**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc75292334)

[1.1 Характеристика места прохождения практики 4](#_Toc75292335)

[1.2 Техническое задание 5](#_Toc75292336)

[1.3 Выбор и обоснование средств разработки программного продукта 8](#_Toc75292337)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 12](#_Toc75292338)

[2.1 Реализация структуры создания «Веб-сайта»: 12](#_Toc75292339)

[2.2 Разработка программного продукта 12](#_Toc75292340)

[2.3 Тестирование и отладка 15](#_Toc75292341)

[2.4 Руководство пользователя 17](#_Toc75292342)

[ЗАКЛЮЧЕНИЕ 20](#_Toc75292343)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 22](#_Toc75292344)

# **ВВЕДЕНИЕ**

Производственная практика — это часть учебного процесса, которая позволяет продемонстрировать практические навыки выполнения работы квалифицированных рабочих и специалистов.

Во время прохождения производственной практики идёт закрепление и конкретизация результатов теоретического и практического обучения в рамках программы, а также приобретения у студентов умения и навыков практической работы по присваиваемой специальности.

Производственная практика по модулю ПМ 03 «Соадминистрирование и автоматизация баз данных и серверов» проходила на базе ФКПОУ НГГТКИ Министерства труда и социальной защиты РФ. Перед прохождение производственной практики был пройден инструктаж по технике безопасности. Данная практика проходила в период с 8.06.2021 по 21.06.2021. Продолжительность составляла 72 учебных часа.

Актуальность данной практики заключается в необходимости создания программных продуктов, помогающих мобильно и коммуникабельно осуществлять определённую деятельность на предприятии или в учреждении.

Цель производственной практики − закрепление, расширение, углубление и систематизация знаний, полученных при изучении общепрофессиональных, специальных и технологических дисциплин, а также первоначальное ознакомление с производственным процессом и начальную адаптацию к профессиональной деятельности.

Для достижения поставленной цели были сформулированы следующие задачи:

* охарактеризовать место прохождение практики;
* проанализировать информационно вычислительную систему предприятия;
* произвести ремонта и обслуживания оргтехники и ПК;
* осуществить установку и настройку программного обеспечения.
* реализовать учебный проект.

Объектом автоматизации служит «Веб-сайт» для общества с ограниченной ответственностью «Техник-Сервис».

Предметами автоматизации является возможность оптимизации процесса связанных с обработкой информации и её обмена с пользователями, а также применение функций для коммуникабельности переходов на сайт.

**1 ОБЩАЯ ЧАСТЬ**

## **Характеристика места прохождения практики**

Компания ООО «Техник-Сервис» была зарегистрирована 25.12.2015 имеет статус по строительству жилых и нежилых зданий.

Дополнительные виды деятельности по ОКВЭД:

|  |
| --- |
| * Сбор и обработка сточных вод |
| * Строительство инженерных коммуникаций для водоснабжения и водоотведения, газоснабжения |
| * Строительство коммунальных объектов для обеспечения электроэнергией и телекоммуникациями |
| * Строительство местных линий электропередачи и связи |
| * Производство электромонтажных работ |
| * Производство прочих строительно-монтажных работ |
| * Производство штукатурных работ |
| * Работы столярные и плотничные |
| * Работы по устройству покрытий полов и облицовке стен |
| * Производство малярных и стекольных работ |
| * Производство прочих отделочных и завершающих работ |
| * Производство кровельных работ |
| * Работы строительные специализированные прочие, не включенные в другие группировки |
| * Работы гидроизоляционные |
| * Деятельность агентов по оптовой торговле строительными материалами |
| * Торговля оптовая прочими машинами и оборудованием   И так далие. |

## **1.2 Техническое задание**

Общие требования

Сайт должен быть разработан с использованием системы управлением сайтами Joomla. После окончания работ необходимо обеспечить возможность Заказчику самостоятельно вносить изменения (редактировать) в структуру и содержимое сайта.

По окончании работ Исполнитель обязан предоставить полностью функционирующий сайт, исходные графические материалы по дизайну, все необходимые данные для доступа к системе управления сайтом (аккаунты, пароли, адреса серверов и т.п.).

Требования к дизайну сайта

Дизайн должен быть выдержан в строгих и мягких тонах. Использовать преимущественно сине-голубые оттенки. Дизайн сайта должен быть выполнен с использованием языка HTML и CSS, при необходимости для создания отдельных графических элементов допустимо использование технологии FLASH.

Сайт должен корректно отображаться в браузерах Microsoft Internet Explorer 6.0, 7.0; Mozilla FireFox 2.0, 3.0; Opera 9.0; Yandex browser.

