Государственное АВТОНОМНОЕ профессиональное образовательное учреждение

свердловской области

«Каменск-Уральский политехнический колледж»

Специальность 09.02.07 Информационные системы и программирование

Группа ВИСиП20-401

КУРСОВОЙ ПРОЕКТ

по МДК 09.01 ПРОЕКТИРОВАНИЕ И разработка

ВЕБ-ПРИЛОЖЕНИЙ

тема: Разработка ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ АВТОСЕРВИСА

Выполнил:

А.А. Галахов

Проверил:

Е.Э. Цупра

2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc156225374)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc156225375)

[1.1. Описание предметной области 4](#_Toc156225376)

[1.2. Постановка задачи 7](#_Toc156225377)

[2. ПРОЕКТНАЯ ЧАСТЬ 9](#_Toc156225378)

[2.1. Проектирование и реализация базы данных веб-приложения 9](#_Toc156225379)

[2.2. Разработка маршрутизации и контроллеров веб-приложения 11](#_Toc156225380)

[2.3. Разработка представлений для страниц веб-приложения 12](#_Toc156225381)

[ЗАКЛЮЧЕНИЕ 30](#_Toc156225382)

[СПИСОК ЛИТЕРАТУРЫ 31](#_Toc156225383)

# ВВЕДЕНИЕ

Web-приложение - это прикладное программное обеспечение, логика которого распределена между сервером и клиентом, а обмен информацией происходит по сети. Клиентская часть реализует пользовательский интерфейс, а серверная - получает и обрабатывает запросы от клиента, выполняет вычисления, формирует веб-страницу и отправляет её клиенту согласно протоколу HTTP.

В современном мире практически каждое предприятие имеет свой собственный веб-сайт или веб-приложение. Также, созданное веб-приложение способствует решению некоторых задач, это ускоряет обслуживание клиентов и приводит к увеличению прибыли предприятия.

Автосервис - это комплекс услуг, связанный с техническим обслуживанием и ремонтом авто и мототранспорта.

Как правило, автосервис осуществляется на станциях технического обслуживания (СТО) и ориентирован как на личный автотранспорт физических лиц, так и на автопарк организаций.

Актуальность бизнеса по ремонту и обслуживанию автотранспорта связана со взрывным ростом уровня автомобилизации.

Актуальность исследования обусловлена необходимостью предоставления полной достоверной информации о предоставляемых услугах, а также оформлению заявки на ремонт в автосервис.

Цель данного проекта разработать серверную часть веб-приложения для автосервиса.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Разработка функционального веб-приложения для автосервиса
* Создание формы для записи пользователей на ремонт
* Создание системы управления заявками пользователей
* Создание системы добавления и изменения услуг и категорий услуг
* Создание системы добавления и изменения сотрудников и их должностей

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Описание предметной области

Автосервис — это комплекс услуг, связанный с техническим обслуживанием и ремонтом авто и мототранспорта.

Как правило, автосервис осуществляется на станциях технического обслуживания (СТО) и ориентирован как на личный автотранспорт физических лиц, так и на автопарк организаций.

В автосервисе, предоставляющем качественные услуги, вы увидите сложное оборудование для автосервиса, оборудование для шиномонтажа, диагностики и регулировки развал-схождения, стенд для правки кузовов и сварочное оборудование, окрасочную камеру и зоны подготовки, установку для обслуживание кондиционеров и многое другое, повстречаете специалистов, способных не только в короткие сроки определить неисправность, но и устранить ее в кратчайшие сроки.

Сервис (сервисная система) – это совокупность средств, способов и методов предоставления платных услуг по приобретению, эффективному использованию, обеспечению работоспособности, экономичности, дорожной и экологической безопасности автотранспортных средств в течение всего срока их службы.

Сервисные услуги включают следующие основные виды работ:

* подбор и доставку необходимых автомобилей, оборудования, запасных частей и материалов;
* куплю и продажу новых и подержанных автомобилей и агрегатов;
* предпродажное обслуживание и гарантийный ремонт;
* заправку, мойку, уборку и хранение;
* техническое обслуживание и ремонт автотранспортных средств;
* инструментальный технический осмотр и подготовку к нему;
* продажу запасных частей, материалов, комплектующих изделий и принадлежностей;
* предоставление автотранспортных средств в прокат и лизинг;
* техническую помощь на линии, эвакуацию, услуги по самообслуживанию;
* модернизацию, переоборудование и тюнинг автотранспортных средств;
* сбор и утилизацию отходов, образующихся при эксплуатации автотранспортных средств;
* информационное обеспечение владельцев автотранспортных средств;
* обучение и консультацию персонала автотранспортных предприятий, предпринимателей, физических лиц – владельцев автотранспортных средств.

Автосервис можно рассматривать как инфраструктуру автомобильного транспорта, включающую в себя системы торговли, поддержания работоспособности и восстановления автомобилей, его технической эксплуатации, использования и устранения вредных последствий.

В традиционном понимании автосервис подразумевает весь комплекс услуг, оказываемых владельцам автотранспорта:

а) торговые услуги:

* продажа автомобилей и запасных частей;
* предоставление автомобилей во временное пользование (аренду или прокат);
* оценка стоимости подержанных автомобилей;

б) услуги ТО и Р:

* техническое обслуживание и ремонт автомобилей, отдельных агрегатов и узлов на СТОА, специализированных и смешанных пунктах и участках ТО и Р;
* мойка автомобилей;
* косметический ремонт, отделка, покраска автомобилей;
* заправка автомобилей топливом, маслами, рабочими жидкостями, замена смазки;
* предоставление производственной площади, оборудования и инструмента для самостоятельного технического обслуживания и ремонта индивидуальных автомобилей;
* консультации по обслуживанию и ремонту автомобилей;

в) дополнительные услуги;

* организация и охрана автостоянок, мотелей и кемпингов;
* тонирование, укрепление стекол и фар;
* установка, заправка и диагностика кондиционеров;
* установка автосигнализации, аудиотехники.

Услуги, оказываемые владельцам автомобилей фирмами, производящими эти автомобили, как непосредственно, так и через представительства, выделены в отдельную группу услуг, отнесенную к фирменному обслуживанию автомобилей:

* гарантийное техническое обслуживание и ремонт автомобилей на СТО, организуемые фирмами – производителями;
* продажа автомобилей и запасных частей через фирменную торговую сеть;
* предпродажная подготовка автомобилей (обязательная и заказная);
* доставка проданных автомобилей владельцам;
* организация выставок и автосалонов, рекламирующих новые конструкции автомобилей и комплектующих элементов;
* выпуск литературы по техническому обслуживанию, ремонту и диагностике автомобилей;
* оказание компьютерных услуг индивидуальным владельцам и предприятиям при планировании упреждающей замены агрегатов и узлов с использованием теории надежности, массового обслуживания и других математических методов.

