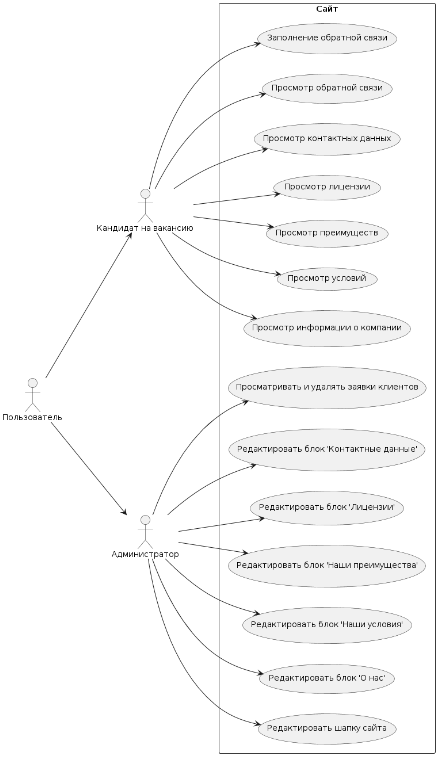


Рисунок 2. Диаграмма вариантов использования

На диаграмме показано, что у кандидата на вакансию и администратора разные уровни доступа к приложению:

1. кандидат на вакансию – может просматривать информацию на сайте и отправлять заявки;
2. администратор – может добавлять, изменять и удалять информацию на сайте.

На рисунках 3,4 представлены диаграммы последовательности администратора и кандидата на вакансию для визуализации последовательности действий, выполняемых при взаимодействии с системой. Они помогают понять, как каждый пользователь взаимодействует с сайтом: как они входят в систему, как отправляют заявки, и какие шаги предпринимают для выполнения определенных задач. [18]

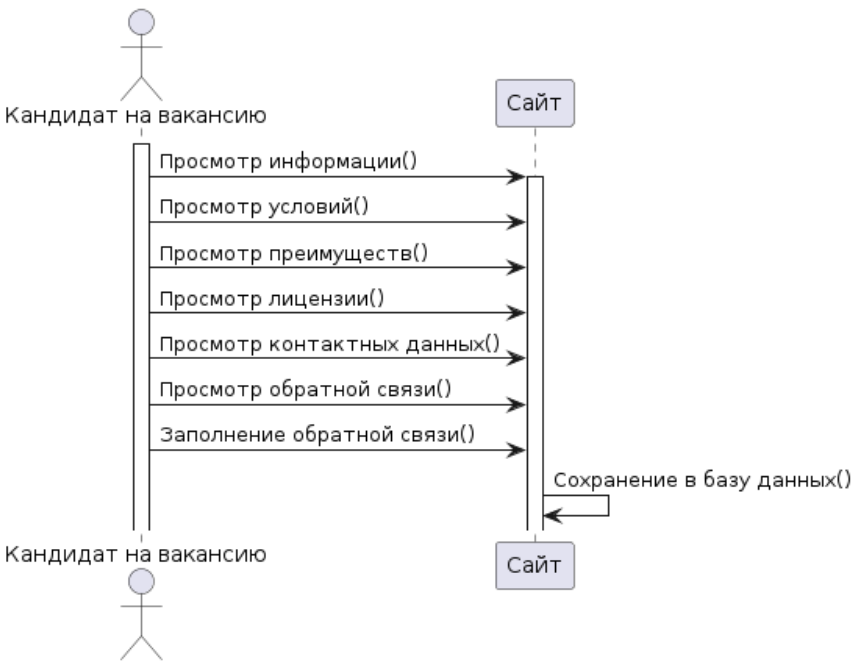


Рисунок 3. Диаграмма последовательности кандидата на вакансию

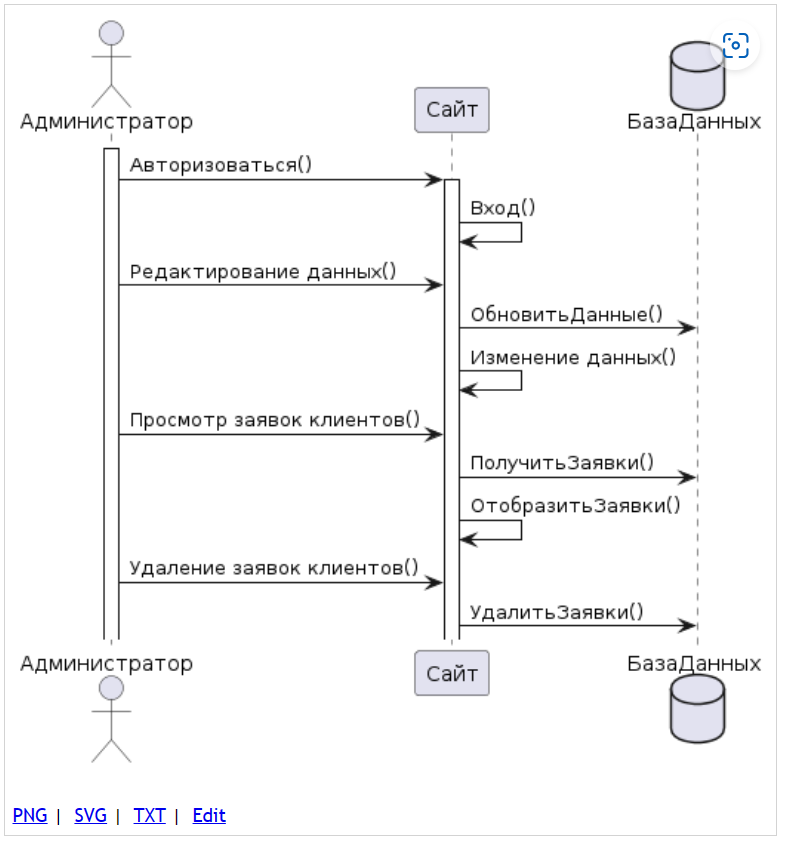


Рисунок 4. Диаграмма последовательности администратора

# **2.2. Проектирование программного обеспечения**

На рисунке 5 представлена структурная схема. [17]

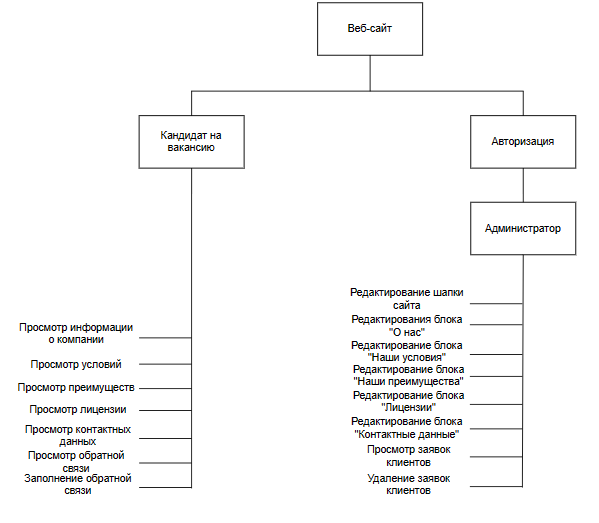


Рисунок 5. Структурная схема

На данном рисунке показана структура приложения и возможности пользователя исходящие из его роли.

Также в ходе разработки индивидуального проекта была разработана функциональная схема, представленная на рисунке 6. [17]

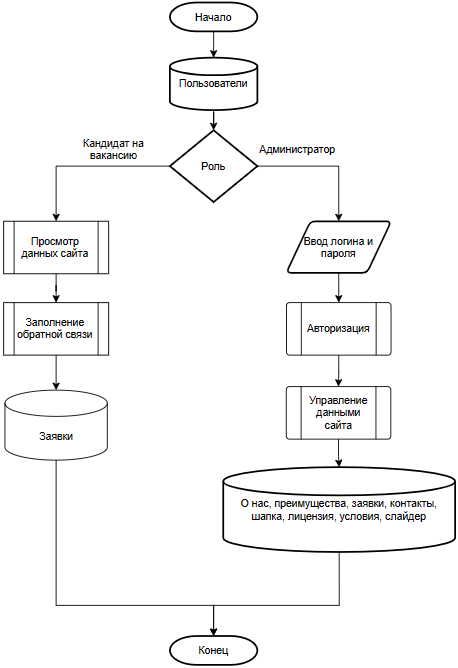


Рисунок 6. Функциональная схема

Функциональная схема подробно описывает работу сайта охранного предприятия.

На рисунке 7 представлена ER-диаграмма базы данных информационной системы «Компьютерный салон» построенная в СУБД MySQL.

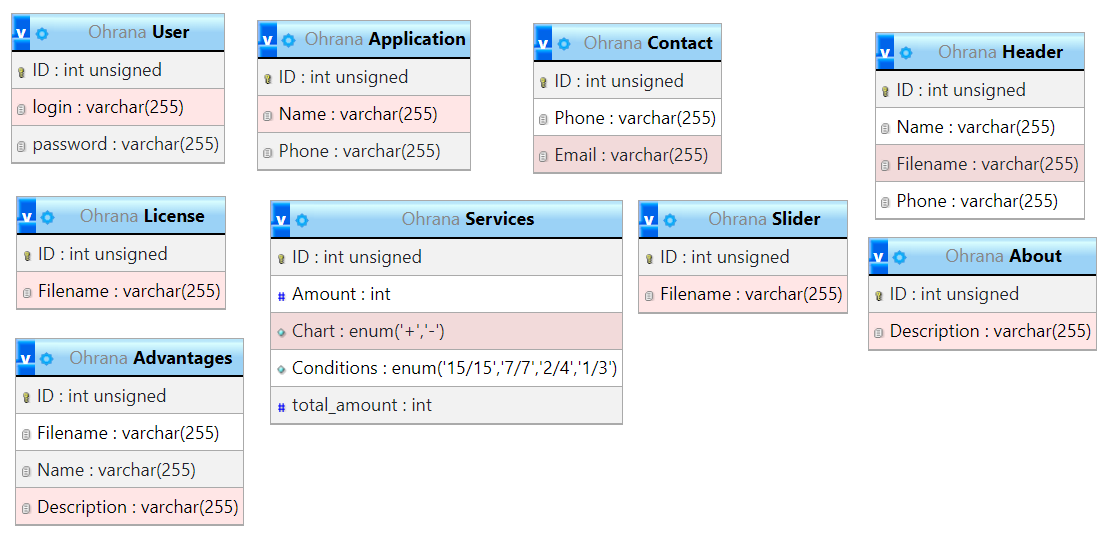


Рисунок 7. ER-диаграмма

В таблице представлен словарь данных для ER-диаграммы, который описывает структуру всех таблиц базы данных (см. Приложение №1).

# **2.3. Разработка программного обеспечения**

В начале разработки приложения нужно подключить созданную ранее базу данных.

Ниже приведен фрагмент кода, демонстрирующий, как осуществляется подключение к базе данных:

|  |
| --- |
| <?php  $user = "root";  $password = "";  $host="localhost";  $db="Ohrana";  $dbh = 'mysql:host='.$host.';dbname='.$db.';charset=utf8';  $pdo=new PDO($dbh, $user, $password); |

При открытии сайта пользователь попадает на основную страницу, где изображена вёрстка сайта.

Изначально был реализован код для общих стилей, таких как фон, отступы, шрифт, цвет текста. Также была сделана плавная прокрутка страницы при переходе по навигации и сделан общий стиль для названия каждого блока. Для реализации этого был использован следующий код:

|  |
| --- |
| body {      margin: 0;      padding: 0;      font-family: 'Inter', sans-serif;      background: #f0f0f0;      color: #333;  }  html {      scroll-behavior: smooth;  }  .title {      text-align: center;      font-size: 32px;      margin-top: 50px;  } |

После этого для удобства было сделано разбиение файлов на несколько частей, чтобы работать над каждым блоком отдельно. Ниже представлен код, который отображает все пути к этим файлам:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>ЧОО "Пельтаст"</title>      <link rel="stylesheet" href="css/style.css">      <link rel="stylesheet" href="css/font-awesome.css">  </head>  <body>      <?php require 'public/header.php'?>      <?php require 'public/invite.php'?>      <script src="js/navigation.js"></script>      <?php require 'public/about.php'?>      <?php require 'public/service.php'?>      <?php require 'public/advantages.php'?>      <?php require 'public/license.php'?>      <?php require 'public/contact.php'?>      <?php require 'public/footer.php'?>      <script src="js/telephone.js"></script>  </body>  </html> |

Сверху находится шапка сайта с навигацией, которая представлена на рисунке 8.

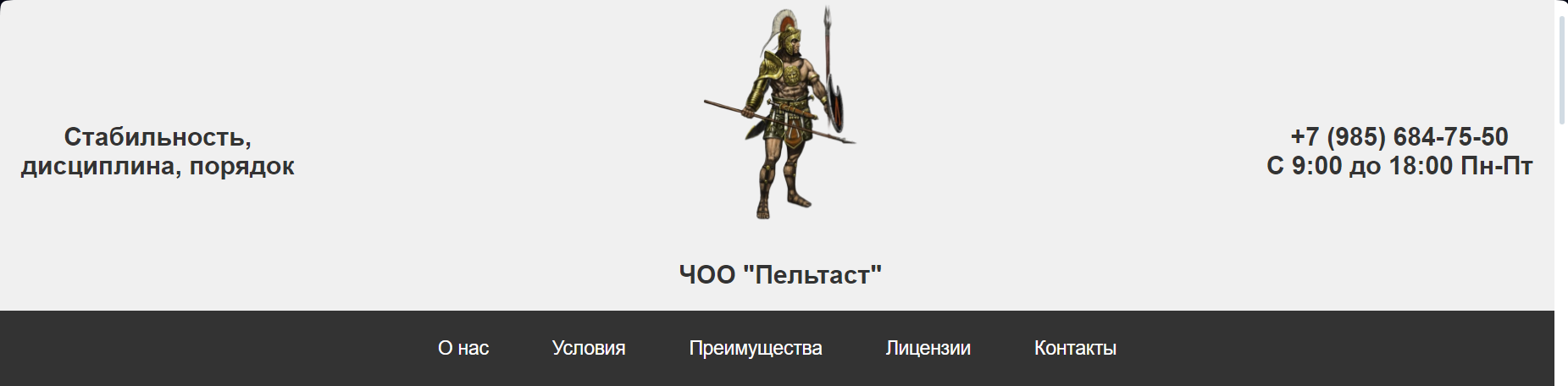


Рисунок 8. Шапка сайта

Для реализации разметки был использован код, представленный ниже:

|  |
| --- |
| <?php require\_once './functions/connect.php';?>    <?php      $main = $pdo->prepare("SELECT \* FROM Header");      $main->execute();      $uslugi=$main->fetch(PDO::FETCH\_OBJ);  ?>  <header>  <div class="header-container">      <div class="Three\_blocks border-box">          <div class="One\_line centered-content">              <h2 class="title-text">Стабильность, <br/> дисциплина, порядок</h2>          </div>          <div class="One\_line centered-content logo-block">              <div>                  <img src="admin/header/img/<?php echo $uslugi->Filename?>" alt="Логотип" width="150" height="200"/> <br/>                  <h2 class="title-text"><?php echo $uslugi->Name?></h2>              </div>          </div>          <div class="One\_line centered-content contact-block">              <div class="contact-info">                  <h2><?php echo $uslugi->Phone?><br/></h2>                  <h2>С 9:00 до 18:00 Пн-Пт</h2>              </div>          </div>      </div>  </div>          <nav class="navigation">            <ul>                <li><a href="#us">О нас</a></li>                <li><a href="#services">Условия</a></li>                <li><a href="#advantages">Преимущества</a></li>                <li><a href="#licenses">Лицензии</a></li>                <li><a href="#contacts">Контакты</a></li>            </ul>        </nav>  </header> |

Для реализации переходов по навигации был написан следующий код JavaScript, который представлен ниже:

|  |
| --- |
| const navLinks = document.querySelectorAll('.navigation a');  navLinks.forEach(link => {      link.addEventListener('click', function(event) {          event.preventDefault();          const targetId = this.getAttribute('href');          const targetElement = document.querySelector(targetId);          targetElement.scrollIntoView();      });  });  const servicesButton = document.getElementById('servicesButton');  servicesButton.addEventListener('click', function() {      const servicesSection = document.getElementById('services');      servicesSection.scrollIntoView({ behavior: 'smooth' });  });  function scrollToBottom() {      window.scrollTo(0, document.body.scrollHeight);  } |

После представлен блок с изображением, текстовым оверлеем и двумя кнопками, которые также работают, как и навигация.