Необходимо создать структуру (шаблон) сайта, состоящую из следующих элементов:

1. «шапка» (хедер). В данном блоке необходимо расположить логотип предприятия и кнопки основного меню.
2. отображения главного меню сайта. Данное меню должно содержать перечень всех основных страниц. Требования к меню будут изложены далее. Меню должно поддерживать иерархическую структуру, дочерние элементы меню должны быть “переходящими”. Меню должно располагаться по центру «шапки» (хедера) сайта;
3. “подвал” (футер) сайта. В данном блоке необходимо разместить краткую контактную информацию о предприятии и возможность выхода в соцсеть;
4. Блок отображения различного рода графической информации (рекламные баннеры, изображения и т.п.). Данные блоки могут добавляться по мере необходимости.

В дизайне сайта не должны присутствовать:

1. мелькающие баннеры;
2. много сливающегося текста;
3. тёмные и агрессивные цветовые сочетания, и графические решения.

Требования к функциональности сайта

Сайт должен позволять пользователям:

1. осуществлять навигацию по сайту (переход между страницами);
2. возможность зарегистрироваться не зарегистрированному пользователю либо авторизоваться;
3. подать заявку;

Система управления сайтом должна позволять:

1. управлять страницами сайта (добавлять, удалять, изменять их содержимое);
2. управлять элементами меню;
3. добавлять/удалять свои заявки на сайте;
4. загружать на сайт графический материал (фото-видео изображения, различные файлы и т.п.).

Требования к содержимому сайта

Необходимо создать следующие страницы сайта:

1. главная страница сайта;
2. страница «Главная страница»;
3. страница «Каталог»;
4. страница «Подробно»;
5. страница «Заявка»;
6. страниц «Админ»;

Необходимо создать меню навигации по сайту. Меню должно состоять из следующих элементов:

1. Главная страница
2. Каталог
   1. Сантехнические работы
   2. Отопление
   3. Работы по электропроводки
   4. Разное
3. Заявка
4. Выход

Требования к представлению главной страницы сайта.

Главная страница сайта должна содержать графическую часть, навигационное меню сайта, а также контентную область для того, чтобы посетитель сайта с первой страницы мог получить вводную информацию об услугах компании.

Контентная область первой страницы должна делиться на следующие разделы:

1. краткая контактная информация – наименование, телефон и e-mail компании;
2. вверху страницы отображаются облегченная навигационная панель, которая обеспечивает переход к основным пунктам меню сайта (Главная, Заявка и т.д.);

Требования к представлению страница «Каталог»:

На данной странице необходимо разместить; прайс работ по категориям с возможностью ознакомиться с услугой подробнее и сделать заявку.

Требования к представлению страница «Заявка»:

На данной странице необходимо реализовать возможность оформить заявку, или несколько заявок, отследить итоговую сумму к оплате. Так же можно удалить заявку.

## **1.3 Выбор и обоснование средств разработки программного продукта**

Для реализации программного продукта использовалась программа Visual Studio Code 2017:

Несмотря на то, что VS Code делает Микрософт, это бесплатный продукт с открытым исходным кодом, доступный на всех платформах. Благодаря этому и своим возможностям VS Code стал одной из самых популярных сред для разработки в мире показано на рисунке 1.