При оказании услуг между клиентом и автосервисом оформляется договор о предоставлении обслуживания транспорта.

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой «черный ящик» со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня. Наиболее важная функция расположена в верхнем левом углу. А соединяются функции между собой при помощи стрелок и описаний функциональных блоков. При этом каждый вид стрелки или активности имеет собственное значение. Данная модель позволяет описать все основные виды процессов, как административные, так и организационные. Стрелки могут быть:

* Входящие – вводящие, которые ставят определенную задачу.
* Исходящие – выводящие результат деятельности.
* Контроль (сверху вниз) – механизмы управления (положения, инструкции и пр.).
* Механизмы (снизу вверх) – что используется для того, чтобы произвести необходимую работу.



Рисунок - 1 Функциональная модель в нотации IDEF0

Вводящие стрелки – «Информация о клиенте», «Информация об автомобиле». Это те вводные, которые необходимы для начала работы. Управляющие для написания статьи – это «Правила оформления».

А в роли «Механизмов» выступают сотрудник автосервиса и оборудование. В данном случае сотрудник производит выполнение сервисных работ и оказание услуг, используя оборудование.

## Постановка задачи

Создание веб-приложения - это на сегодня однозначно один из самых прогрессивных путей инвестирования времени и ресурсов в развитие компании, ведь его внедрение позволяет:

* автоматизировать бизнес-процессы и для сотрудников, и для персонала;
* привлечь внимание целевой аудитории и выделиться на фоне конкурентов;
* сделать решение востребованных задач проще, надежнее и безопаснее.

Приходим к выводу, что создание веб-приложения для автоматизации оформления заявок на ремонт автомобиля положительно повлияет на развитие бизнеса.

В данном приложении будет доступно: просмотр информации об автосервисе, прайс его услуг и запись на ремонт. Что позволит любому пользователю быстро получить информацию и решить для себя нужен-ли ему именно этот автосервис или нет.

При разработке веб-приложения будет использоваться язык программирования PHP. Для разработки структуры веб-страниц будет использоваться язык гипертекстовой разметки HTML5. Для стилизации и визуального оформления страницы будет использоваться CSS. Для обеспечения интерактивности и клиентской логики будет использоваться язык программирования JavaScript.

# ПРОЕКТНАЯ ЧАСТЬ

# Проектирование и реализация базы данных веб-приложения

Для начала проектирования базы данных веб-приложения автосервису необходима ER-модель, которая будет визуально представлять схему базы данных.

ER-модель (модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области.

На рисунке 2 представлена ER-модель логической структуры базы данных веб-приложения автосервиса.

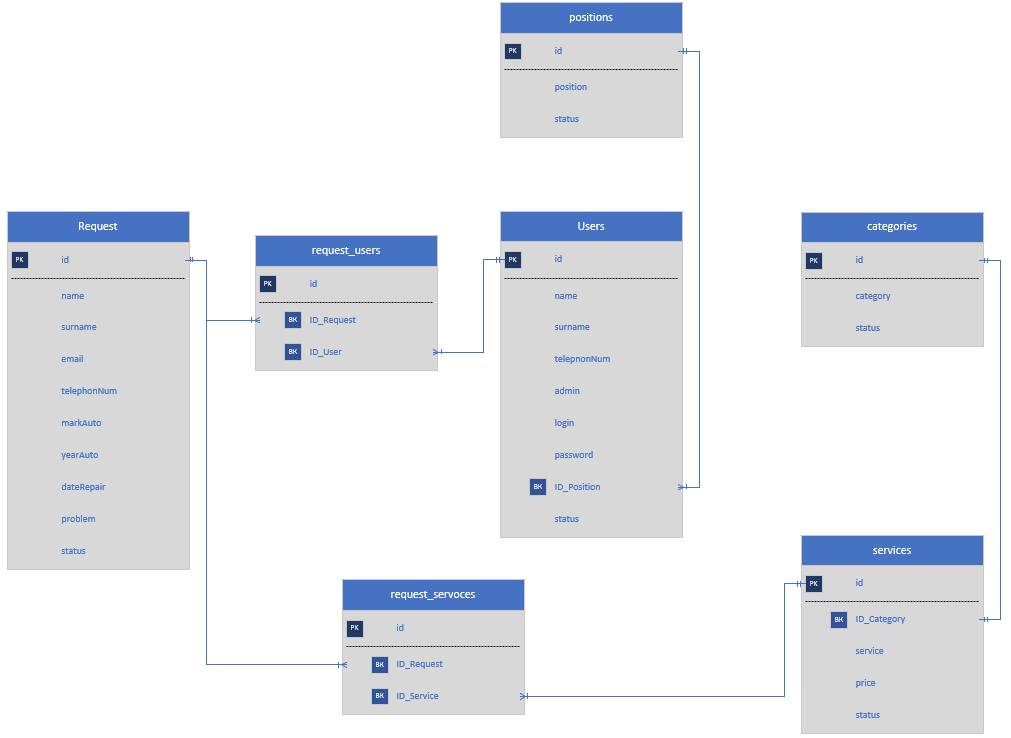


Рисунок 2 - ER-модель логической структуры базы данных

Таблица Users содержит информацию о работниках. Таблица содержит следующие поля:

* id – уникальный идентификатор пользователя
* name, surname – имя фамилия сотрудника
* telephonNum – телефон сотрудника
* ID\_position – уникальный идентификатор категории, имеет связь один ко многим с таблицей positions
* status – статус работника
* admin – это поле показывает является-ли сотрудник администратором
* login, password – логин и пароль администратора

Таблица positions содержит информацию о должностях. Таблица содержит следующие поля:

* id – уникальный идентификатор должности
* position – название должности
* status – статус должности

Таблица request\_users содержит информацию о работниках выполняющих ремонт. Таблица содержит следующие поля:

* id – уникальный идентификатор
* ID\_Request – уникальный идентификатор заявки, имеет связь один ко многим с таблицей Request
* ID\_User – уникальный идентификатор работника, имеет связь один ко многим с таблицей users

Таблица requests содержит информацию об отправленной заявке на ремонт. Таблица содержит следующие поля:

* id – уникальный идентификатор заявки
* name, surname – имя и фамилия отправителя заявки
* email, telephonNum – почта и номер телефона отправителя для обратной связи
* markAuto, yearAuto – марка и год выпуска автомобиля
* dateRepair – дата ремонта
* problem – проблема отправителя
* status – статус заявки

Таблица request\_services содержит информацию о выполняемых услугах. Таблица содержит следующие поля:

* id – уникальный идентификатор
* ID\_Request – уникальный идентификатор заявки, имеет связь один ко многим с таблицей Request
* ID\_Services – уникальный идентификатор услуги, имеет связь один ко многим с таблицей services

Таблица services содержит информацию о всех услугах. Таблица содержит следующие поля:

* id – уникальный идентификатор услуги
* service – название услуги
* ID\_Category– уникальный идентификатор категории, имеет связь один ко многим с таблицей categories
* price – цена услуги
* status – статус услуги