Ниже представлен код реализации разметки этого блока:

|  |
| --- |
| <div class="image-container">          <img src="photo/maket1.jpg" alt="Охрана">          <div class="overlay-text">Приглашаем вас присоединиться <br/> к нашей профессиональной охранной службе</div>          <div class="buttons-container">              <button class="button" id="showFormButton" onclick="scrollToBottom()">Получить консультацию</button>              <button class="button1" id="servicesButton">Посмотреть условия</button>          </div>  </div> |

На рисунке 9 представлен блок «О нас», включающий описание компании и демонстрацию её деятельности через слайдер, с помощью которого можно пролистывать фотографии.

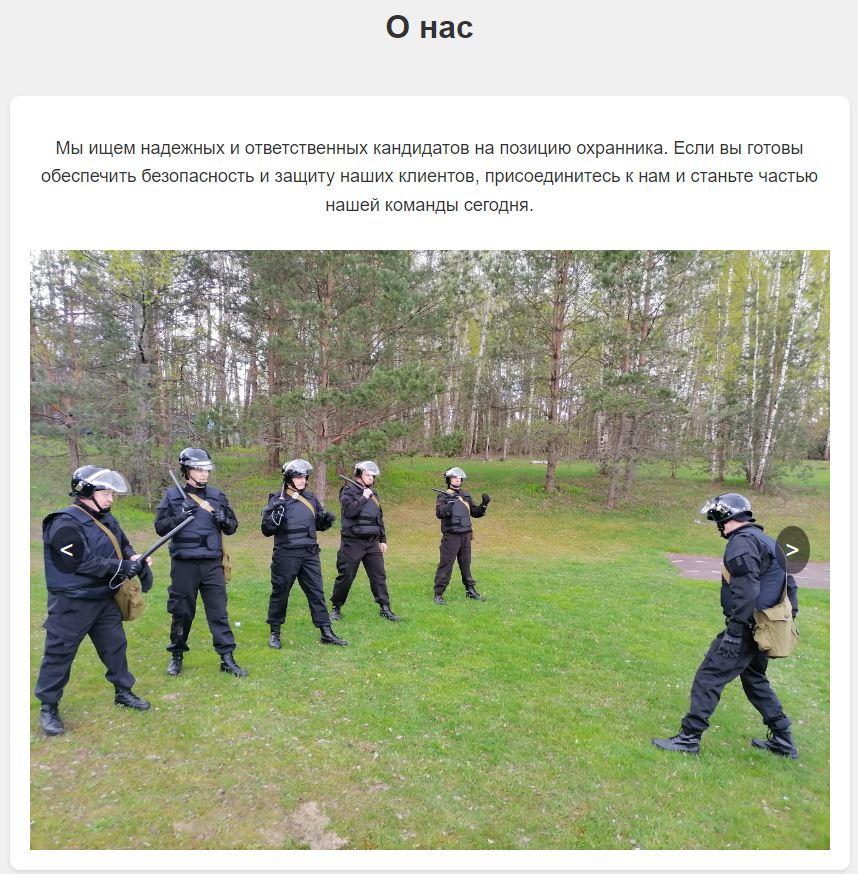


Рисунок 9. Блок «О нас»

Для реализации разметки в этом блоке был использован код, который представлен ниже:

|  |
| --- |
| <?php require\_once './functions/connect.php';?>    <?php      $main = $pdo->prepare("SELECT \* FROM About");      $main -> execute();      $about = $main->fetch(PDO::FETCH\_OBJ);      $main2 = $pdo->prepare("SELECT \* FROM Slider");      $main2 -> execute();      $sliders =  $main2->fetchAll(PDO::FETCH\_OBJ);  ?>  <section id="us">        <h1 class="title">О нас</h1>        <div class="container1">          <p><?php echo $about->Description?></p>          <div class="slider-container">              <div class="slider">              <?php foreach($sliders as $slider):?>                  <img src="/admin/about/img/<?php echo $slider->Filename?>">              <?php endforeach ?>              </div>              <button class="prev-button">&lt;</button>              <button class="next-button">&gt</button>          </div>          <script src="js/slider.js"></script>        </div>  </section> |

Для реализации слайдера был написан следующий код JavaScript, который представлен ниже:

|  |
| --- |
| const slider = document.querySelector('.slider');  const prevButton = document.querySelector('.prev-button');  const nextButton = document.querySelector('.next-button');  const slides = Array.from(slider.querySelectorAll("img"));  const slideCount = slides.length;  let slideIndex = 0;  prevButton.addEventListener('click', showPreviousSlide);  nextButton.addEventListener('click', showNextSlide);  function showPreviousSlide() {      slideIndex = (slideIndex - 1 + slideCount) % slideCount;      updateSlider();  }  function showNextSlide() {      slideIndex = (slideIndex + 1) % slideCount;      updateSlider();  }  function updateSlider() {      slides.forEach((slide, index) => {          if (index === slideIndex) {              slide.style.display = 'block';          } else {              slide.style.display = 'none';          }      });  }  updateSlider(); |

На рисунке 10 представлен блок «Наши условия». Поскольку сведения об охраняемых объектах являются конфиденциальными, кандидаты на трудоустройство могут узнать только общую информацию: ставки по объектам, наличие питания, график работы и предполагаемую зарплату.



Рисунок 10. Блок «Наши условия»

Для реализации разметки в этом блоке был использован код, который представлен ниже:

|  |
| --- |
| <?php require\_once ‘./functions/connect.php’;?>  <?php      $sql2 = $pdo->prepare(“SELECT \* FROM Services”);      $sql2->execute();      $uslugi=$sql2->fetchAll(PDO::FETCH\_OBJ);  ?>  <section id=”services”>    <h1 class=”title”>Наши условия</h1>    <div class=”container1”>    <div class=”pricing-table”>      <table>        <tr>          <th>Ставки по объектам</th>          <th>Питание</th>          <th>График</th>          <th>Месячная зарплата</th>        </tr>        <?php foreach($uslugi as $usluga):?>        <tr>          <td><?php echo $usluga->Amount?> рублей</td>          <td><?php echo $usluga->Chart?></td>          <td><?php echo $usluga->Conditions?></td>          <td><?php echo $usluga->total\_amount?> рублей</td>        </tr>        <?php endforeach ?>      </table>    </div>    </div>  </section> |

На рисунке 11 представлен блок «Наши преимущества», где описаны основные преимущества и добавлены изображения небольшого размера для улучшения восприятия.

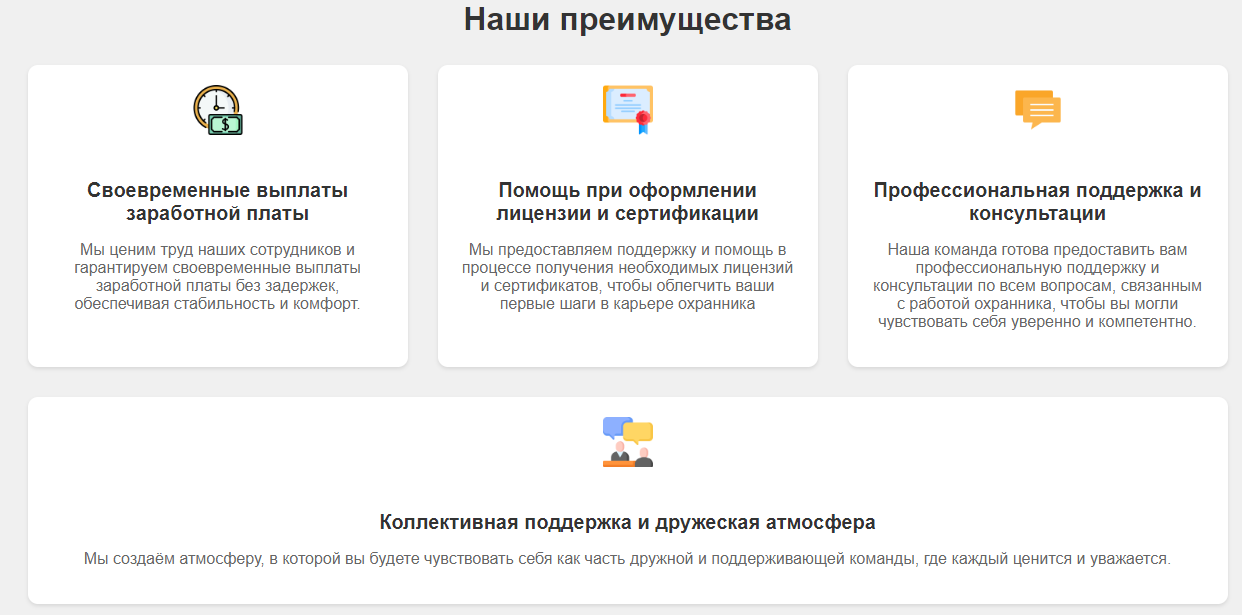


Рисунок 11. Блок «Наши преимущества»

Для реализации разметки в этом блоке был использован код, который представлен ниже:

|  |
| --- |
| <?php require\_once ‘./functions/connect.php’;?>  <?php      $sql2 = $pdo->prepare(“SELECT \* FROM Advantages”);      $sql2->execute();      $advantages=$sql2->fetchAll(PDO::FETCH\_OBJ);  ?>  <section id=”advantages”>      <h1 class=”title”>Наши преимущества</h1>      <div class=”advantages-section”>      <?php foreach($advantages as $advantage):?>          <div class=”advantage”>            <img class=”advantage-icon” src=”admin/advantages/img/<?php echo $advantage->Filename?>”>            <h3 class=”advantage-title”><?php echo $advantage->Name?> <br></h3>            <p class=”advantage-description”> <?php echo $advantage->Description?></p>          </div>          <?php endforeach ?>      </div>  </section> |

На рисунке 12 представлен блок «Лицензии», где отображается фотография лицензии, которая подтверждает, что компания законно осуществляет частную охранную деятельность.



Рисунок 12. Блок «Лицензии»

Для реализации разметки в этом блоке был использован код, который представлен ниже:

|  |
| --- |
| <?php require\_once './functions/connect.php';?>    <?php      $main = $pdo->prepare("SELECT \* FROM License");      $main->execute();      $uslugi=$main->fetch(PDO::FETCH\_OBJ);  ?>  <section id="licenses">      <h1 class="title">Лицензии</h1>      <div class="documents-container">          <div class="document-item">            <img src="photo/<?php echo $uslugi->Filename?>" alt="Document Image" class="document-image">          </div>      </div>  </section> |

На рисунке 13 представлен блок «Контактные данные», где указан адрес офиса, ближайшее метро, телефон, почта и график работы. Также включена интерактивная карта, которая позволяет клиентам построить маршрут от своей точки до офиса компании.

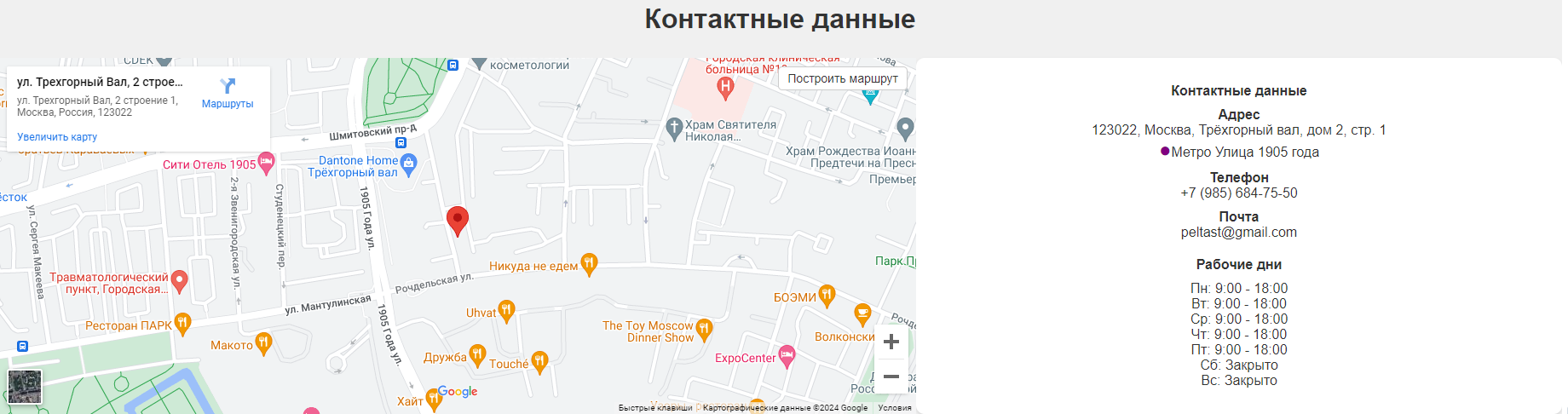


Рисунок 13. Блок «Контактные данные»

Для реализации разметки в этом блоке был использован код, который представлен ниже:

|  |
| --- |
| <?php require\_once './functions/connect.php';?>    <?php      $main = $pdo->prepare("SELECT \* FROM Contact");      $main->execute();      $uslugi=$main->fetch(PDO::FETCH\_OBJ);  ?>  <section id="contacts">      <h1 class="title">Контактные данные</h1>      <div class="container">        <div class="map-container">          <div class="map-buttons">              <a href="https://www.google.com/maps/dir/Current+Location/55.75754867308599,37.558065576812304" target="\_blank">Построить маршрут</a>          </div>          <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2245.0548904696866!2d37.558065576812304!3d55.75754867308599!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x46b54bd272b15555%3A0x16a17abf1f455823!2z0YPQuy4g0KLRgNC10YXQs9C-0YDQvdGL0Lkg0JLQsNC7LCAyINGB0YLRgNC-0LXQvdC40LUgMSwg0JzQvtGB0LrQstCwLCDQoNC-0YHRgdC40Y8sIDEyMzAyMg!5e0!3m2!1sru!2sus!4v1711477248759!5m2!1sru!2sus" width="100%" height="100%" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>        </div>          <div class="info-container">            <p><strong>Контактные данные</strong></p>            <p><strong>Адрес</strong><br>            123022, Москва, Трёхгорный вал, дом 2, стр. 1<br>            <span style="color: purple; font-size: 24px;">&#9679;</span>Метро Улица 1905 года</p>            <p><strong>Телефон</strong><br>            <?php echo $uslugi->Phone?></p>            <p><strong>Почта</strong><br>            <?php echo $uslugi->Email?></p>            <div class="hours">              <p><strong>Рабочие дни </strong></p>              <p>Пн: 9:00 - 18:00<br>              Вт: 9:00 - 18:00<br>              Ср: 9:00 - 18:00<br>              Чт: 9:00 - 18:00<br>              Пт: 9:00 - 18:00<br>              Сб: Закрыто<br>              Вс: Закрыто</p>            </div>          </div>      </div>  </section> |

На рисунке 14 представлен блок «Остались вопросы?», где отображается два поля, с именем, номером телефона и кнопка отправить.



Рисунок 14. Блок «Остались вопросы»

Для реализации разметки в этом блоке был использован код, который представлен ниже:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Форма с CAPTCHA</title>  <link rel="stylesheet" href="css/captcha.css">  </head>  <body>  <footer>      <h1 class="title">Остались вопросы?</h1>      <div class="question-form-container">          <form id="contact-form" action="process.php" method="post">              <p>Отправьте заявку и получите консультацию в течение 10 минут</p>              <input type="text" id="name" name="name" placeholder="Ваше имя" required>              <input type="tel" id="phone" name="phone" placeholder="Ваш телефон" required>              <button type="button" onclick="validateAndSubmit()">Отправить</button>              <input type="hidden" id="captcha" name="captcha" value="">          </form>      </div>  </footer>  <script src="/js/check.js"></script>  <div id="captcha-modal" class="modal">      <div class="modal-content">          <span class="close" onclick="closeCaptchaModal()">&times;</span>          <p>Введите код CAPTCHA:</p>          <input type="text" id="captcha-input" placeholder="Введите код с картинки" required>          <img id="modal-captcha-image" src="" alt="CAPTCHA">          <button type="button" class="confirm-button" onclick="submitForm()">Подтвердить</button>      </div>  </div>  <script src="js/captcha.js"></script>  </body>  </html> |

Для реализации проверок был использован следующий код на JavaScript:

|  |
| --- |
| function validateAndSubmit() {      var name = document.getElementById('name').value.trim();      var phone = document.getElementById('phone').value.trim();      if (name === '' && phone === ''){          alert('Пожалуйста, заполните поле ваше имя и номер телефона.')      }      else if (name === '') {          alert('Пожалуйста, заполните поле ваше имя.');          return;      }      else if (phone === '') {          alert('Пожалуйста, введите номер телефона.');          return;      }      else if (phone.length !== 18) {          alert('Пожалуйста, введите полный номер телефона.');          return;      }      showCaptchaModal();  } |

Также, у телефона есть маска, которая реализована на JavaScript. Ниже представлен код её реализации:

|  |
| --- |
| document.getElementById('phone').addEventListener('input', function (e) {      var input = e.target.value.replace(/\D/g, '');      if (input.startsWith('8')) {          input = '7' + input.slice(1);      }      input = input.substring(0, 11);      var formatted = '+7';      if (input.length > 1) {          formatted += ' (' + input.substring(1, 4);      }      if (input.length >= 5) {          formatted += ') ' + input.substring(4, 7);      }      if (input.length >= 8) {          formatted += '-' + input.substring(7, 9);      }      if (input.length >= 10) {          formatted += '-' + input.substring(9, 11);      }      e.target.value = formatted;  }); |

Далее обрабатывается функция отображения, проверки правильности ввода «captcha» и её закрытия. В случае правильности ввода капчи, страница обновляется. Ниже представлен код реализации этого на JavaScript:

|  |
| --- |
| function showCaptchaModal() {      var modal = document.getElementById("captcha-modal");      modal.style.display = "block";      var xhttp = new XMLHttpRequest();      xhttp.onreadystatechange = function() {          if (this.readyState == 4 && this.status == 200) {              document.getElementById("modal-captcha-image").src = "captcha.php";          }      };      xhttp.open("GET", "captcha.php", true);      xhttp.send();  }  function closeCaptchaModal() {      document.getElementById("captcha-modal").style.display = "none";  }  function submitForm() {      var captchaInput = document.getElementById("captcha-input").value;      document.getElementById("captcha").value = captchaInput;      var xhttp = new XMLHttpRequest();      xhttp.onreadystatechange = function() {          if (this.readyState == 4 && this.status == 200) {              if (this.responseText === "true") {                  alert("Ваша заявка успешно отправлена!");                  refreshPage();              } else {                  alert("Неправильный код. Пожалуйста, попробуйте еще раз.");              }          }      };      xhttp.open("POST", "process.php", true);      xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");      xhttp.send("name=" + encodeURIComponent(document.getElementById("name").value) + "&phone=" + encodeURIComponent(document.getElementById("phone").value) + "&captcha=" + encodeURIComponent(captchaInput));  }  function refreshPage() {      window.location.reload();  } |

Ниже представлен код, где генерируется изображение с кодом «captcha»:

|  |
| --- |
| <?php  session\_start();  $randomNumber = rand(1000, 9999);  $\_SESSION['captcha'] = $randomNumber;  $captchaImage = imagecreate(100, 30);  $bgColor = imagecolorallocate($captchaImage, 255, 255, 255);  $textColor = imagecolorallocate($captchaImage, 0, 0, 0);  imagestring($captchaImage, 5, 10, 5, $randomNumber, $textColor);  header("Content-type: image/png");  imagepng($captchaImage);  imagedestroy($captchaImage); |

В случае, если «captcha» ведена правильно, то она отправляется в базу данных. Ниже представлен код, как это реализуется:

|  |
| --- |
| <?php  session\_start();  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {      if ($\_POST['captcha'] != $\_SESSION['captcha']) {          echo "Неправильный код CAPTCHA. Пожалуйста, попробуйте еще раз.";          exit();      }      $name = $\_POST['name'];      $phone = $\_POST['phone'];      $sql = "INSERT INTO Application (name, phone) VALUES (:name, :phone)";      $stmt = $pdo->prepare($sql);      $stmt->bindParam(':name', $name);      $stmt->bindParam(':phone', $phone);      if ($stmt->execute()) {          echo "true";      } else {          echo "Произошла ошибка при отправке заявки. Пожалуйста, попробуйте еще раз.";      }  } |

Далее был сделан стиль и адаптивность под разные устройства (телефоны, планшеты и настольные компьютеры) для всей страницы. Для реализации этого был использован следующий код (см. Приложение №2).

У администратора имеется свой «личный кабинет». Для того, чтобы попасть в него, нужно в строке со ссылкой на сайт ввести «/login.php». После этого, он попадает на страницу авторизации, которая представлена на рисунке 15.

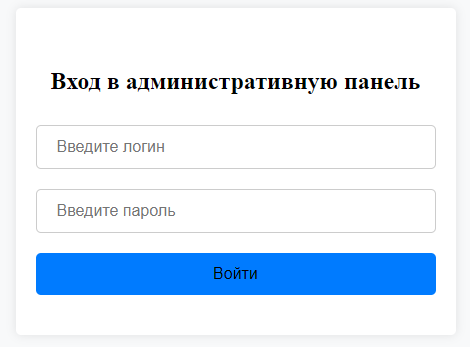
****

Рисунок 15. Авторизация

Перед тем как пользователь зайдёт в административную панель, ему нужно авторизоваться.

Для реализации формы авторизации был использован следующий код:

|  |
| --- |
| <body>      <div class="login-container">          <h2 class="login-title">Вход в административную панель</h2>          <form id="loginForm" action="admin/admin.php" method="post" class="login-form" onsubmit="return validateForm()">              <div class="form-group">                  <input type="text" class="form-control" placeholder="Введите логин" name="login" id="login">              </div>              <div class="form-group">                  <input type="password" class="form-control" placeholder="Введите пароль" name="password" id="password">              </div>              <button type="submit" class="btn btn-primary btn-block">Войти</button>          </form>      </div>  </body> |

Для реализации проверки данных для входа пользователя на страницу используется следующий код, представленный ниже:

|  |
| --- |
| <?php  session\_start();  require\_once '../functions/connect.php';  $login = $\_POST["login"];  $password = $\_POST["password"];  $hashedPassword = hash('sha256', $password);  $sql = $pdo->prepare("SELECT ID, login, password FROM User WHERE login = :login AND password = :password");  $sql->execute(array('login' => $login, 'password' => $hashedPassword));  $array = $sql->fetch(PDO::FETCH\_ASSOC);  if ($array["ID"] > 0) {      $\_SESSION['login'] = $array['login'];      header('Location:/admin.php');      exit();  } else {      header('Location:/login.php');      exit();  } |

Данный код получает данные для входа (логин и пароль) из POST-запроса, «хеширует» введённый пароль и проверяет наличие пользователя в базе данных с указанными логином и «хешированным» паролем.

При успешной авторизации открывается главная страница администратора, где отображаются ссылки на блоки, которые можно редактировать. Эта страница представлена на рисунке 16.

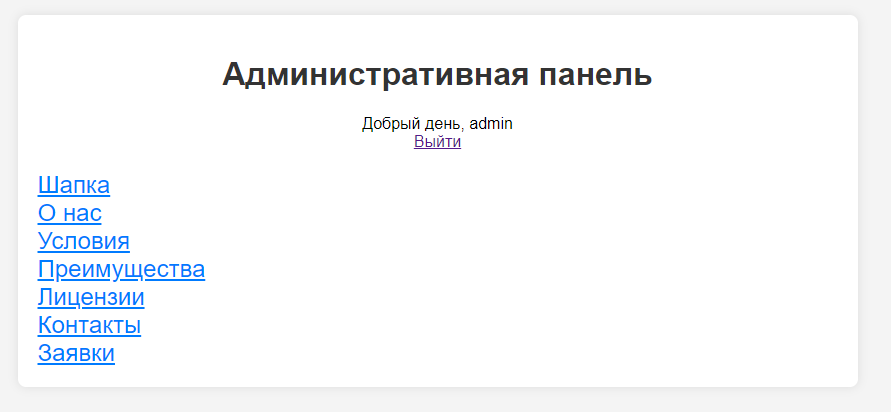


Рисунок 16. Административная панель

Код реализации этой страницы представлен в Приложении 3.

На каждой странице есть гиперссылка, которая называется «Выйти», при нажатии на неё администратор попадает на главную страницу сайта.

Ниже представлен код реализации этой гиперссылки:

|  |
| --- |
| <?php  session\_start();  unset($\_SESSION['login']);  header('Location:/index.php'); |

При переходе в шапку сайта, администратор может изменить название организации, её логотип и номер телефона. Эта страница представлена на рисунке 17.

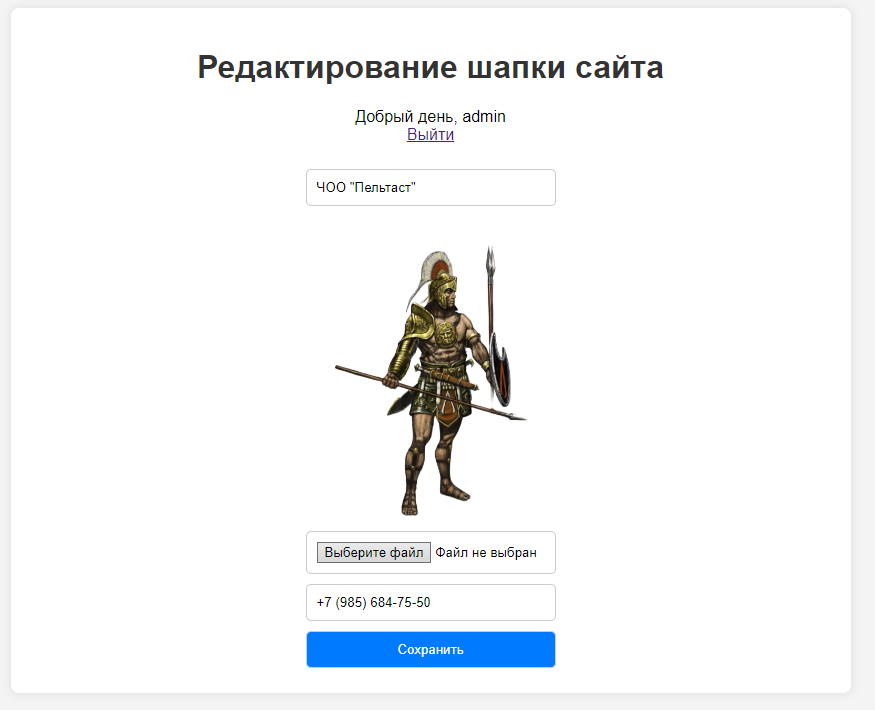


Рисунок 17. Редактирование шапки сайта

Для реализации формы этой страницы был использован следующий код, который представлен в Приложении 3.

Также, была создана отдельная страница, где обрабатывается редактирование данных на сайте. Везде, где добавляются фотографии, были сделаны проверки, такие как на расширение файла и на размер. На сайте есть два места, где отображается номер телефона, если изменить в одном месте, то в другом он тоже изменится.

Код реализации этой страницы представлен в Приложении №4.

Везде, где добавляются фотографии, были сделаны проверки, такие как на расширение файла и на размер.