Таблица categories содержит информацию о всех услугах. Таблица содержит следующие поля:

* id – уникальный идентификатор категории
* category – название категории
* status – статус категории

# Разработка маршрутизации и контроллеров веб-приложения

Главная страница: /index.php

Страница оставления заявки: /order/index.php

Страница входа в панель работника автосервиса: /panel/signin.php

Панель работника автосервиса: /panel/index.php

Список заявок: /panel/requests/index.php

Информация о заявке: /panel/requests/view.php?id={ID}

Удаление заявки: /panel/requests/delete.php?id={ID}

Назначение работника на заявку: /panel/requests/assign.php?id={ID}&userId={USER\_ID}

Назначение услуги на заявку: /panel/requests/assignService.php?id={ID}&serviceId={SERVICE\_ID}

Снять работника с заявки: /panel/requests/unassign.php?id={ID}&userId={USER\_ID}

Убрать услугу с заявки: /panel/requests/unassignService.php?id={ID}&serviceId={SERVICE\_ID}

Указать статус заявки: /panel/requests/setstatus.php?id={ID}

Список зарегистрированных пользователей: /panel/users/index.php

Список категорий услуг: /panel/categories/index.php

Редактирование категории: /panel/categories/edit.php?id={ID}

Удаление категории: /panel/categories/delete.php?id={ID}

Список услуг: /panel/services/index.php

Добавление услуги: /panel/services/create.php

Удаление услуги: /panel/services/delete.php?id={ID}

Редактирование услуги: /panel/services/edit.php?id={ID}

Список должностей: /panel/positions/index.php

Добавление должности: /panel/positions/create.php

Удаление должности: /panel/positions/delete.php?id={ID}

Редактирование должности: /panel/positions/edit.php?id={ID}

# Разработка представлений для страниц веб-приложения

Разработка представлений главной страницы и страницы отправки заявки на ремонт

Главная страница состоит из нескольких разделов – слайдов с информацией, краткий список услуг, отзывы и расположение на карте.

Указание метаданных:

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

Подключение стилей и скриптов, необходимых для корректного отображения и функционирования страницы:

    <link rel="stylesheet" href="css/grid.css" />

    <link rel="stylesheet" href="css/styles.css" />

    <script src="js/index.js"></script>

Код для отображения расположения на карте:

<div style="position:relative;overflow:hidden;">

            <a href="https://yandex.ru/maps/54/yekaterinburg/?utm\_medium=mapframe&utm\_source=maps" style="color:#eee;font-size:12px;position:absolute;top:0px;">

                Екатеринбург</a>

            <a href="https://yandex.ru/maps/geo/yekaterinburg/53166537/?ll=60.607511%2C56.745518&utm\_medium=mapframe&utm\_source=maps&z=10.26" style="color:#eee;font-size:12px;position:absolute;top:14px;">

                Екатеринбург — Яндекс Карты

            </a>

            <iframe src="https://yandex.ru/map-widget/v1/?ll=60.607511%2C56.745518&mode=poi&poi%5Bpoint%5D=60.597393%2C56.837993&poi%5Buri%5D=ymapsbm1%3A%2F%2Fgeo%3Fdata%3DCgg1MzE2NjUzNxJP0KDQvtGB0YHQuNGPLCDQodCy0LXRgNC00LvQvtCy0YHQutCw0Y8g0L7QsdC70LDRgdGC0YwsINCV0LrQsNGC0LXRgNC40L3QsdGD0YDQsyIKDdBjckIVIFpjQg%2C%2C&z=10.26" width="99.5%" height="400" frameborder="1" allowfullscreen="true" style="position:relative;">