Ниже представлен код реализации этих проверок:

|  |
| --- |
| if(isset($\_FILES['Filename']) && $\_FILES['Filename']['error'] !== UPLOAD\_ERR\_NO\_FILE) {          $list = ['.php', '.zip', '.js', '.html'];          $extension = strtolower(pathinfo($\_FILES['Filename']['name'], PATHINFO\_EXTENSION));          if (in\_array("." . $extension, $list)) {              exit("Расширение файла не подходит");          }          $type = getimagesize($\_FILES['Filename']['tmp\_name']);          if (!$type || ($type['mime'] != 'image/png' && $type['mime'] != 'image/jpeg')) {              exit("Тип файла не подходит, можно загрузить только .png, .jpg, .jpeg");          }          if ($\_FILES['Filename']['size'] > 50 \* 1024 \* 1000) {              exit("Размер файла превышен, он не должен превышать 50 мб");          }          $upload = 'img/' . $\_FILES['Filename']['name'];          if (!move\_uploaded\_file($\_FILES['Filename']['tmp\_name'], $upload)) {              exit("Ошибка при загрузке файла");          }  } |

Администратору доступны для редактирования и другие блоки. При переходе на страницу «О нас», он может редактировать описание, а также может добавлять, изменять и удалять фотографии в слайдере. Была реализована функция, чтобы при добавлении фотографий, они были одного размера, чтобы кнопки перелистывания слайдера оставались на месте. Эта страница представлена на рисунке 18.

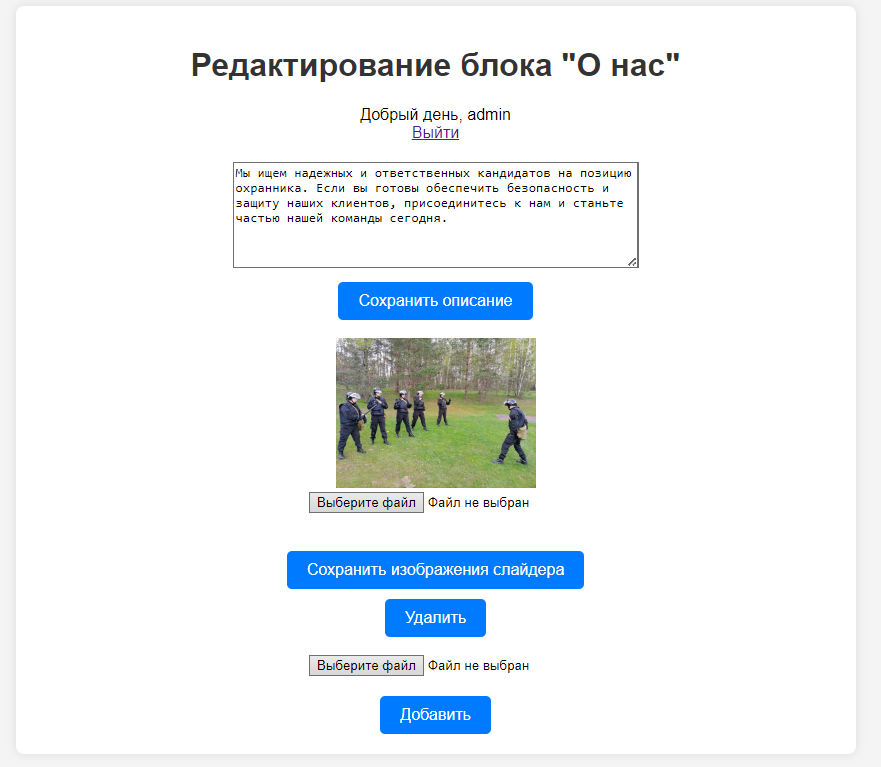


Рисунок 18. Редактирование блока «О нас»

Код формы этой страницы представлен в Приложении №3.

Затем было реализовано редактирование описание, код которого представлен ниже:

|  |
| --- |
| <?  if (isset($\_POST["save\_description"])) {      $description = $\_POST["Description"];      $sql = "UPDATE About SET Description=:Description";      $query = $pdo->prepare($sql);      $query->execute(["Description" => $description]);  }  echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/about.php">'; |

Для добавления, редактирования и удаления фотографий из слайдера, был реализован следующий код, который представлен в Приложении №4.

При переходе на страницу «Условия», администратор может редактировать ставку по объектом, наличие питания, график работы и общую сумму в месяц. Эта страница представлена на рисунке 19.

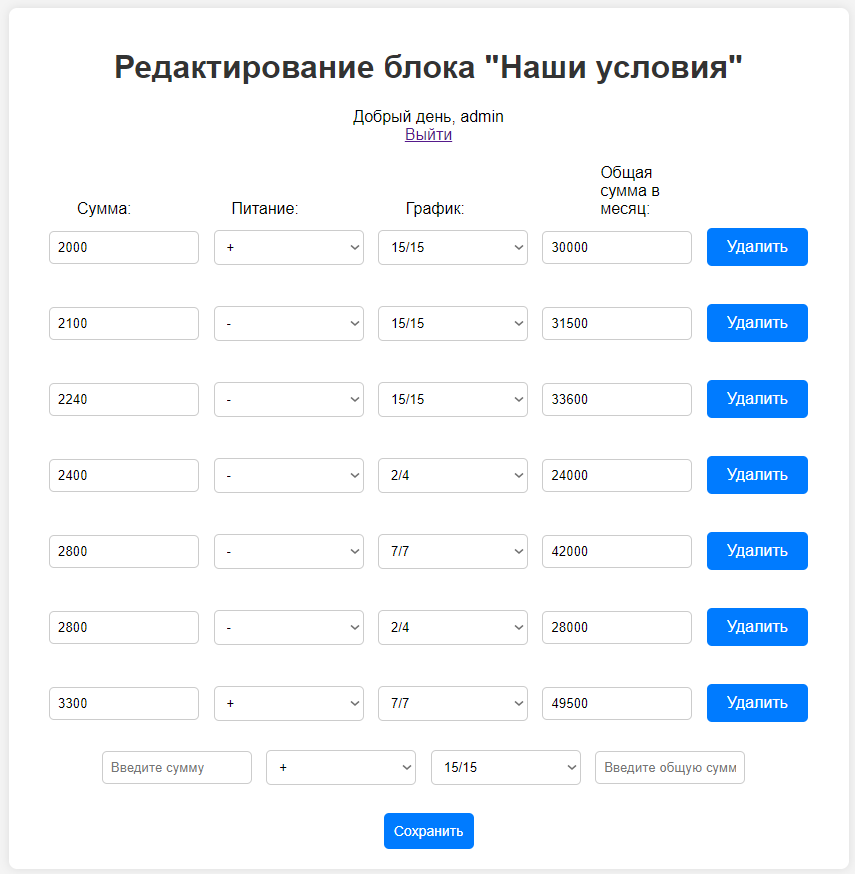


Рисунок 19. Редактирование блока «Наши условия»

Код формы этой страницы представлен в Приложении №3.

Для добавления, редактирования и удаления условий, был реализован следующий код, который представлен в Приложении №4.

Также была реализована функция, которая считает общую сумму исходя из ставки по объектам и графику, код которой представлен ниже:

|  |
| --- |
| function calculateTotalAmount(amountField, conditionsField, totalField) {      const rate = parseFloat(amountField.value) || 0;      const conditions = conditionsField.value;      const daysInMonth = 30;      let workDays = 0;      if (conditions === '15/15') {          workDays = 15;      } else if (conditions === '7/7') {          workDays = Math.floor(daysInMonth / 2);      } else if (conditions === '2/4') {          workDays = Math.floor(daysInMonth / 3);      } else if (conditions === '1/3') {          workDays = Math.floor(daysInMonth / 4);      }      totalField.value = (rate \* workDays).toFixed(2);  }  document.addEventListener('DOMContentLoaded', () => {      const formGroups = document.querySelectorAll('.form-group');      formGroups.forEach(group => {          const amountField = group.querySelector('input[name="Amount[]"], input[name="new\_Amount[]"]');          const chartField = group.querySelector('select[name="Chart[]"], select[name="new\_Chart[]"]');          const conditionsField = group.querySelector('select[name="Conditions[]"], select[name="new\_Conditions[]"]');          const totalField = group.querySelector('input[name="total\_amount[]"], input[name="new\_total\_amount[]"]');          if (amountField && conditionsField && totalField) {              const updateTotal = () => calculateTotalAmount(amountField, conditionsField, totalField);              amountField.addEventListener('input', updateTotal);              conditionsField.addEventListener('change', updateTotal);          }      });  }); |

При переходе на страницу «Преимущества», администратор может редактировать основные преимущества и прикреплять к ним маленький фотографии. Эта страница представлена на рисунке 20.

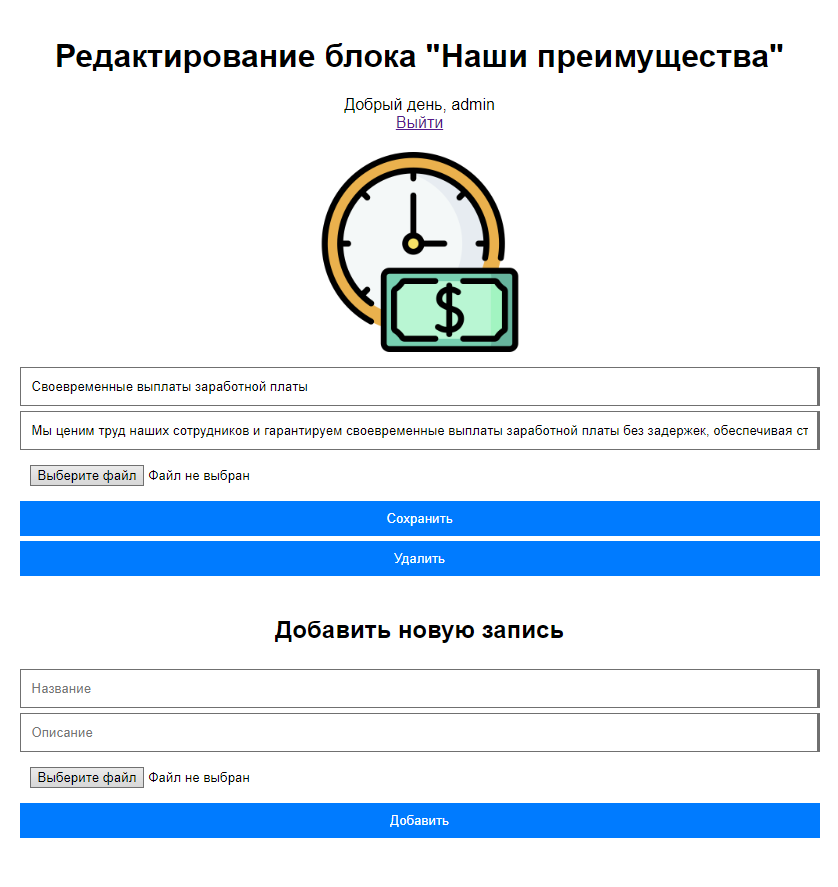


Рисунок 20. Редактирование блока «Наши преимущества»

Код формы этой страницы представлен в Приложении №3.

Для добавления, редактирования и удаления преимуществ, был реализован следующий код, который представлен в Приложении №4.

При переходе на страницу «Лицензии», администратор может обновлять фотографию лицензии. Эта страница представлена на рисунке 21.

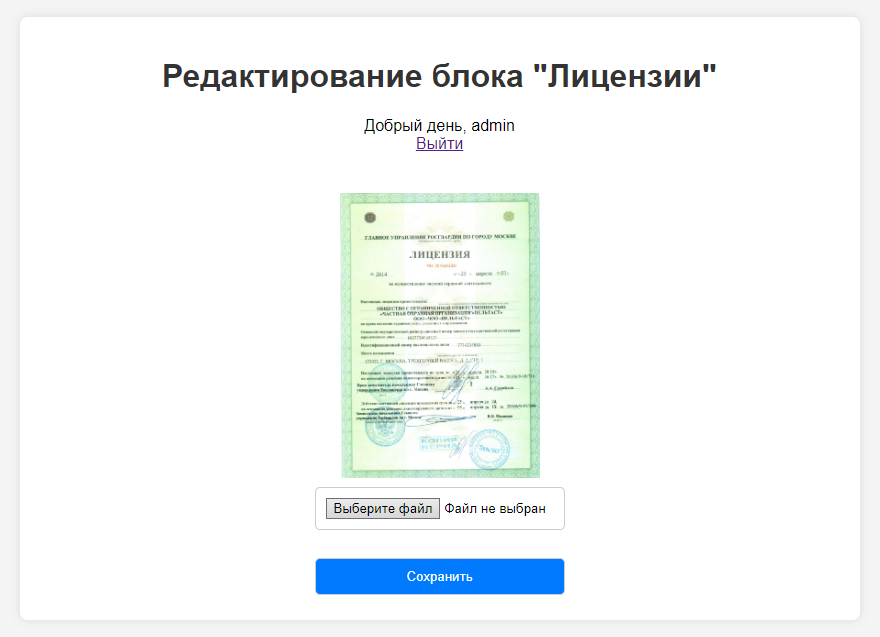


Рисунок 21. Редактирование блок «Лицензии»

Код формы этой страницы представлен в Приложении №3.

Для обновления фотографии был реализован код, который представлен в Приложении №4.

При переходе на страницу «Контакты», администратор может обновлять телефон и почту. Эта страница представлена на рисунке 22.

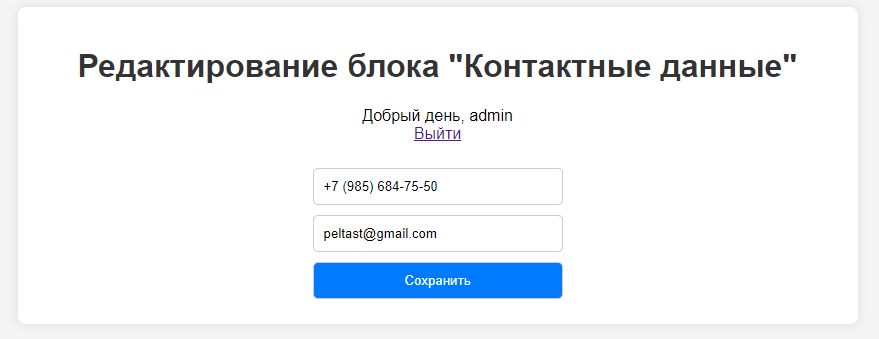


Рисунок 22. Редактирование блока «Контактные данные»

Код реализации формы этой страницы представлен в Приложении №3.

Для обновления телефона и почты, был реализован следующий код:

|  |
| --- |
| if (isset($\_POST["save"])) {      $phone = $\_POST["Phone"];      $email = $\_POST["Email"];      $sql = "UPDATE Contact SET Phone=:Phone, Email=:Email";      $query = $pdo->prepare($sql);      $query->execute(["Phone" => $phone, "Email" => $email]);      $sqlContact = "UPDATE Header SET Phone=:Phone";      $queryContact = $pdo->prepare($sqlContact);      $queryContact->execute(["Phone" => $phone]);  }  echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/contact.php">'; |

При переходе на страницу «Заявки», администратор может просматривать телефоны и имена, а также их удалять. Эта страница представлена на рисунке 23.

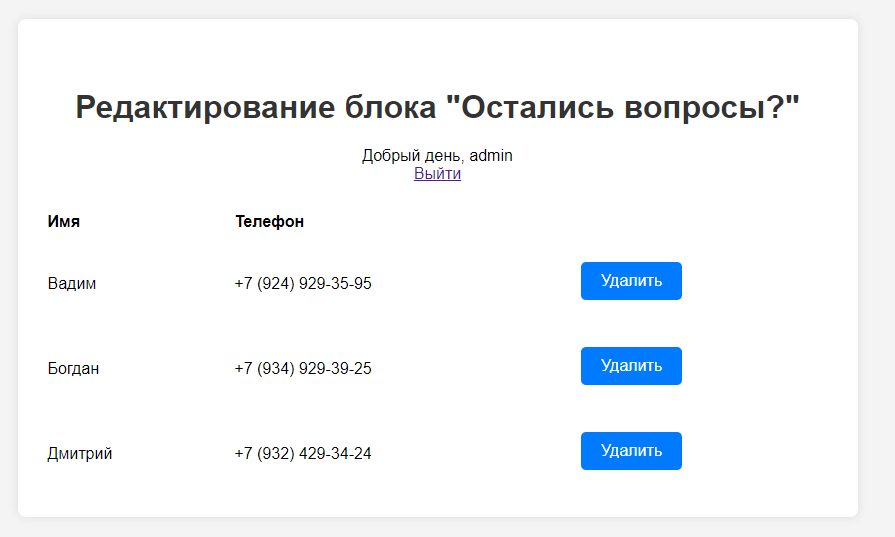


Рисунок 23. Редактирование блока «Остались вопросы?»

Код формы этой страницы представлен в Приложении №3.

Для удаления заявок, был реализован следующий код:

|  |
| --- |
| if(isset($\_POST["delete"])) {      $idToDelete = $\_POST["delete"];      $sql = "DELETE FROM Application WHERE ID=:ID";      $query = $pdo->prepare($sql);      $query->execute(["ID" => $idToDelete]);      echo "Запрос на удаление выполнен для ID: " . $idToDelete;  }  echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/application.php">'; |

Также был реализован запрос, который обновляет страницу в случае, если появилась новая заявка, код этого представлен ниже:

|  |
| --- |
| $sql = $pdo->query("SELECT COUNT(\*) as count FROM Application");  $count = $sql->fetchColumn();  if ($count > 0) {      echo '<meta http-equiv="refresh" content="5">';  } |

Для страниц в административной странице был реализован единый стиль, фрагмент кода которого представлен в Приложении №2.

После изменения информации при нажатии на кнопку назад в браузере, будет высвечиваться информация, которая была изменена. Для того, чтобы этого избежать, на каждой странице была реализована гиперссылка «Назад», которая сразу ведёт нас на главную страницу. Она представлена на рисунке 24.



Рисунок 24. Гиперссылка назад

Ниже представлен код, как она работает:

|  |
| --- |
| <a href="/admin.php" class="back-link">Назад</a> |

При выходе из аккаунта администратора и вставки ссылки на страницу в админ панели на странице сайта будет написано «Невозможно зайти таким способом. Это представлено на рисунке 25.

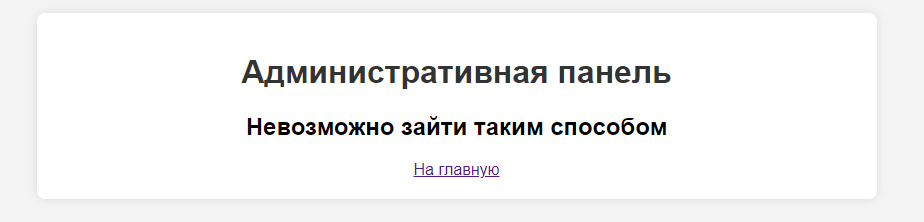


Рисунок 25. Переход по ссылке без авторизации администратора

Ниже представлен фрагмент кода реализации:

|  |
| --- |
| <?php if(!empty($\_SESSION["login"])) :?>  Основной код  <?php else:          echo '<h2>Невозможно зайти таким способом</h2>';          echo '<a href="/">На главную</a>';  ?> |

# **2.4. Отладка и тестирование программного обеспечения**

На рисунке 26 показан интерфейс навигации разделов на сайте.



Рисунок 26. Навигация сайта

При нажатии на разделы: «О нас», «Условия», «Преимущества», «Лицензии», «Контакты» выполняется переход пользователя на ту секцию, которую он выбрал.

На рисунке 27 показаны 2 кнопки, которые работают также, как и навигация.

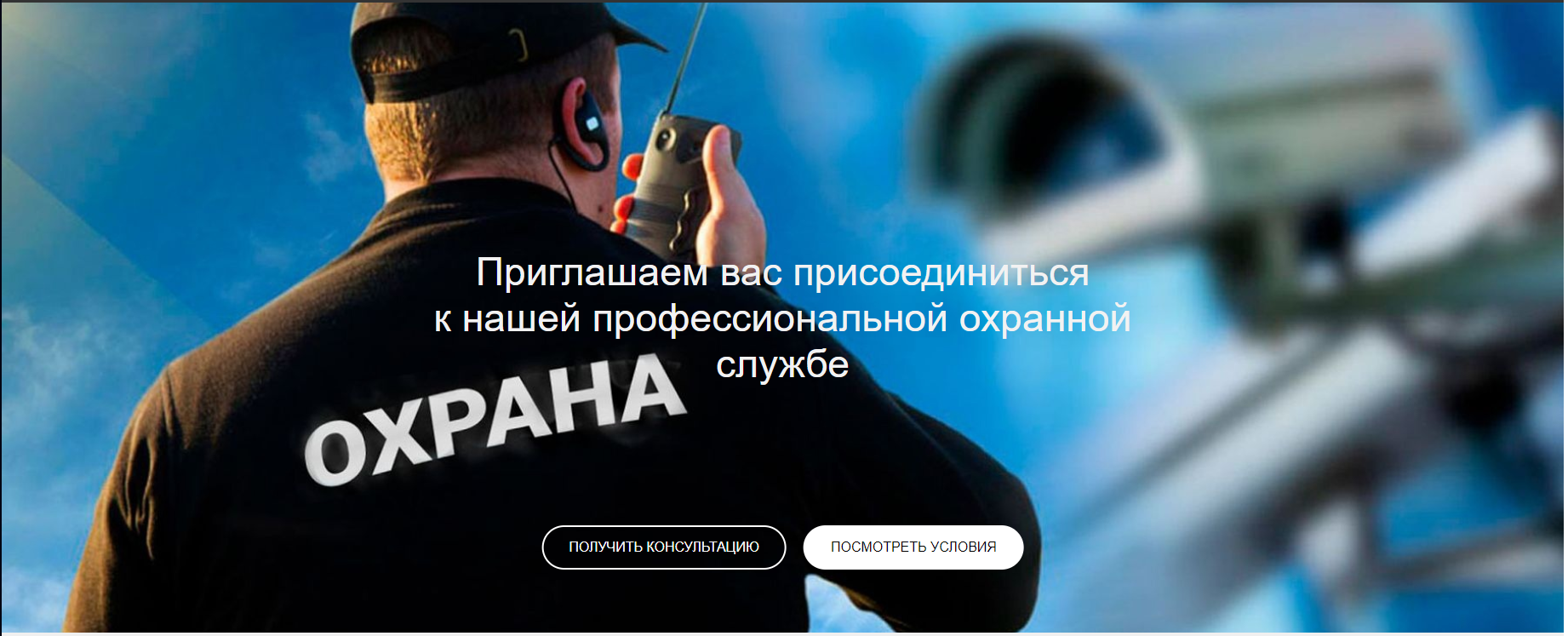


Рисунок 27. Навигация сайта

На рисунках 28-30 показано тестирование слайдера в разделе «О нас», который можно переключать при нажатии на кнопки по бокам вперёд и назад.



Рисунок 28. Тестирование слайдера

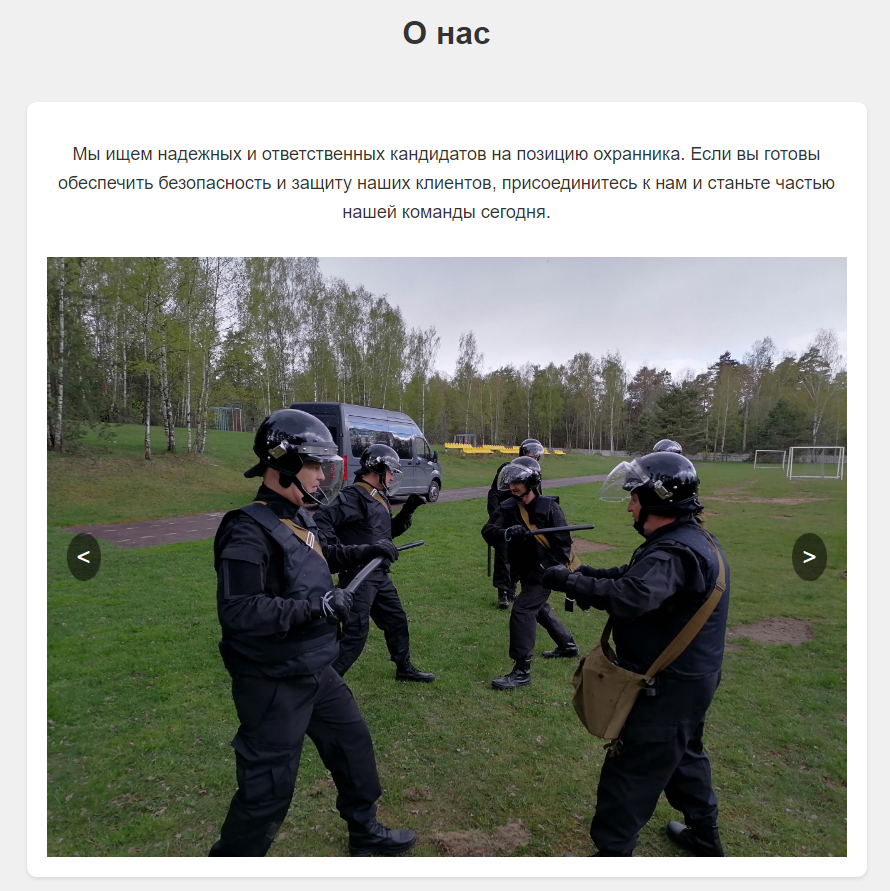


Рисунок 29. Тестирование слайдера

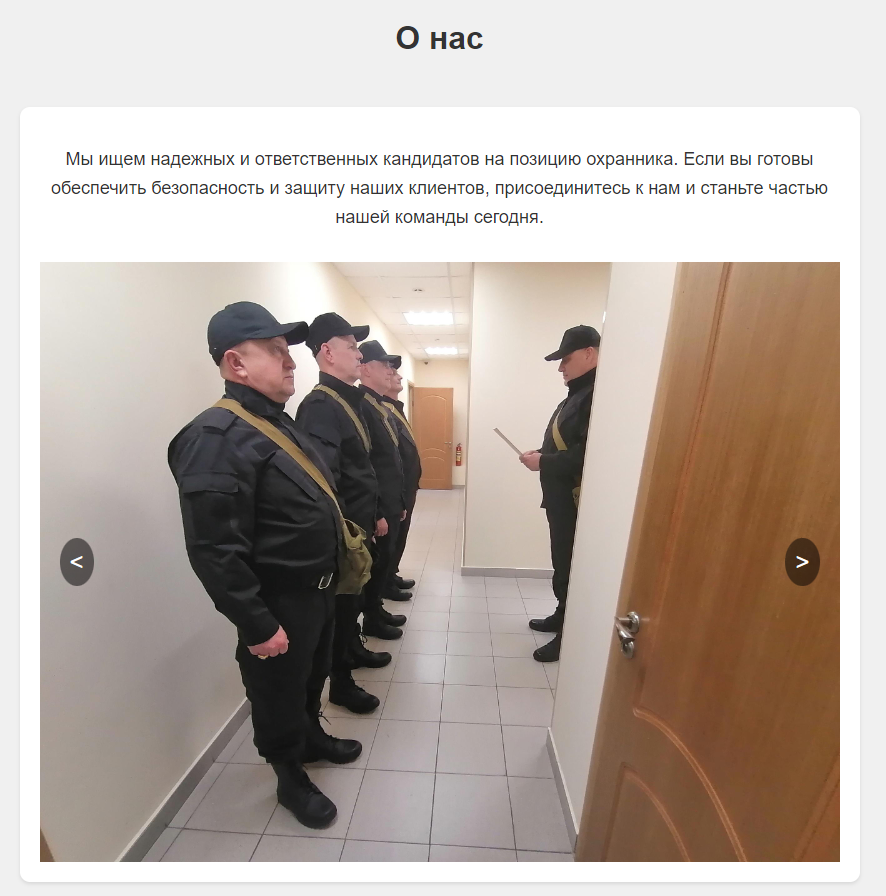


Рисунок 30. Тестирование слайдера

В таблице 2 представлены результаты отладки и тестирования веб-сайта охранного предприятия блока «Остались вопросы?».