            </iframe>

        </div>

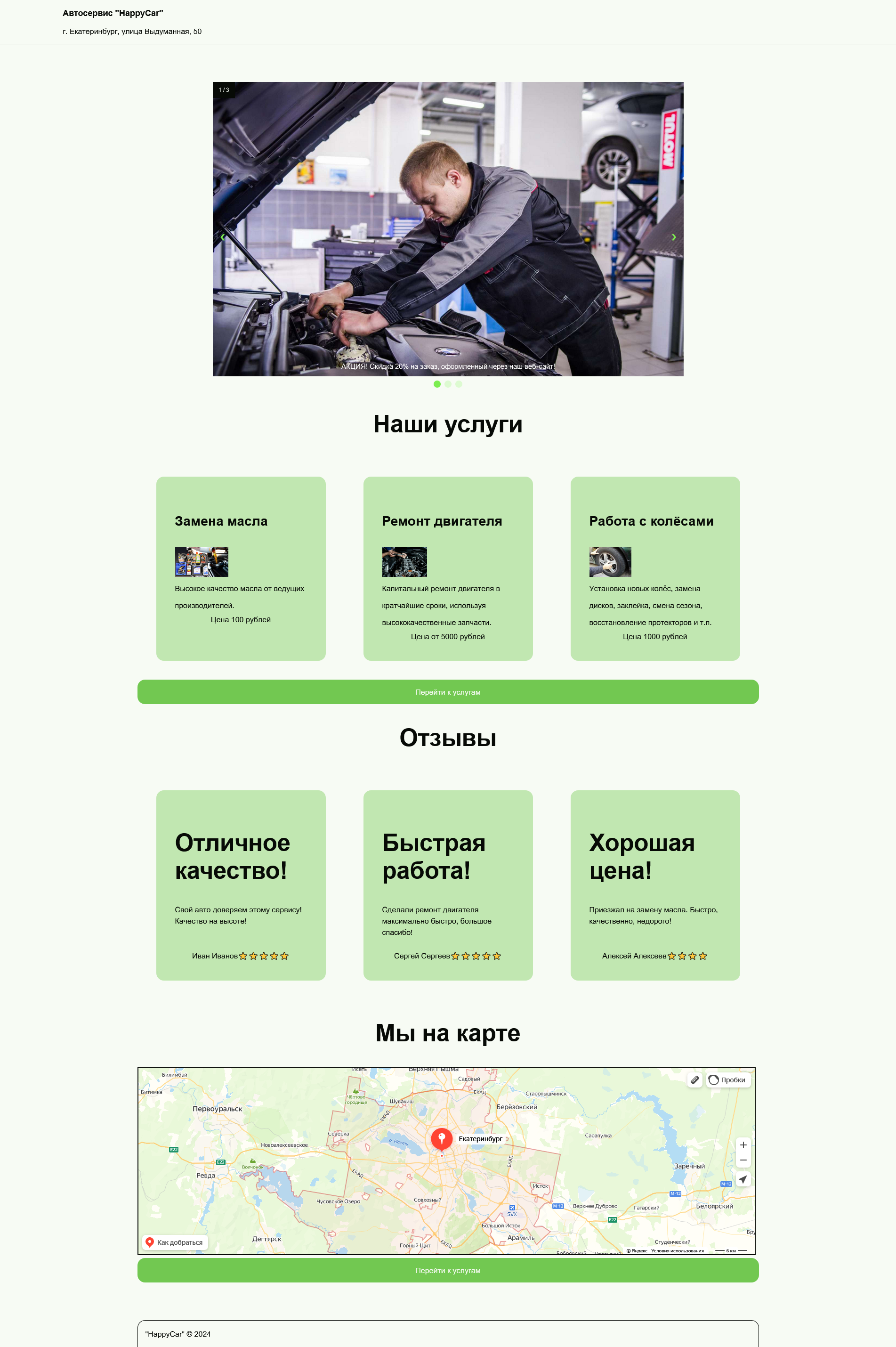


Рисунок 3 – Главная страница сайта

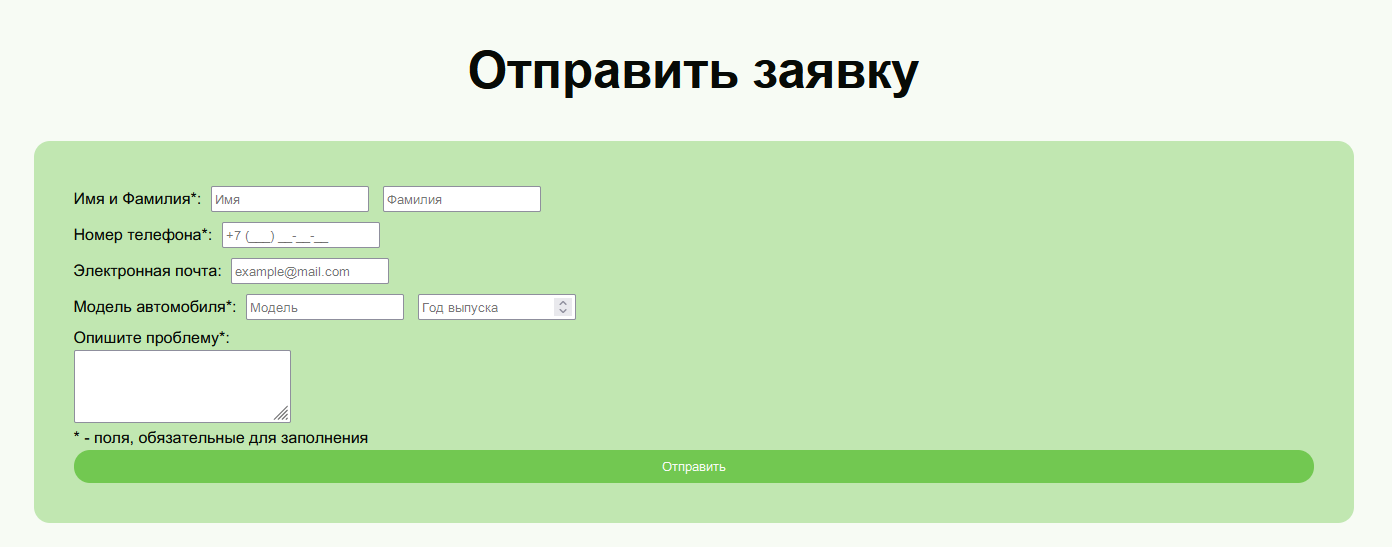


Рисунок 4 – Форма отправки заявки

<form class="row section-card" action="send.php" method="POST">

                <div>

                    <span>Имя и Фамилия\*:</span> <input name="firstName" type="text" required placeholder="Имя"> <input name="lastName" type="text" required placeholder="Фамилия">

                </div>

                <div>

                    <span>Номер телефона\*:</span> <input name="phoneNumber" type="tel" placeholder="+7 (\_\_\_) \_\_-\_\_-\_\_" required>

                </div>

                <div>

                    <span>Электронная почта:</span> <input name="email" type="email" placeholder="example@mail.com">

                </div>

                <div>

                    <span>Модель автомобиля\*:</span> <input type="text" name="carMark" required placeholder="Модель"> <input type="number" name="carYear" required placeholder="Год выпуска">

                </div>

                <div>

                    <span>Опишите проблему\*:</span> <br> <textarea name="problem" type="text" required></textarea>

                </div>

                <span>\* - поля, обязательные для заполнения</span>

                <button type="submit" class="primary-button">Отправить</button>

            </form>

Разработка представлений панели управления

Для управления сайтом имеется отдельный раздел сайта, недоступный обычным пользователям.

Для уменьшения написанного кода используется общий файл “db.php”, содержащий часто используемые фрагменты кода, а также код для подключения к базе данных.

$host = "localhost";

$db = "happycar";

$user = "root";

$password = "";

$link = mysqli\_connect($host, $user, $password) or trigger\_error(mysqli\_error($link), E\_USER\_ERROR);

mysqli\_select\_db($link, $db);

$dbUser = null;

if (isset($\_SESSION["user\_id"])) {

    $query = "SELECT \* FROM `users` WHERE `id`='" . $\_SESSION["user\_id"] . "' LIMIT 1";

    $result = mysqli\_query($link, $query);

    $dbUser = mysqli\_fetch\_assoc($result);

}

Прежде чем получить доступ к панели управления необходимо совершить авторизацию.

Для проверки авторизации используется следующий код:

session\_start();

if (!isset($\_SESSION["user\_id"])) {

    header("Location: signin.php");

    die();

}

require\_once("../connections/db.php");

if (!isset($dbUser)) {

    die();

}

Форма для логина:

<form action="login.php" method="POST">

        <input name="login" type="text" placeholder="Логин" required>

        <input name="password" type="password" placeholder="Пароль" required>

        <input class="primary-button" type="submit" value="Вход">

    </form>

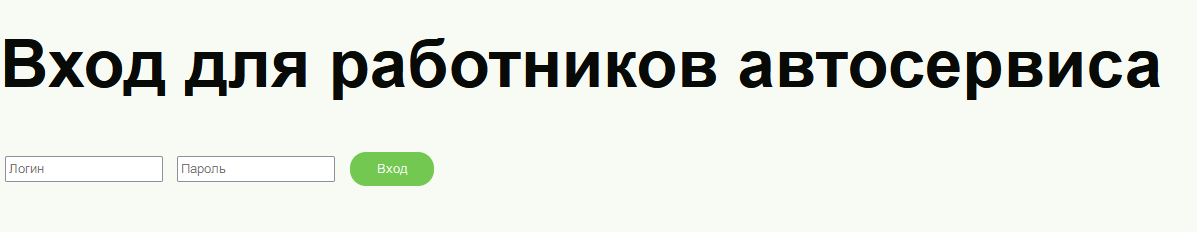


Рисунок 5 – Форма входа в панель управления

По нажатии кнопки «Вход» выполняется следующий код:

session\_start();

require\_once("../connections/db.php");

$username = $\_POST["login"];

$password = $\_POST["password"];

$query = "SELECT \* FROM `users` WHERE `login`=?";

$stmt = $link->prepare($query);

$stmt->bind\_param("s", $username);

$stmt->execute();

$result = $stmt->get\_result()->fetch\_assoc();

$stmt->close();

if ($result == false) {

    echo "Пользователь не найден";

} else {

    $dbUsername = $result["login"];

    $dbPassword = $result["password"];

    if (password\_verify($password, $dbPassword)) {

        $\_SESSION["username"] = $dbUsername;

        $\_SESSION["user\_id"] = $result["id"];

        header("Location: index.php");

        die();

    } else {

        echo "Пользователь не найден";

    }

}

После авторизации работник автосервиса увидит главную страницу панели управления, откуда можно получить доступ к другим элементам управления, таким как список заявок. Если у работника есть права администратора, то список доступных элементов расширен.

Код для отображения страницы:

<body>

    <main>

        <h1>Панель работника автосервиса</h1>

        Вы вошли как: <b><?php echo $dbUser["firstName"] . " " . $dbUser["lastName"] ?></b> <br>

        <?php

        $userPosition = getDbUserPosition($link, $dbUser);

        echo "Ваша должность: <b>" . $userPosition["position"] . "</b><br>";

        if ($dbUser["isAdmin"]) {

        ?>

            <a class="primary-button" href="categories/index.php">Список категорий услуг</a> <br>

            <a class="primary-button" href="services/index.php">Список услуг</a> <br>

            <a class="primary-button" href="users/index.php">Список пользователей</a> <br>

            <a class="primary-button" href="positions/index.php">Список должностей</a> <br>

        <?php

        }

        ?>

        <a class="primary-button" href="requests/index.php">Список заявок</a> <br>

        <?php

        ?>

        <a class="danger-button" href="logout.php">Выйти из аккаунта</a>

    </main>

</body>

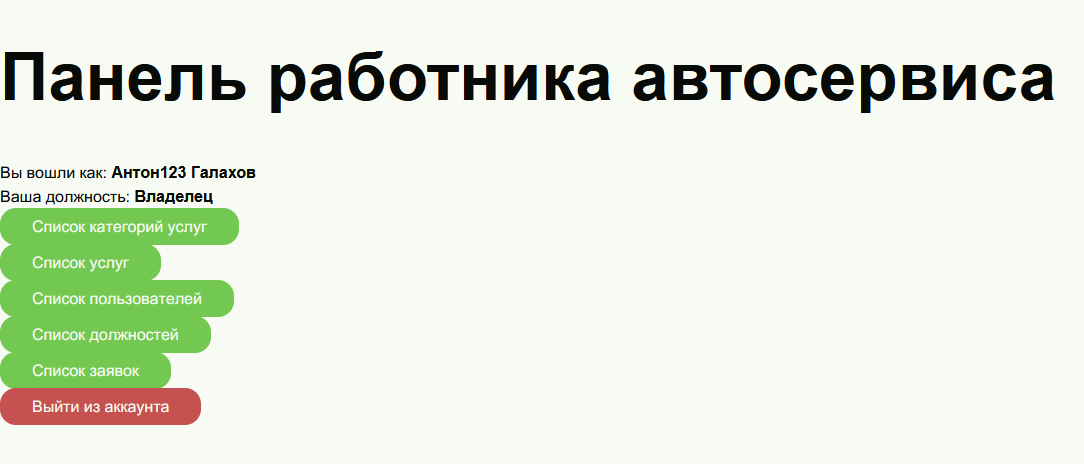


Рисунок 6 – Вид главной страницы панели управления при наличии прав администратора

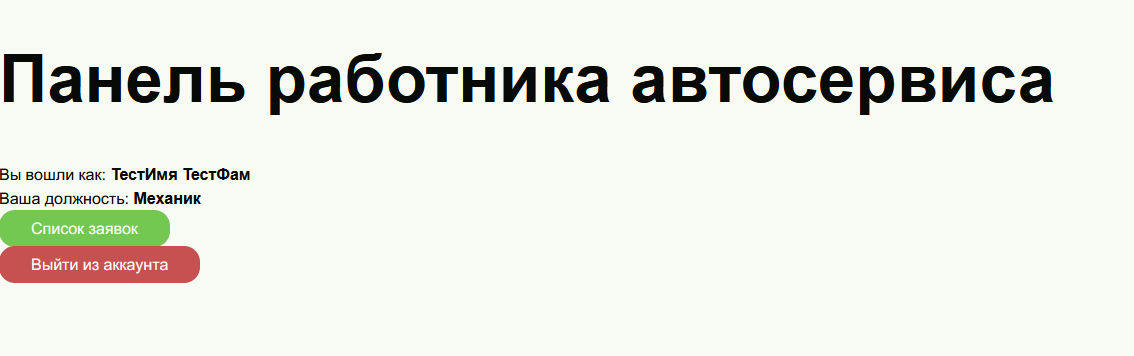


Рисунок 7 – Вид главной страницы панели управления при отсутствии прав администратора

На странице списка категорий услуг можно просмотреть список созданных категорий, добавить новые категории, изменить информацию о них или удалить их. Для отображения всего списка выполняется запрос к базе данных и используется цикл while.

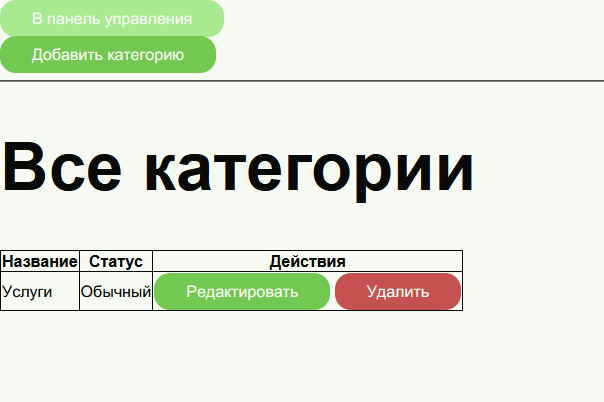


Рисунок 8 – Вид списка категорий услуг

<table>

        <thead>

            <th>Название</th>

            <th>Статус</th>

            <th>Действия</th>

        </thead>

        <tbody>

            <?php

            $allCategories = getDbCategories($link);

            while ($row = mysqli\_fetch\_array($allCategories)) {

            ?>

                <tr>

                    <td><?php echo $row["category"] ?></td>

                    <td><?php echo localizeStatus($row["status"]) ?></td>

                    <td>

                        <a class="primary-button" href="edit.php?id=<?php echo $row["id"] ?>">Редактировать</a>

                        <a class="danger-button" href="delete.php?id=<?php echo $row["id"] ?>">Удалить</a>

                    </td>

                </tr>

            <?php

            }

            ?>

        </tbody>

    </table>

На странице списка услуг можно увидеть список всех доступных услуг, их цены, создать новые услуги или изменить их категории.