Таблица 2. Отладка и тестирование блока «Остались вопросы?»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  теста | Входные данные | Вводимое значение | Ожидаемая реакция программы | Фактическая реакция программы | Ошибка выявлена |
| 1 | имя и телефон |  | Сообщение «Пожалуйста, заполните поле ваше имя и телефон.» | На рисунке 31 | нет |
| 2 | имя | Андрей | Сообщение «Пожалуйста, введите номер телефона.» | На рисунке 32 | нет |
| 3 | имя и телефон | Андрей, +7 (977) 232 | Сообщение «Пожалуйста, введите полный номер телефона.» | На рисунке 33 | нет |
| 4 | Имя и телефон | Андрей, +7 (977) 232-97-97 | Открытие модального окна с капчей | На рисунке 34 | нет |
| 5 | капча | 2292 | Сообщение «Неправильный код. Пожалуйста, попробуйте ещё раз» | На рисунке 35 | нет |
| 6 | капча | 2290 | Сообщение «Ваша заявка успешно отправлена» | На рисунке 36 | нет |

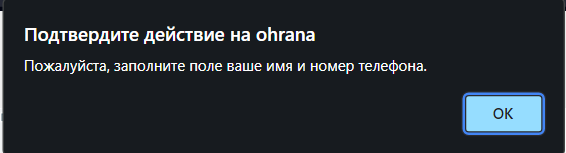
****

Рисунок 31. Результат теста 1

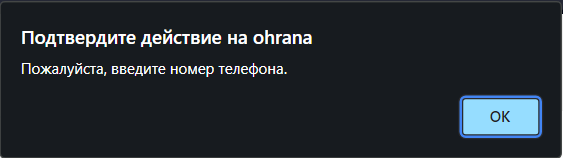


Рисунок 32. Результат теста 2

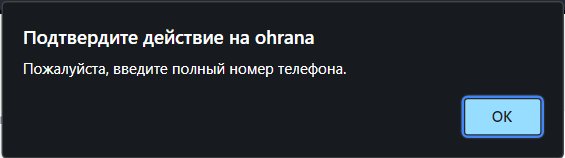


Рисунок 33. Результат теста 3

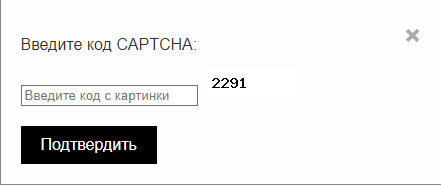


Рисунок 34. Результат теста 4



Рисунок 35. Результат теста 5

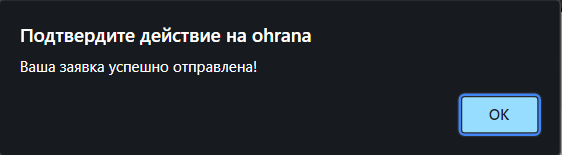


Рисунок 36. Результат теста 6

В таблице 3 представлены результаты отладки и тестирования веб-сайта охранного предприятия на странице авторизации.

Таблица 3. Отладка и тестирование авторизации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  теста | Входные данные | Вводимое значение | Ожидаемая реакция программы | Фактическая реакция программы | Ошибка выявлена |
| 1 | логин и пароль |  | Перезагрузка страницы | На рисунке 37 | нет |
| 2 | логин и пароль | Admin, admin777 | Открытие административной панели | На рисунке 38 | нет |

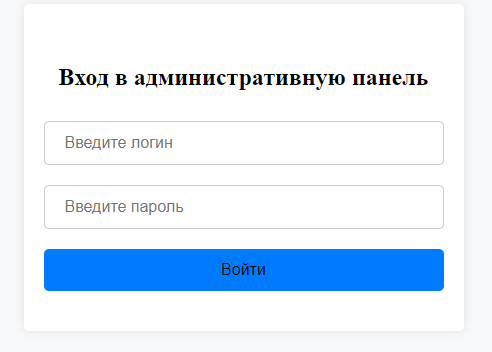


Рисунок 37. Перезагрузка страницы

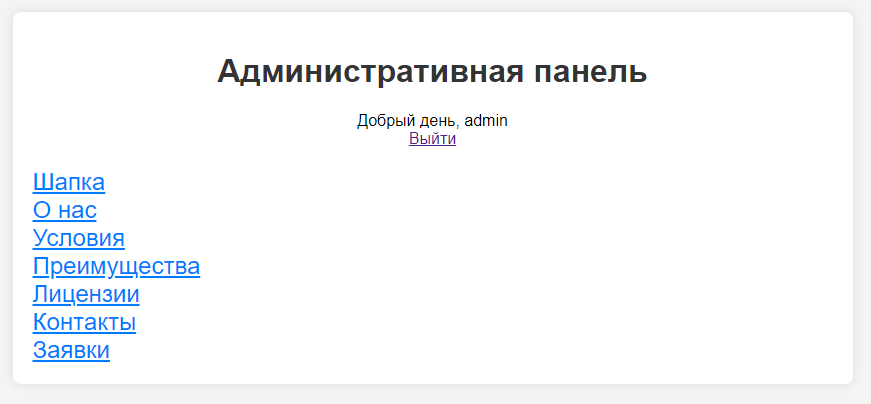


Рисунок 38. Правильные входные данные

В таблице 4 представлены результаты отладки и тестирования веб-сайта охранного предприятия на добавление фотографий.

Таблица 4. Отладка и тестирование добавления фотографий

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  теста | Входные данные | Вводимое значение | Ожидаемая реакция программы | Фактическая реакция программы | Ошибка выявлена |
| 1 | файл | .doc | Тип файла не подходит, можно загрузить только .png, .jpg, .jpeg | На рисунке 39 | нет |

Продолжение таблицы 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  теста | Входные данные | Вводимое значение | Ожидаемая реакция программы | Фактическая реакция программы | Ошибка выявлена |
| 2 | фотография | 62 мб | Размер файла превышен, он не должен превышать 50 мб | На рисунке 40 | нет |

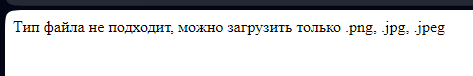


Рисунок 39. Файл .doc



Рисунок 40. Фотография 62 мб

# **2.5. Руководство по использованию программы**

Сайт охранного предприятия разработан для предоставления информации о компании. Пользователь должен иметь следующие минимальные системные требования:

* веб-браузер с поддержкой HTML и CSS;
* подключение к интернету.

Основные разделы сайта включают:

* шапка: здесь представлен логотип компании, её название, номер телефона и график работы;
* раздел «О нас»: раздел, где есть объявление о наборе и слайдер с фотографиями, которые показывают деятельность компании;
* условия: раздел, где описаны предоставляемые условия, такие как ставки по объектам, наличие питания, график работы и заработная плата в месяц;
* преимущества: здесь представлены основные преимущества компании;
* лицензии: раздел, где представлена фотография лицензии;
* контактные данные: раздел, где можно найти карту с отмеченным адресом офиса компании, а также кнопку для построения маршрута от вашей точки. Кроме того, здесь указаны адрес офиса, контактный телефон, адрес электронной почты и график работы.

Настройка использования сайта включает функцию обратной связи, где пользователь может указать свое имя для более персонализированного обращения, а также оставить номер телефона. В течение 10 минут после отправки сообщения через форму обратной связи, сотрудник свяжется по указанному номеру.

# **ЗАКЛЮЧЕНИЕ**

В ходе разработки веб-сайта для охранного предприятия решены ключевые задачи, определенные в функциональных и эксплуатационных требованиях.

Выполненные задачи:

* входные данные успешно интегрированы, включая информацию о компании, условиях и предложениях, контактных данных, а также фотографии, демонстрирующие деятельность предприятия;
* создан удобный и структурированный интерфейс сайта, который позволяет пользователям легко найти необходимую информацию о предприятии. Контактные данные представлены в доступной форме для удобства связи с компанией;
* реализованы функциональные требования, включая отображение информации о предприятии и его условиях, просмотр фотографий, а также форму обратной связи для клиентов;
* сайт адаптирован для корректного отображения на различных устройствах, таких как планшеты, смартфоны и ПК. Дизайн и контент сайта адаптированы под разные разрешения экранов;
* применены меры по защите от кибератак и взлома для обеспечения безопасности сайта и данных клиентов;
* перед запуском сайта были проведены тестирование функционала и отладка для выявления и устранения обнаруженных ошибок;
* предусмотрена техническая поддержка для пользователей, а также обновление информации на сайте по мере необходимости.

Веб-сайт охранного предприятия успешно справляется с поставленными задачами и обеспечивает пользователям удобный доступ к необходимой информации, а также эффективное взаимодействие с компанией.

Кроме того, особое внимание уделялось безопасности данных. Применяемые меры защиты от кибератак и взлома гарантируют надёжность работы сайта в целом.

Таким образом, разработанный веб-сайт охранного предприятия не только соответствует высоким стандартам функциональности и удобства использования, но и обеспечивает высокий уровень безопасности, что делает его надёжным инструментом для пользователей и компании в целом.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ**

*Законодательные и нормативные акты:*

1. ГОСТ Р 7.0.12-2011 Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила. – М.: Стандартинформ, 2012. – 61 с.
2. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2010. – 92 с.
3. ГОСТ 7.32-2017 Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: Стандартинформ, 2017. – 47 с.
4. ГОСТ 7.82-2001 Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2001. – 39 с.
5. ГОСТ Р 7.0.100-2018 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2018. – 122 с.
6. ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления. – М.: Стандартинформ, 2008. – 32 с.
7. Единая система программной документации. – М.: Стандартинформ, 2005. – 128 с.

*Учебная и научная литература:*

1. Иванова, Г.С. Технология программирования: учебник для студентов вузов обуч. по напр. «Информатика и вычислительная техника» / Г.С. Иванова. – 3-е изд., стер. – Москва: Кнорус, 2018. – 333 с.
2. Павловская, Т.А. C#. Программирование на языке высокого уровня: учебник для студентов вузов. – СПб: Питер, 2020. – 432 с.
3. Перлова, О.Н., Ляпина, О.П., Гусева, А.В. Проектирование и разработка информационных систем: учебник. – 2-е изд, стер. – М.: Издательский центр «Академия», 2018. – 256 с.
4. Федорова, Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: Учебное пособие / Г.Н. Федорова. – М.: КУРС: ИНФРА-М, 2019. – 336 с. – (Среднее профессиональное образование).
5. Фуфаев, Д.Э., Фуфаев, Э.В., Разработка и эксплуатация автоматизированных информационных систем: учебное пособие для студентов учреждений среднего профессионального образования по специальности "Информатика и вычислительная техника" / Д.Э. Фуфаев, Э.В. Фуфаев. ­– 6-е изд., стер. ­­– М.: Издательский центр «Академия», 2018. – 302 с.

*Интернет-документы:*

1. Аксёнова Т.Г. Онлайн-курс по технологиям разработки программного обеспечения [Электронный ресурс]. – Режим доступа: <https://classroom.google.com/c/NTc2MzExNTI0MTY0> (дата обращения: 03.02.2024).
2. ЧОП "ФРОНТ-СБ" [Электронный ресурс]. – Режим доступа: [https://www.front-sb.ru/](https://www.front-sb.ru/%20) (дата обращения: 05.02.2024).
3. ЧОП "ЛЮБЕР" [Электронный ресурс]. – Режим доступа: <https://ohrana-luber.ru/> (дата обращения 05.02.2024).
4. ЧОО "ПРИОРИТЕТ" [Электронный ресурс]. – Режим доступа: <https://chop.expert/> (дата обращение 05.02.2024).
5. Интернет-сервис для построения схем и диаграмм Draw.io. – [Электронный ресурс]. – Режим доступа: [https://app.diagrams.net/](https://app.diagrams.net/%20) (дата обращения: 20.03.2024).
6. Интернет-сервис для построения схем и диаграмм PlantUML. – [Электронный ресурс]. – Режим доступа: <https://plantuml.com/ru/> (дата обращения: 20.03.2024).
7. Интернет-сервис для создания прототипов приложения Figma – [Электронный ресурс]. – Режим доступа: <https://goo.su/zRWwxDA> (дата обращения: 20.03.2024).
8. Справочник по HTML [Электронный ресурс]. Режим доступа: – <https://htmlbook.ru/html> (дата обращения: 25.03.2024)
9. Справочник по CSS [Электронный ресурс]. Режим доступа: – <https://htmlbook.ru/css> (дата обращения: 27.03.2024)
10. Основной учебник PHP для новичков [Электронный ресурс]. Режим доступа: <https://code.mu/> (дата обращения: 01.04.2024)
11. Практикум по Технологии разработки программных продуктов А.В. Рудакова и Г.Н. Федоровой – [Электронный ресурс]. – Режим доступа: <http://itam.irk.ru/biblioavia/catalog/1505188188_351.pdf> (дата обращения: 23.02.2023).
12. Microsoft Office – [Электронный ресурс]. – Режим доступа: [https://www.microsoft.com/ru-ru/microsoft-365/microsoft-office](https://www.microsoft.com/ru-ru/microsoft-365/microsoft-office%20) (дата обращения: 20.01.2023).

# **ПРИЛОЖЕНИЕ**

**Приложение №1**

Таблица 1. Словарь базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| **Словарь данных** | | | |
| **About** | | | |
| Ключ | Поле | Обязательное | Примечание |
| Первичный | ID | Да | Идентификатор |
|  | Description | Нет | Описание |
| **Advantages** | | | |
| Первичный | ID | Да | Идентификатор |
|  | Filename | Нет | Фотография |
|  | Name | Нет | Название |
|  | Description | Нет | Описание |
| **Application** | | | |
| Первичный | ID | Да | Идентификатор |
|  | Name | Нет | Имя |
|  | Phone | Нет | Телефон |
| **Contact** | | | |
| Первичный | ID | Да | Идентификатор |
|  | Phone | Да | Телефон |
|  | Email | Нет | Почта |
| **Header** | | | |
| Первичный | ID | Да | Идентификатор |
|  | Name | Нет | Название организации |
|  | Filename | Да | Логотип |
|  | Phone | Да | Телефон |
| **License** | | | |
| Первичный | ID | Да | Идентификатор |
|  | Filename | Да | Фотография |
| **Services** | | | |
| Первичный | ID | Да | Идентификатор |
|  | Amount | Да | Ставка по объектам |
|  | Chart | Да | Наличие питания |
|  | Conditions | Да | График работы |
|  | total\_amount | Да | Общая сумма в месяц |
| **Slider** | | | |
| Первичный | ID | Да | Идентификатор |
|  | Filename | Да | Фотография |
| **User** | | | |
| Первичный | ID | Да | Идентификатор |
|  | login | Да | Логин |
|  | password | Да | Пароль |