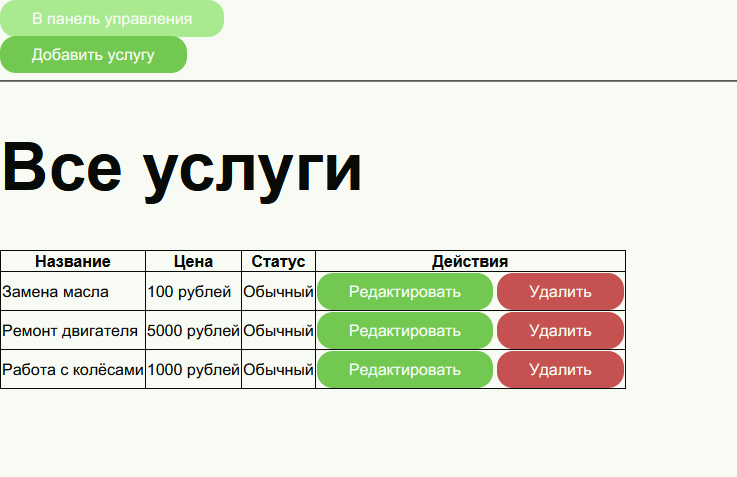


Рисунок 9 – Вид списка услуг

<table>

        <thead>

            <th>Название</th>

            <th>Цена</th>

            <th>Статус</th>

            <th>Действия</th>

        </thead>

        <tbody>

            <?php

            $allServices = getDbServices($link);

            while ($row = mysqli\_fetch\_array($allServices)) {

            ?>

                <tr>

                    <td><?php echo $row["service"] ?></td>

                    <td><?php echo $row["price"] . " рублей" ?></td>

                    <td><?php echo localizeStatus($row["status"]) ?></td>

                    <td>

                        <a class="primary-button" href="edit.php?id=<?php echo $row["id"] ?>">Редактировать</a>

                        <a class="danger-button" href="delete.php?id=<?php echo $row["id"] ?>">Удалить</a>

                    </td>

                </tr>

            <?php

            }

            ?>

        </tbody>

    </table>

На странице списка пользователей администраторы могут создавать новых пользователей, редактировать информацию о существующих, выдавать или забирать права администратора, а также удалять пользователей. Для защиты от совершения ошибок администратор не может отобрать права администратора у самого себя или удалить свой аккаунт.

У панели управления автосервиса закрытая регистрация пользователей, так как обычные клиенты не должны иметь доступ к панели управления, или иметь аккаунт в автосервисе.

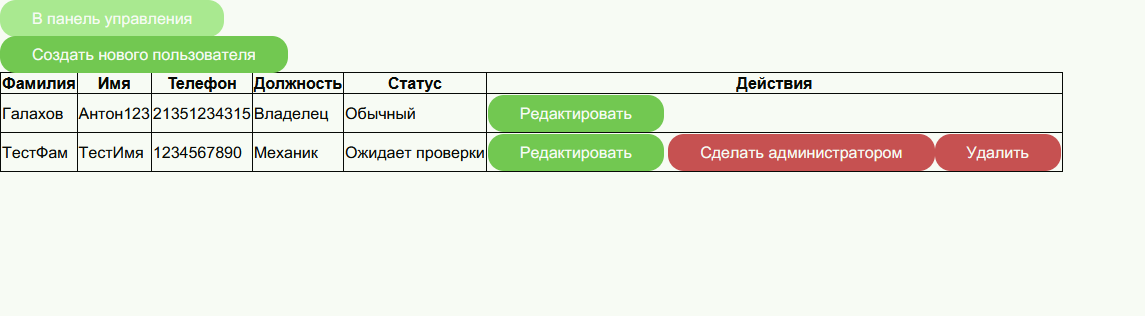


Рисунок 10 – Вид списка пользователей

<table>

        <thead>

            <th>Фамилия</th>

            <th>Имя</th>

            <th>Телефон</th>

            <th>Должность</th>

            <th>Статус</th>

            <th>Действия</th>

        </thead>

        <tbody>

            <?php

            $query = "SELECT \* FROM `users`";

            $result = mysqli\_query($link, $query);

            while ($row = mysqli\_fetch\_array($result)) {

                $position = getDbUserPosition($link, $row);

            ?>

                <tr>

                    <td><?php echo $row["lastName"] ?></td>

                    <td><?php echo $row["firstName"] ?></td>

                    <td><?php echo $row["phoneNumber"] ?></td>

                    <td><?php echo $position["position"] ?></td>

                    <td><?php echo localizeStatus($row["status"]) ?></td>

                    <td>

                        <a class="primary-button" href="edit.php?id=<?php echo $row["id"] ?>">Редактировать</a>

                        <?php

                        if ($row["id"] != $dbUser["id"]) {

                            if ($row["isAdmin"]) {

                                echo "<a class=\"danger-button\" href=\"demote.php?id=" . $row["id"] . "\">Убрать права администратора</a>";

                            } else {

                                echo "<a class=\"danger-button\" href=\"promote.php?id=" . $row["id"] . "\">Сделать администратором</a>";

                            }

                            echo "<a class=\"danger-button\" href=\"delete.php?id=" . $row["id"] . "\">Удалить</a>";

                        }

                        ?>

                    </td>

                </tr>

            <?php

            }

            ?>

        </tbody>

    </table>

На странице списка должностей администратор может посмотреть имеющиеся должности, редактировать их, создавать новые или удалять существующие.

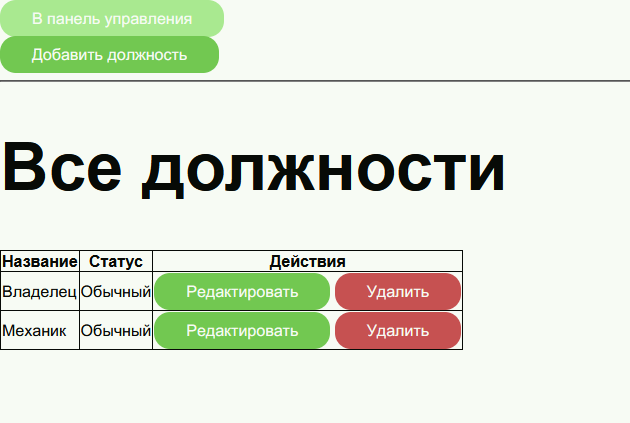


Рисунок 11 – Вид списка должностей

<table>

        <thead>

            <th>Название</th>

            <th>Статус</th>

            <th>Действия</th>

        </thead>

        <tbody>

            <?php

            $allPositions = getDbPositions($link);

            while ($row = mysqli\_fetch\_array($allPositions)) {

            ?>

                <tr>

                    <td><?php echo $row["position"] ?></td>

                    <td><?php echo localizeStatus($row["status"]) ?></td>

                    <td>

                        <a class="primary-button" href="edit.php?id=<?php echo $row["id"] ?>">Редактировать</a>

                        <a class="danger-button" href="delete.php?id=<?php echo $row["id"] ?>">Удалить</a>

                    </td>

                </tr>

            <?php

            }

            ?>

        </tbody>

    </table>

На странице списка заявок пользователь может посмотреть полный список заявок и просмотреть полную информацию о заявках. Администраторы могут редактировать информацию о заявках или удалять их.