**Приложение №2**

Листинг 2.1 Стиль и адаптивность под различные устройства основной страницы

|  |
| --- |
| .header-container {      padding: 0 20px;  }  .Three\_blocks {      display: flex;      justify-content: space-between;      align-items: center;  }  .One\_line {      display: flex;      align-items: center;  }  .centered-content {      text-align: center;  }  .logo-block img {      width: 150px;      height: auto;      margin-bottom: 10px;  }  .contact-block h2 {      font-size: 24px;      margin: 0;  }  .title-text {      font-size: 24px;      font-weight: bold;  }  .contact-info {      font-size: 24px;      font-weight: bold;  }  nav ul {      list-style-type: none;      padding: 0;      text-align: center;  }  .navigation {      background-color: #333;      padding: 10px 0;  }  .navigation ul {      list-style-type: none;      padding: 0;      text-align: center;      display: flex;      justify-content: center;  }  .navigation ul li {      display: inline;      margin: 0 20px;  }  .navigation ul li a {      text-decoration: none;      color: #fff;      font-size: 18px;      transition: color 0.3s ease;      padding: 5px 10px;      border-radius: 5px;  }  .navigation ul li a:hover {      color: #ccc;      background-color: #555;  }  @media (max-width: 768px) {      .Three\_blocks {          flex-direction: column;          align-items: center;      }      .One\_line {          width: 100%;          justify-content: center;          margin-bottom: 20px;      }      .navigation ul {          flex-direction: column;      }      .navigation ul li {          margin: 10px 0;      }      .navigation ul li a {          font-size: 16px;      }      .logo-block img {          width: 100px;          height: auto;      }      .contact-block h2, .title-text {          font-size: 18px;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .Three\_blocks {          flex-direction: row;          justify-content: space-between;      }      .One\_line {          justify-content: center;      }      .navigation ul {          flex-direction: row;          justify-content: center;      }      .navigation ul li {          margin: 0 15px;      }      .navigation ul li a {          font-size: 16px;      }      .logo-block img {          width: 120px;          height: auto;      }      .contact-block h2, .title-text {          font-size: 20px;      }  }  @media (min-width: 1025px) {      .Three\_blocks {          justify-content: space-between;      }      .navigation ul li a {          font-size: 18px;      }      .logo-block img {          width: 150px;          height: auto;      }      .contact-block h2, .title-text {          font-size: 24px;      }  }  /\* Блок с изображением \*/  .image-container {      position: relative;      height: 80vh;      overflow: hidden;  }  .image-container img {      width: 100%;      height: 100%;      object-fit: cover;  }  .overlay-text {      position: absolute;      top: 50%;      left: 50%;      transform: translate(-50%, -50%);      color: #f2f2f2;      font-size: 2.5vw;      text-align: center;  }  .buttons-container {      position: absolute;      bottom: 10%;      left: 50%;      transform: translateX(-50%);      display: flex;      gap: 20px;  }  .button, .button1 {      padding: 15px 30px;      border: none;      border-radius: 30px;      font-size: 16px;      text-transform: uppercase;      cursor: pointer;  }  .button {      background-color: transparent;      color: #fff;      border: 2px solid #fff;      transition: background-color 0.3s ease, color 0.3s ease;  }  .button:hover {      background-color: #fff;      color: #333;  }  .button1 {      background-color: #fff;      color: #000;      border: 2px solid transparent;      transition: background-color 0.3s ease, color 0.3s ease;  }  .button1:hover {      background-color: #000;      color: #fff;  }  @media (max-width: 768px) {      .image-container {          height: 60vh;      }      .overlay-text {          font-size: 2.5vw;      }      .buttons-container {          flex-direction: column;          bottom: 5%;      }      .button, .button1 {          font-size: 14px;          padding: 10px 20px;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .image-container {          height: 70vh;      }      .overlay-text {          font-size: 2.5vw;      }      .buttons-container {          flex-direction: row;          gap: 10px;      }      .button, .button1 {          font-size: 15px;          padding: 12px 25px;      }  }  @media (min-width: 1025px) {      .image-container {          height: 80vh;      }      .overlay-text {          font-size: 2.5vw;      }      .buttons-container {          gap: 20px;      }      .button, .button1 {          font-size: 16px;          padding: 15px 30px;      }  }  /\* Блок "О нас" \*/  .container1 {      max-width: 800px;      margin: 50px auto;      padding: 20px;      background-color: #ffffff;      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);      border-radius: 10px;  }  .container1 h1 {      font-size: 24px;      text-align: center;  }  .container1 p {      font-size: 18px;      line-height: 1.6;      text-align: center;      margin-bottom: 30px;  }  .slider-container {      position: relative;      margin-top: 30px;  }  .slider {      display: flex;      transition: transform 0.5s ease;  }  .slider img {      width: 100%;      height: auto;      object-fit: cover;  }  .prev-button, .next-button {      position: absolute;      top: 50%;      transform: translateY(-50%);      background-color: rgba(0, 0, 0, 0.5);      color: #fff;      border: none;      padding: 10px;      border-radius: 50%;      cursor: pointer;      transition: background-color 0.3s ease;      font-size: 24px;  }  .prev-button {      left: 20px;  }  .next-button {      right: 20px;  }  .prev-button:hover, .next-button:hover {      background-color: rgba(0, 0, 0, 0.7);  }  @media (max-width: 768px) {      .container1 {          padding: 10px;          margin: 20px auto;      }      .container1 h1 {          font-size: 20px;      }      .container1 p {          font-size: 16px;      }      .title {          font-size: 24px;      }      .prev-button, .next-button {          font-size: 18px;          padding: 8px;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .container1 {          padding: 15px;          margin: 30px auto;      }      .container1 h1 {          font-size: 22px;      }      .container1 p {          font-size: 17px;      }      .title {          font-size: 28px;      }      .prev-button, .next-button {          font-size: 20px;          padding: 10px;      }  }  @media (min-width: 1025px) {      .container1 {          padding: 20px;          margin: 50px auto;      }      .container1 h1 {          font-size: 24px;      }      .container1 p {          font-size: 18px;      }      .title {          font-size: 32px;      }      .prev-button, .next-button {          font-size: 24px;          padding: 10px;      }  }  /\* Стили для таблицы условий \*/  .pricing-table {      max-width: 800px;      margin: 0 auto;      margin-bottom: 50px;      text-align: center;      background-color: #fff;  }  .pricing-table table {      width: 100%;      border-collapse: collapse;      margin: 0 auto;  }  .pricing-table th, .pricing-table td {      padding: 10px;      border-bottom: 1px solid #fff;      text-align: left;  }  .pricing-table th {      background-color: #fff;      font-size: 20px;      text-align: center;  }  .pricing-table td {      font-size: 18px;      color: #333;      text-align: center;  }  @media (max-width: 768px) {      .pricing-table, .pricing-table table {          width: 100%;      }      .pricing-table th, .pricing-table td {          padding: 8px;          font-size: 16px;      }      .pricing-table th {          font-size: 16px;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .pricing-table, .pricing-table table {          width: 95%;      }      .pricing-table th, .pricing-table td {          padding: 10px;          font-size: 17px;      }      .pricing-table th {          font-size: 18px;      }  }  @media (min-width: 1025px) {      .pricing-table, .pricing-table table {          max-width: 800px;      }      .pricing-table th, .pricing-table td {          padding: 12px;          font-size: 18px;      }      .pricing-table th {          font-size: 20px;      }  }  /\* Преимущества \*/  .advantages-section {      max-width: 1200px;      margin: 0 auto 50px auto;      display: flex;      flex-wrap: wrap;      gap: 30px;  }  .advantage {      flex: 1 1 300px;      padding: 20px;      background-color: #fff;      border-radius: 10px;      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);      text-align: center;  }  .advantage-title {      font-size: 20px;      margin-bottom: 10px;  }  .advantage-description {      font-size: 16px;      color: #666;      text-align: center;  }  .advantage-icon {      width: 50px;      height: auto;      margin-bottom: 20px;  }  @media (max-width: 768px) {      .advantages-section {          flex-direction: column;          gap: 20px;      }      .advantage {          flex: 1 1 auto;          margin: 0 10px;      }      .advantage-title {          font-size: 18px;      }      .advantage-description {          font-size: 14px;      }      .advantage-icon {          width: 40px;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .advantages-section {          flex-direction: row;          flex-wrap: wrap;      }      .advantage {          flex: 1 1 45%;          margin: 0 10px;      }      .advantage-title, .advantage-description {          font-size: 16px;      }      .advantage-icon {          width: 45px;      }  }  @media (min-width: 1025px) {      .advantages-section {          flex-direction: row;          gap: 30px;      }      .advantage {          flex: 1 1 300px;      }      .advantage-title {          font-size: 20px;      }      .advantage-description {          font-size: 16px;      }      .advantage-icon {          width: 50px;      }  }  /\* Документы \*/  .documents-container {      max-width: 1200px;      margin: 0 auto;      display: flex;      flex-wrap: wrap;      justify-content: center;      gap: 20px;  }  .document-item {      flex: 0 0 calc(50% - 20px);      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  }  .document-image {      width: 100%;      height: auto;  }  @media (max-width: 768px) {      .document-item {          flex: 0 0 100%;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .document-item {          flex: 0 0 calc(50% - 20px);      }  }  @media (min-width: 1025px) {      .documents-container {          justify-content: center;      }      .document-item {          flex: 0 0 calc(33.333% - 20px);      }  }  /\* Контактная информация \*/  .container {      display: flex;      height: 100%;  }  .map-container {      position: relative;      flex: 1 1 60%;  }  .map-buttons {      position: absolute;      top: 10px;      right: 10px;      z-index: 1;  }  .map-buttons a {      display: block;      margin-bottom: 5px;      background-color: #fff;      padding: 5px 10px;      border: 1px solid #ccc;      border-radius: 5px;      text-decoration: none;      color: #333;      font-size: 14px;  }  .info-container {      flex: 1 1 40%;      background-color: #fff;      padding: 20px;      border-radius: 10px;      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);      text-align: center;  }  .contact1 {      margin-top: 0;  }  .info-container p {      margin: 10px 0;  }  .hours {      margin-top: 20px;  }  @media (max-width: 768px) {      .container {          flex-direction: column;      }      .map-container {          flex: 1 1 auto;          height: 300px;      }      .info-container {          flex: 1 1 auto;          margin-top: 20px;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .container {          flex-direction: row;      }      .map-container, .info-container {          flex: 1 1 50%;      }  }  @media (min-width: 1025px) {      .container {          flex-direction: row;      }      .map-container {          flex: 1 1 60%;      }      .info-container {          flex: 1 1 40%;      }  }  /\* Форма обратной связи \*/  .question-form-container {      max-width: 600px;      margin: 0 auto;      padding: 20px;      text-align: center;      background-color: #fff;      border-radius: 10px;      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  }  .question-form-container h2 {      margin-bottom: 20px;  }  form#contact-form {      display: flex;      flex-direction: column;      align-items: center;  }  form#contact-form input[type="text"],  form#contact-form input[type="tel"] {      margin: 10px 0;      padding: 15px;      border: 2px solid #ccc;      border-radius: 30px;      font-size: 16px;      outline: none;      width: 70%;  }  form#contact-form button {      padding: 15px 30px;      margin-top: 20px;      border: none;      border-radius: 30px;      background-color: #000;      color: #fff;      font-size: 16px;      cursor: pointer;      outline: none;      width: auto;  }  form#contact-form button:hover {      background-color: #333;  }  @media (max-width: 768px) {      .question-form-container {          padding: 10px;          max-width: 100%;      }      form#contact-form input[type="text"], form#contact-form input[type="tel"] {          width: 90%;      }      form#contact-form button {          padding: 10px 20px;      }  }  @media (min-width: 769px) and (max-width: 1024px) {      .question-form-container {          max-width: 80%;      }      form#contact-form input[type="text"], form#contact-form input[type="tel"] {          width: 80%;      }  }  @media (min-width: 1025px) {      .question-form-container {          max-width: 600px;      }      form#contact-form input[type="text"], form#contact-form input[type="tel"], form#contact-form button {          width: 70%;      }  } |

Листинг 2.2 Стиль для авторизации

|  |
| --- |
| body {      background-color: #f8f9fa;  }  .login-container {      max-width: 400px;      margin: auto;      padding: 40px 20px;      background-color: #fff;      border-radius: 5px;      box-shadow: 0 0 10px rgba(0,0,0,0.1);  }  .login-title {      text-align: center;      margin-bottom: 30px;  }  .form-group {      margin-bottom: 20px;  }  .form-control {      width: 100%;      padding: 12px 20px;      font-size: 16px;      border-radius: 5px;      border: 1px solid #ccc;      box-sizing: border-box;  }  .btn {      width: 100%;      padding: 12px;      font-size: 16px;      border-radius: 5px;      cursor: pointer;  }  .btn-primary {      background-color: #007bff;      border: none;  }  .btn-primary:hover {      background-color: #0056b3;  } |

Листинг 2.3 Фрагмент кода единого стиля административной панели

|  |
| --- |
| body {      font-family: Arial, sans-serif;      background-color: #f4f4f4;      margin: 0;      padding: 0;  }  .container {      max-width: 800px;      margin: 0 auto;      padding: 20px;      background-color: #fff;      border-radius: 8px;      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);      text-align: center;      margin-top: 50px;  }  h1 {      color: #333;  }  .welcome-message {      margin-bottom: 20px;  }  input[type="file"],  input[type="submit"] {      width: 250px;      padding: 10px;      margin: 5px;      border-radius: 5px;      border: 1px solid #ccc;      box-sizing: border-box;      cursor: pointer;      transition: background-color 0.3s ease;  }  input[type="submit"] {      background-color: #007bff;      color: white;  }  input[type="submit"]:hover {      background-color: #0056b3;  }  img {      margin-top: 20px;      max-width: 200px;  }  .back-link {      position: absolute;      top: 20px;      left: 20px;      color: #007bff;      padding: 10px 20px;      text-decoration: none;      font-size: 16px;      border-radius: 5px;      transition: color 0.3s ease;  }  .back-link:hover {      color: #0056b3;  } |

**Приложение №3**

Листинг 3.1 Форма главной страницы административной панели

|  |
| --- |
| <div class="container">      <h1 class="title">Административная панель</h1>          </div>          <div class="navigation">              <a href="/admin/header.php">Шапка</a> <br>              <a href="/admin/about.php">О нас</a> <br>              <a href="/admin/service.php">Условия</a> <br>              <a href="/admin/advantages.php">Преимущества</a> <br>              <a href="/admin/license.php">Лицензии</a> <br>              <a href="/admin/contact.php">Контакты</a> <br>              <a href="/admin/application.php">Заявки</a> <br>          </div>  </div> |

Листинг 3.2 Форма шапки

|  |
| --- |
| <body>  <div class="container">          <?php              $sql=$pdo->prepare("SELECT \* FROM Header");              $sql->execute();              $res=$sql->fetch(PDO::FETCH\_OBJ);          ?>          <form action="/admin/header/header.php" method="post" enctype="multipart/form-data">              <input type="text" name="Name" placeholder="Введите название" value="<?php echo htmlspecialchars($res->Name); ?>">              <br>              <img src="header/img/<?php echo $res->Filename?>" width="200">              <br>              <input type="file" name="Filename">              <br>              <input type="text" id="phone" name="Phone" placeholder="Введите телефон" value="<?php echo $res->Phone?>">              <br>              <input type="submit" name="save" value="Сохранить">          </form>      </div>      <script src="/js/telephone.js"></script>  </body> |

Листинг 3.3 Форма блока «О нас»

|  |
| --- |
| <div class="container">      <h1>Редактирование блока "О нас"</h1>          </div>          <div class="form-container">          <?php          $sql=$pdo->prepare("SELECT \* FROM About");          $sql->execute();          $res=$sql->fetch(PDO::FETCH\_OBJ);          ?>          <form action="/admin/about/about1.php" method="post" enctype="multipart/form-data">              <textarea name="Description" placeholder="Введите текст" class="textarea"><?php echo $res->Description; ?></textarea>              <br>              <input type="submit" name="save\_description" value="Сохранить"> <br>          </form>          <?php              $sql=$pdo->prepare("SELECT \* FROM Slider");              $sql->execute();              while ($res = $sql->fetch(PDO::FETCH\_OBJ)):              ?>          <form action="/admin/about/about2.php" method="post" enctype="multipart/form-data">              <br><input type="hidden" name="update\_id[]" value="<?php echo $res->ID; ?>">              <img src="about/img/<?php echo $res->Filename?>" width="200"><br>              <input type="file" name="Filename"> <br>              <br>              <input type="submit" name="save\_slider\_images" value="Сохранить изображения слайдера"> <br>              <button type="submit" name="delete" value="<?php echo $res->ID; ?>">Удалить</button> <br>              <br>          </form>          <?php endwhile ?>          <form class="add-form" action="/admin/about/about2.php" method="post" enctype="multipart/form-data">              <input type="file" name="new\_Filename"> <br>              <input class="save1" type="submit" name="add" value="Добавить"> <br>          </form>          </div>  </div> |

Листинг 3.4 Форма блока «Наши условия»

|  |
| --- |
| <div class="container">      <h1>Редактирование блока "Наши условия"</h1>          <form action="/admin/service/service.php" method="post" enctype="multipart/form-data">              <label class="label-amount" for="amount">Сумма:</label>              <label class="label-chart" for="chart">Питание:</label>              <label class="label-conditions" for="conditions">График:</label>              <label class="label-total-amount" for="total\_amount">Общая сумма в месяц:</label><br>              <?php              $sql = $pdo->prepare("SELECT \* FROM Services");              $sql->execute();              while ($res = $sql->fetch(PDO::FETCH\_OBJ)):                  ?>                  <div class="form-group">                      <input type="hidden" name="update\_id[]" value="<?php echo $res->ID; ?>">                      <input class="input-field" type="text" name="Amount[]" value="<?php echo $res->Amount; ?>">                      <select name="Chart[]" class="input-field">                          <option value="+" <?php if ($res->Chart == '+') echo 'selected'; ?>>+</option>                          <option value="-" <?php if ($res->Chart == '-') echo 'selected'; ?>>-</option>                      </select>                      <select name="Conditions[]" class="input-field">                          <option value="15/15" <?php if ($res->Conditions == '15/15') echo 'selected'; ?>>15/15</option>                          <option value="7/7" <?php if ($res->Conditions == '7/7') echo 'selected'; ?>>7/7</option>                          <option value="2/4" <?php if ($res->Conditions == '2/4') echo 'selected'; ?>>2/4</option>                          <option value="1/3" <?php if ($res->Conditions == '1/3') echo 'selected'; ?>>1/3</option>                      </select>                      <input class="input-field" type="text" name="total\_amount[]" value="<?php echo $res->total\_amount?>">                      <button type="submit" name="delete" value="<?php echo $res->ID; ?>">Удалить</button>                  </div>                  <br>              <?php endwhile ?>              <div class="form-group">                  <input class="input-field" type="text" name="new\_Amount[]" placeholder="Введите сумму">                  <select class="input-field" name="new\_Chart[]">                      <option value="+">+</option>                      <option value="-">-</option>                  </select>                  <select class="input-field" name="new\_Conditions[]">                      <option value="15/15">15/15</option>                      <option value="7/7">7/7</option>                      <option value="2/4">2/4</option>                      <option value="1/3">1/3</option>                  </select>                  <input class="input-field" type="text" name="new\_total\_amount[]" placeholder="Введите общую сумму">              </div>              <br>              <input class="input1" type="submit" name="save" value="Сохранить">          </form>  </div> |

Листинг 3.5 Форма блока «Наши преимущества»

|  |
| --- |
| <div class="container">      <h1>Редактирование блока "Наши преимущества"</h1>          </div>          <?php              $sql = $pdo->prepare("SELECT \* FROM Advantages");              $sql->execute();              while ($res = $sql->fetch(PDO::FETCH\_OBJ)):          ?>          <form class="advantage-form" action="/admin/advantages/advantages.php" method="post" enctype="multipart/form-data">              <input type="hidden" name="update\_id[]" value="<?php echo $res->ID; ?>">              <img src="advantages/img/<?php echo $res->Filename?>" alt="Преимущество <?php echo $res->Name; ?>">              <input type="text" name="Name" placeholder="Введите название" value="<?php echo $res->Name; ?>"> <br>              <input type="text" name="Description" placeholder="Введите описание" value="<?php echo $res->Description?>"> <br>              <input type="file" name="Filename"> <br>              <input class="save1" type="submit" name="save" value="Сохранить"> <br>              <button type="submit" name="delete" value="<?php echo $res->ID; ?>">Удалить</button> <br>          </form>          <?php endwhile ?>          <h2>Добавить новую запись</h2>          <form class="add-form" action="/admin/advantages/advantages.php" method="post" enctype="multipart/form-data">              <input type="text" name="new\_Name" placeholder="Название"> <br>              <input type="text" name="new\_Description" placeholder="Описание"> <br>              <input type="file" name="new\_Filename"> <br>              <input class="save1" type="submit" name="add" value="Добавить"> <br>          </form>  </div> |

Листинг 3.6 Форма блока «Лицензии»

|  |
| --- |
| <div class="container">      <h1>Редактирование блока "Лицензии"</h1>          </div>          <?php              $sql=$pdo->prepare("SELECT \* FROM License");              $sql->execute();              $res=$sql->fetch(PDO::FETCH\_OBJ);          ?>          <form action="/admin/license/license.php" method="post" enctype="multipart/form-data">              <img src="license/img/<?php echo $res->Filename?>" width="200"> <br>              <input type="file" name="Filename"><br><br>              <input type="submit" name="save" value="Сохранить">          </form>  </div> |

Листинг 3.7 Форма блока «Контактные данные»

|  |
| --- |
| <body>      <div class="container">      <h1>Редактирование блока "Контактные данные"</h1>          </div>          <?php              $sql=$pdo->prepare("SELECT \* FROM Contact");              $sql->execute();              $res=$sql->fetch(PDO::FETCH\_OBJ);          ?>  <form action="/admin/contact/contact.php" method="post">      <input type="text" id="phone" name="Phone" placeholder="Введите телефон" value="<?php echo $res->Phone; ?>"> <br>      <input type="text" name="Email" placeholder="Введите почту" value="<?php echo $res->Email; ?>"> <br>      <input class="save1" type="submit" name="save" value="Сохранить">  </form>      </div>      <script src="/js/telephone.js"></script>  </body> |

Листинг 3.8 Форма блока «Остались вопросы?»

|  |
| --- |
| <div class="container">      <h1>Редактирование блока "Остались вопросы?"</h1>          </div>          <table>              <tr>                  <th>Имя</th>                  <th>Телефон</th>              </tr>              <?php              $sql = $pdo->prepare("SELECT \* FROM Application");              $sql->execute();              $res = $sql->fetchAll(PDO::FETCH\_ASSOC);              ?>              <?php foreach ($res as $row): ?>                  <tr>                      <td><?= $row['Name']; ?></td>                      <td><?= $row['Phone']; ?></td>                      <td>                          <form action="/admin/application/application.php" method="POST">                              <input type="hidden" name="delete" value="<?= $row['ID']; ?>">                              <button type="submit">Удалить</button>                          </form>                      </td>                  </tr>              <?php endforeach; ?>          </table>  </div> |

**Приложение №4**

Листинг 4.1 Редактирование шапки

|  |
| --- |
| if (isset($\_POST["save"])) {      $name = $\_POST["Name"];      $phone = $\_POST["Phone"];          $sql = "UPDATE Header SET Name=:Name, Filename=:Filename, Phone=:Phone";          $query = $pdo->prepare($sql);          $query->execute(["Name" => $name, "Filename" => $\_FILES['Filename']['name'], "Phone" => $phone]);          $sqlContact = "UPDATE Contact SET Phone=:Phone";          $queryContact = $pdo->prepare($sqlContact);          $queryContact->execute(["Phone" => $phone]);      } else {          $sql = "UPDATE Header SET Name=:Name, Phone=:Phone";          $query = $pdo->prepare($sql);          $query->execute(["Name" => $name, "Phone" => $phone]);          $sqlContact = "UPDATE Contact SET  Phone=:Phone";          $queryContact = $pdo->prepare($sqlContact);          $queryContact->execute([ "Phone" => $phone]);      }      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/header.php">';  } |

Листинг 4.2 Редактирование слайдера с фотографиями

|  |
| --- |
| function resizeImage($sourcePath, $destinationPath, $targetWidth, $targetHeight, $extension) {      list($width, $height) = getimagesize($sourcePath);      $src = null;      switch ($extension) {          case 'jpg':          case 'jpeg':              $src = imagecreatefromjpeg($sourcePath);              break;          case 'png':              $src = imagecreatefrompng($sourcePath);              break;      }      $dst = imagecreatetruecolor($targetWidth, $targetHeight);      imagecopyresampled($dst, $src, 0, 0, 0, 0, $targetWidth, $targetHeight, $width, $height);      switch ($extension) {          case 'jpg':          case 'jpeg':              imagejpeg($dst, $destinationPath);              break;          case 'png':              imagepng($dst, $destinationPath);              break;      }      imagedestroy($src);      imagedestroy($dst);  }  if (isset($\_POST["save\_slider\_images"])) {          $upload = 'img/' . $filename;          resizeImage($tempFile, $upload, 2560, 1920, $extension);          $sql = "UPDATE Slider SET Filename=:Filename WHERE id=:ID";          $query = $pdo->prepare($sql);          $query->execute(["Filename" => $filename, "ID" => $\_POST["update\_id"][0]]);      }      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/about.php">';  }  if (isset($\_POST["delete"])) {      $idToDelete = $\_POST["delete"];      $sql = "DELETE FROM Slider WHERE id=:ID";      $query = $pdo->prepare($sql);      $query->execute(["ID" => $idToDelete]);      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/about.php">';  }  if (isset($\_POST["add"])) {      $new\_filename = null;          $new\_upload = 'img/' . $new\_filename;          resizeImage($new\_tempFile, $new\_upload, 2560, 1920, $new\_extension);      }      $sql = "INSERT INTO Slider (Filename) VALUES (:Filename)";      $query = $pdo->prepare($sql);      $params = ["Filename" => $new\_filename];      $query->execute($params);      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/about.php">';  } |

Листинг 4.3 Редактирование условий

|  |
| --- |
| if (isset($\_POST["save"])) {      if(isset($\_POST["update\_id"])) {          $update\_ids = $\_POST["update\_id"];          $amounts = $\_POST["Amount"];          $charts = $\_POST["Chart"];          $conditions = $\_POST["Conditions"];          $total\_amounts = $\_POST["total\_amount"];          foreach($update\_ids as $index => $update\_id) {              $amount = $amounts[$index];              $chart = $charts[$index];              $condition = $conditions[$index];              $total\_amount = $total\_amounts[$index];              $sql = "UPDATE Services SET Amount=:Amount, Chart=:Chart, Conditions=:Conditions, total\_amount=:total\_amount WHERE id=:ID";              $query = $pdo->prepare($sql);              $query->execute(["Amount" => $amount, "Chart" => $chart, "Conditions" => $condition, "total\_amount" => $total\_amount, "ID" => $update\_id]);          }      }      if(isset($\_POST["new\_Amount"])) {          $new\_amounts = $\_POST["new\_Amount"];          $new\_charts = $\_POST["new\_Chart"];          $new\_conditions = $\_POST["new\_Conditions"];          $new\_total\_amounts = $\_POST["new\_total\_amount"];          foreach($new\_amounts as $index => $new\_amount) {              $new\_chart = $new\_charts[$index];              $new\_condition = $new\_conditions[$index];              $new\_total\_amount = $new\_total\_amounts[$index];              $sql = "INSERT INTO Services (Amount, Chart, Conditions, total\_amount) VALUES (:Amount, :Chart, :Conditions, :total\_amount)";              $query = $pdo->prepare($sql);              $query->execute(["Amount" => $new\_amount, "Chart" => $new\_chart, "Conditions" => $new\_condition, "total\_amount" => $new\_total\_amount]);          }      }      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/service.php">';  }  if(isset($\_POST["delete"])) {      $idToDelete = $\_POST["delete"];      $sql = "DELETE FROM Services WHERE ID=:ID";      $query = $pdo->prepare($sql);      $query->execute(["ID" => $idToDelete]);      echo "Запрос на удаление выполнен для ID: " . $idToDelete;  }  echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/service.php">'; |

Листинг 4.4 Редактирование преимуществ

|  |
| --- |
| if (isset($\_POST["save"])) {      $name = $\_POST["Name"];      $description = $\_POST["Description"];          $sql = "UPDATE Advantages SET Name=:Name, Filename=:Filename, Description=:Description WHERE id=:ID";          $query = $pdo->prepare($sql);          $query->execute(["Name" => $name, "Filename" => $\_FILES['Filename']['name'], "Description" => $description, "ID" => $\_POST["update\_id"][0]]);      } else {          $sql = "UPDATE Advantages SET Name=:Name, Description=:Description WHERE id=:ID";          $query = $pdo->prepare($sql);          $query->execute(["Name" => $name, "Description" => $description, "ID" => $\_POST["update\_id"][0]]);      }      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/advantages.php">';  }  if(isset($\_POST["delete"])) {      $idToDelete = $\_POST["delete"];      $sql = "DELETE FROM Advantages WHERE id=:ID";      $query = $pdo->prepare($sql);      $query->execute(["ID" => $idToDelete]);      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/advantages.php">';  }  if (isset($\_POST["add"])) {      $new\_name = $\_POST["new\_Name"];      $new\_description = $\_POST["new\_Description"];      $new\_filename = null;      if(empty($new\_name) && empty($new\_description)) {          exit("Необходимо заполнить хотя бы одно поле: Name или Description");      }      $sql = "INSERT INTO Advantages (Name, Description, Filename) VALUES (:Name, :Description, :Filename)";      $query = $pdo->prepare($sql);      $params = ["Name" => $new\_name, "Description" => $new\_description, "Filename" => $new\_filename];      $query->execute($params);      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/advantages.php">';  } |

Листинг 4.5 Редактирование лицензии

|  |
| --- |
| if (isset($\_POST["save"])) {          $sql = "UPDATE License SET Filename=:Filename";          $query = $pdo->prepare($sql);          $query->execute(["Filename" => $\_FILES['Filename']['name']]);      }      echo '<meta HTTP-EQUIV="Refresh" Content="0; URL=/admin/license.php">';  } |