Рисунок 12 – Вид списка заявок

<table>

        <thead>

            <th>Номер заявки</th>

            <th>Фамилия</th>

            <th>Имя</th>

            <th>Электронная почта</th>

            <th>Номер телефона</th>

            <th>Модель автомобиля</th>

            <th>Год выпуска</th>

            <th>Дата починки</th>

            <th>Статус заявки</th>

            <th>Действия</th>

        </thead>

        <tbody>

            <?php

            $query = "SELECT \* FROM `requests` ORDER BY `id` DESC";

            $result = mysqli\_query($link, $query);

            while ($row = mysqli\_fetch\_array($result)) {

            ?>

                <tr>

                    <td><?php echo $row["id"] ?></td>

                    <td><?php echo $row["lastName"] ?> </td>

                    <td><?php echo $row["firstName"] ?></td>

                    <td><?php echo $row["email"] ?></td>

                    <td><?php echo $row["phoneNumber"] ?></td>

                    <td><?php echo $row["carMark"] ?></td>

                    <td><?php echo $row["carYear"] ?></td>

                    <td><?php echo $row["repairDate"] ?></td>

                    <td><?php echo localizeStatus($row["status"]) ?></td>

                    <td>

                        <a class="primary-button" href="view.php?id=<?php echo $row["id"] ?>">Подробности</a>

                        <?php

                        if ($dbUser["isAdmin"]) {

                        ?>

                            <a class="primary-button" href="edit.php?id=<?php echo $row["id"] ?>">Редактировать основную информацию</a>

                            <a class="danger-button" href="delete.php?id=<?php echo $row["id"] ?>">Удалить</a>

                        <?php

                        }

                        ?>

                    </td>

                </tr>

            <?php

            }

            ?>

        </tbody>

    </table>

По нажатии кнопки «Подробности» открывается страница с полной информацией о заявке, на которой можно изменить статус заявки, а также назначить работников и услуги.

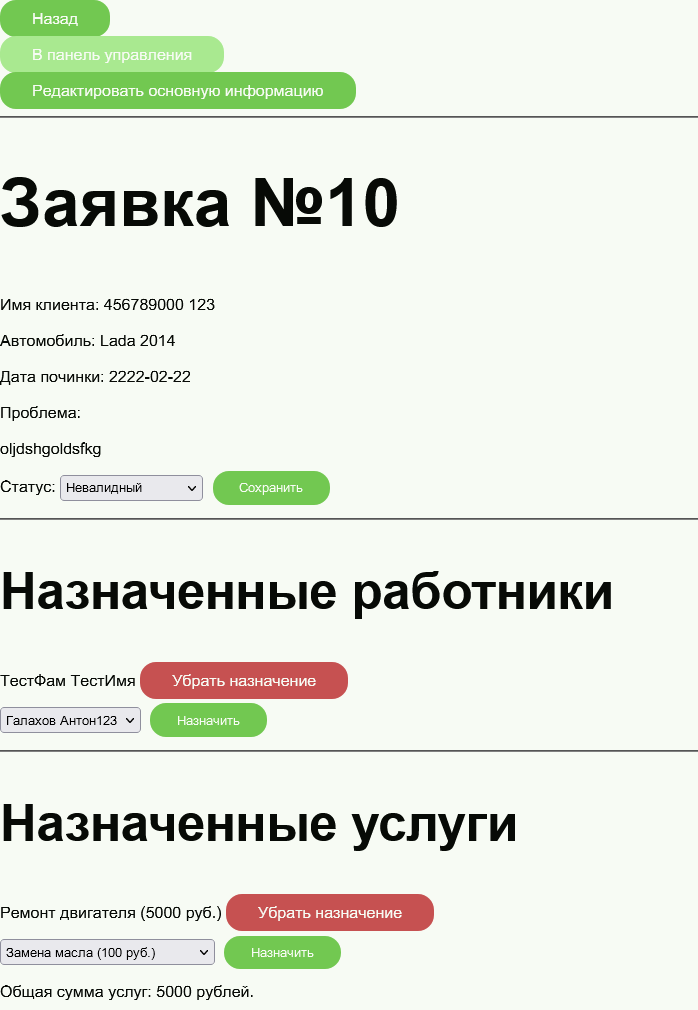


Рисунок 13 – Вид подробной информации о заявке

<body>

    <a class="primary-button" href="index.php">Назад</a> <br>

    <a class="secondary-button" href="/panel/index.php">В панель управления</a> <br>

    <?php

    if ($dbUser["isAdmin"]) {

    ?>

        <a class="primary-button" href="edit.php?id=<?php echo $requestToView["id"] ?>">Редактировать основную информацию</a> <br>

    <?php

    }

    ?>

    <hr>

    <h1>Заявка №<?php echo $requestToView["id"] ?></h1>

    <p>Имя клиента: <?php echo $requestToView["lastName"] . " " . $requestToView["firstName"] ?></p>

    <p>Автомобиль: <?php echo $requestToView["carMark"] . " " . $requestToView["carYear"] ?></p>

    <?php

    if (isset($requestToView["repairDate"])) {

    ?>

        <p>Дата починки: <?php echo $requestToView["repairDate"] ?></p>

    <?php

    }

    ?>

    <p>Проблема:</p>

    <p><?php echo $requestToView["problem"] ?></p>

    <?php

    if ($dbUser["isAdmin"]) {

    ?>

        <form action="setstatus.php" method="POST">

            <label>Статус: </label>

            <input hidden name="id" value="<?php echo $requestToView["id"] ?>">

            <select name="status">

                <option <?php if ($requestToView["status"] == "Pending") echo "selected" ?> value="Pending"><?php echo localizeStatus("Pending") ?></option>

                <option <?php if ($requestToView["status"] == "Normal") echo "selected" ?> value="Normal"><?php echo localizeStatus("Normal") ?></option>

                <option <?php if ($requestToView["status"] == "Invalid") echo "selected" ?> value="Invalid"><?php echo localizeStatus("Invalid") ?></option>

                <option <?php if ($requestToView["status"] == "Completed") echo "selected" ?> value="Completed"><?php echo localizeStatus("Completed") ?></option>

            </select>

            <input class="primary-button" type="submit" value="Сохранить">

        </form>

        <hr>

        <h2>Назначенные работники</h2>

        <div>

            <?php

            $allUsers = getDbRequestUsers($link, $requestToView["id"]);

            while ($row = mysqli\_fetch\_array($allUsers)) {

            ?>

                <p>

                    <span>

                        <?php

                        $assignedUser = getDbUserById($link, $row["userId"]);

                        echo $assignedUser["lastName"] . " " . $assignedUser["firstName"];

                        ?>

                    </span>

                    <a class="danger-button" href="unassign.php?id=<?php echo $requestToView["id"] ?>&userId=<?php echo $row["userId"] ?>">Убрать назначение</a>

                </p>

            <?php

            }

            ?>

        </div>

        <form action="assign.php" method="POST">

            <input hidden name="id" value="<?php echo $requestToView["id"] ?>">

            <select name="userId">

                <?php

                $allUsers = getDbUsers($link);

                while ($row = mysqli\_fetch\_array($allUsers)) {

                ?>

                    <option value="<?php echo $row["id"] ?>"><?php echo $row["lastName"] . " " . $row["firstName"] ?></option>

                <?php

                }

                ?>

            </select>

            <input class="primary-button" type="submit" value="Назначить">

        </form>

        <hr>

        <h2>Назначенные услуги</h2>

        <div>

            <?php

            $totalCost = 0.0;

            $allRequestsAndServices = getDbRequestServices($link, $requestToView["id"]);

            while ($row = mysqli\_fetch\_array($allRequestsAndServices)) {

            ?>

                <p>

                    <span>

                        <?php

                        $assignedService = getDbService($link, $row["serviceId"]);

                        $totalCost += $assignedService["price"];

                        echo $assignedService["service"] . " (" . $assignedService["price"] . " руб.)";

                        ?>

                        <a class="danger-button" href="unassignService.php?id=<?php echo $requestToView["id"] ?>&serviceId=<?php echo $row["serviceId"] ?>">Убрать назначение</a>

                    </span>

                </p>

            <?php

            }

            ?>

        </div>

        <form action="assignService.php" method="POST">

            <input hidden name="id" value="<?php echo $requestToView["id"] ?>">

            <select name="serviceId">

                <?php

                $allServices = getDbServices($link);

                while ($row = mysqli\_fetch\_array($allServices)) {

                ?>

                    <option value="<?php echo $row["id"] ?>"><?php echo $row["service"] . " (" . $row["price"] . " руб.)" ?></option>

                <?php

                }

                ?>

            </select>

            <input class="primary-button" type="submit" value="Назначить">

        </form>

        <p>Общая сумма услуг: <?php echo $totalCost ?> рублей.</p>

    <?php

    } else {

    ?>

        <p>Статус: <?php echo localizeStatus($requestToView["status"]) ?></p>

        <hr>

        <h2>Назначенные работники</h2>

        <div>

            <?php

            $allUsers = getDbRequestUsers($link, $requestToView["id"]);

            while ($row = mysqli\_fetch\_array($allUsers)) {

            ?>

                <p>

                    <span>

                        <?php

                        if ($row["userId"] == $dbUser["id"]) {

                            echo "<b>" . $dbUser["lastName"] . " " . $dbUser["firstName"] . " (Вы)</b>";

                        } else {

                            $assignedUser = getDbUserById($link, $row["userId"]);

                            echo $assignedUser["lastName"] . " " . $assignedUser["firstName"];

                        }

                        ?>

                    </span>

                </p>

            <?php

            }

            ?>

        </div>

        <hr>

        <h2>Назначенные услуги</h2>

        <div>

            <?php

            $totalCost = 0.0;

            $allRequestsAndServices = getDbRequestServices($link, $requestToView["id"]);

            while ($row = mysqli\_fetch\_array($allRequestsAndServices)) {

            ?>

                <p>

                    <span>

                        <?php

                        $assignedService = getDbService($link, $row["serviceId"]);

                        $totalCost += $assignedService["price"];

                        echo $assignedService["service"] . " (" . $assignedService["price"] . " руб.)";

                        ?>

                    </span>

                </p>

            <?php

            }

            ?>

        </div>

        <p>Общая сумма услуг: <?php echo $totalCost ?> рублей.</p>

    <?php

    }

    ?>

</body>

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта был разработан функциональный сайт для автосервиса.

При разработке использовались следующие технологии: Visual Studio Code, язык PHP, язык JavaScript, CSS – каскадная таблица стилей, HTML – язык гипертекстовой разметки.

В рамках проекта были выполнены следующие задачи: создана форма для записи пользователей на ремонт, разработана система управления заявками пользователей, создана система добавления и изменения услуг и категорий услуг, а также создана система добавления и изменения сотрудников и их должностей.

Веб-приложение поможет автосервису расширить клиентскую базу. Наличие онлайн-каталога услуг и цен в веб-приложении позволит покупателям сделать выбор в пользу вашего автосервиса. Клиенты могут изучить услуги и цены на них без посещения автосервиса, а автосервис, в свою очередь, сократит время на консультации.

Наличие качественного и удобного сайта позволит повысить имидж автосервису, профессионально выглядеть на фоне конкурентов и укрепить доверие клиентов.

# СПИСОК ЛИТЕРАТУРЫ

1. Гниденко, И. Г.  Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2020. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/453640

2. Сысолетин, Е. Г.  Разработка интернет-приложений : учебное пособие для среднего профессионального образования / Е. Г. Сысолетин, С. Д. Ростунцев. — Москва : Издательство Юрайт, 2020. — 90 с. — (Профессиональное образование). — ISBN 978-5-534-10015-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/456393

3. Алексеев А.. Введение в Web-дизайн. Учебное пособие. — М.: ДМК Пресс, 2019. — 184 c.

4. Белов В. В. , Чистякова В. И. Проектирование информационных систем. - Под редакцией: Белов В. В. - М.: Академия, 2019

5. Введение в программные системы и их разработку / С.В. Назаров, С.Н. Белоусова, И.А. Бессонова и др. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2019. - 650 с. : схем., табл., ил. - Библиогр. в кн. ;

6. Гарретт Джесс. Веб-дизайн. Элементы опыта взаимодействия. — М.: Символ-Плюс, 2020. — 285 c.

7. Груздев Д.В. Практика ЭВМ-HTML, 3 курс, 2017

8. Дакетт Д. Основы веб-программирования с использованием HTML, XHTML и CSS. – Эксмо, 2018.

9. Долженко, А.И. Технологии командной разработки программного обеспечения информационных систем / А.И. Долженко. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2019. - 301 с. : схем., ил. - Библиогр. в кн. ;

10. Дэвид Макфарланд. Новая большая книга CSS. — М.: Питер, 2018. — 720 c.

11. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2019. - 494 с. : ил. ;

Интернет-ресурсы:

Самоучитель HTML. [Электронный ресурс]: http://htmlbook.ru/samhtml

Учебник по JavaScript. [Электронный ресурс]: https://learn.javascript.ru/

Учебник по CSS. [Электронный ресурс]: https://www.schoolsw3.com/