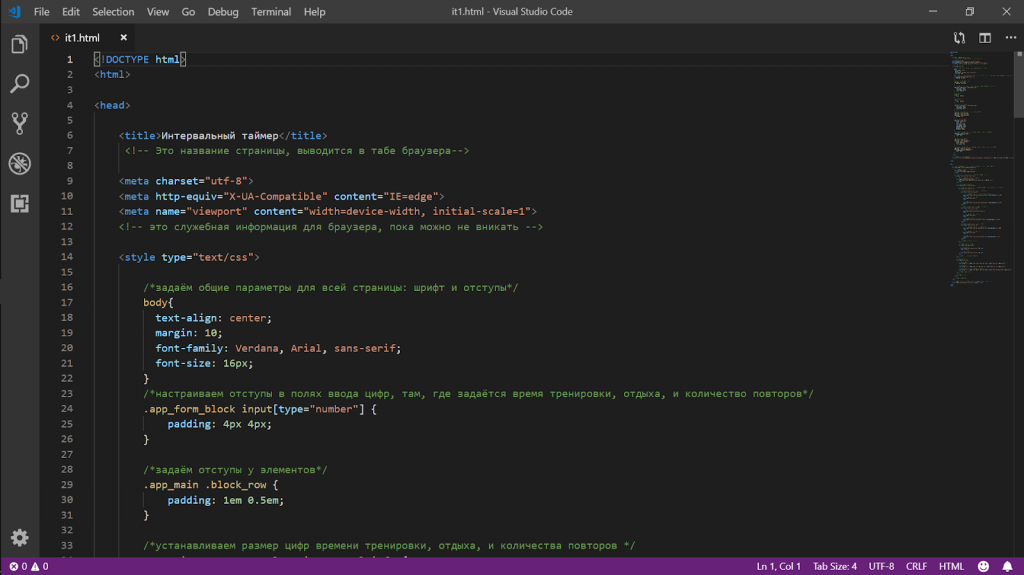


Рисунок 1 – «Среда разработки VS Code»

VS Code распознаёт почти все существующие языки программирования, самостоятельно или с помощью плагинов, и форматирует их соответствующим образом. Кроме этого, у него глубокая поддержка HTML, CSS, JavaScript и PHP — он проследит за парными тегами, закрытыми скобками и ошибками в командах.

Вот самые интересные возможности VS Code.

Умное авто дополнение показано на рисунке 2**.** Программа анализирует, какую команду вы хотите ввести, и предлагает закончить фразу за вас, с подсказками и объяснением. Удобно, если вы забыли порядок следования переменных или как точно звучит нужная команда:

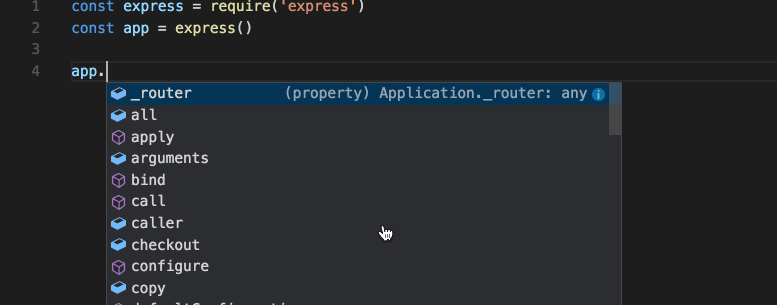


Рисунок 2 – «Умное авто дополнение»

Выполнение скриптов по шагам. Иногда нужно иметь возможность выполнить скрипт не сразу, а по шагам, например, чтобы понять, не зациклилась ли наша программа. Для этого используйте встроенный дебаггер — это программа, которая следит за кодом, ищет в нём ошибки и позволяет выполнять его поэтапно.

Множественное выделение и поиск. Чтобы поменять много одинаковых значений переменных или найти все одинаковые слова или команды, VS Code использует свой алгоритм обработки показано на рисунке 3. Благодаря этому редактировать код становится проще, а замена функций или переменных происходит быстрее.

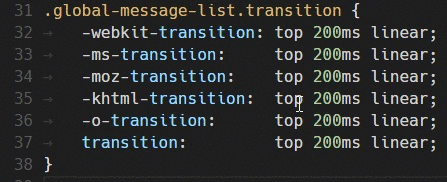


Рисунок 3 – «Алгоритм обработки»

Мультикурсор помогает вводить одинаковые значения сразу на нескольких строках показано на рисунке 4.

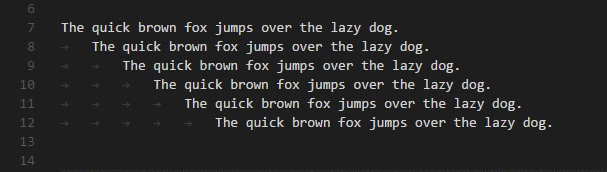


Рисунок 4 – «Мультикурсор»

Найденные одинаковые слова и команды можно тут же заменить на другие. Навигация по коду и описания функций показано на рисунке 5. Когда пишешь большую программу, легко забыть то, что делал в начале — как работает функция или какого типа переменная используется в этом месте. Чтобы этого избежать, VS Code может показывать саму функцию, описание переменной или какие параметры передаются при вызове команды. Ещё это пригодится, если код достался вам по наследству от прошлого разработчика и нужно быстро понять, какие куски кода за что отвечают и как работают:

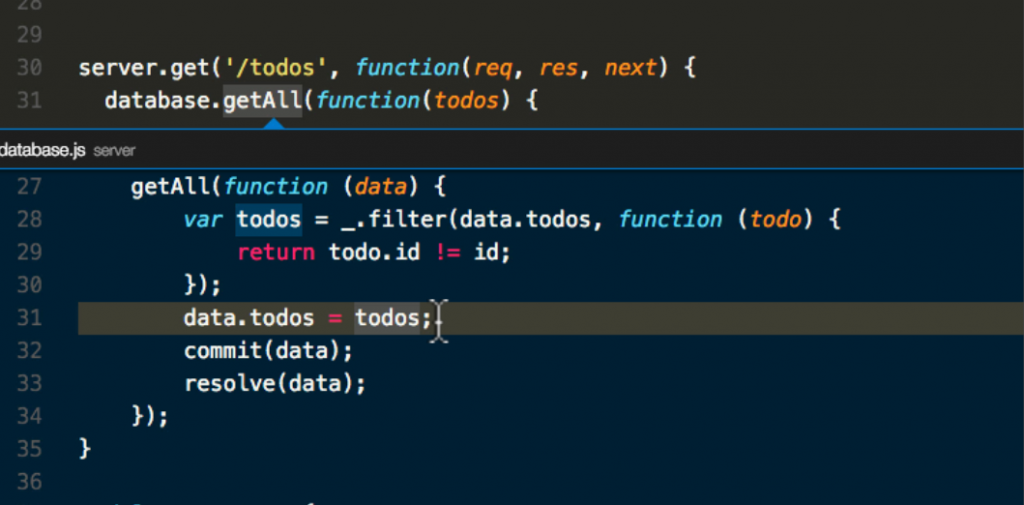


Рисунок 5 – «Навигация по коду»

Сразу после установки VS Code не умеет показывать результаты работы кода, когда мы делаем веб-страницы. Это можно исправить с помощью расширения Live HTML Previewer. Для этого заходим в раздел «Extensions», щёлкая на последнем значке на панели слева или нажимая Ctrl+Shift+X, и начинаем писать «Live HTML Previewer» в строке поиска.

После установки и запуска расширения можно будет сразу видеть, как ваш HTML-код и CSS влияют на внешний вид и поведение страницы. Это особенно удобно, когда строишь сайт с нуля и хочешь сразу понимать, что происходит.

**2 СПЕЦИАЛЬНАЯ ЧАСТЬ**

## **Реализация структуры создания «Веб-сайта»:**

Структура сайта «Техник-Сервис» состоит из нескольких блоков с данными, а именно:

1. Head (шапка) сайта содержит логотип и несколько ссылки для перехода по сайту, такие как «Каталог» - на этом листе предлагается прайс технического сервиза по категориям, «Главная», «Авторизоваться», «Регистрироваться» и для авторизованных пользователей; «Заявка», - реализована функция отправки и удаление заявки «Логин» (телефон) – осуществляет переход в личный кабинет, «Выход» - закрывает сессию пользователя.
2. Body (тело) сайта содержит название компании и графическое оформление.
3. Footer (подвал) сайта дублирует меню шапки, ссылка на информационный раздел ООО «Техник-Сервис»

## **Разработка программного продукта**

Веб-сайт состоит из пяти страниц и двух модельных окон. Главная страница сайта. На главной странице находится логотип сайта и меню, состоящее из 8 ссылок (Главная, поиск по каталогу, каталог, авторизация, регистрация, для авторизованного пользователя ссылки; логин, заявка, выход) показано на листинге 1.

Листинг 1. Ссылочное меню

 <a class="navbar-brand" id="admin" href="https://www.list-org.com/company/8289356" ><img class="my\_logo" src="imge/logo.png">ООО "ТЕХНИК-СЕРВИС</a>

<input class="form-control me-2" type="search" placeholder="Поиск по катологу" aria-label="Search" id="poisk">

<button class="btn btn-outline primary" type="button" onclick="poisck();" id="Search">Поиск<!DOCTYPE html>

 <a class="nav-link" aria-current="page" href="katalog.php" id="catlog">Каталог</li>

<a class="nav-link" href="index.php" id="dlavnay">Главная</a>

<a class="navbarbrand" href="user\_kabinet.php" ><span id="login\_user"></span></a>

<a class="navbar-brand" href="user\_korzina.php" >Заявка</a>

<a class="navbar-brand" href="index.php" id="exit\_button">Выход</a>

<a class="navbar-brand" href="#" data-bs-toggle="modal" data-bs-target="#exampleModal">Авторизация</a>

<a id ="show\_modal\_registration" class="navbar-brand" href="#" data-bs-toggle="modal" data-bs-target="#exampleModal2">Регистрация</a>

На веб страницы «Каталог» создано меню категорий, технического сервиса и несколько, часто выполняемых заявок показано на листинге 2.

Листинг 2. Каталог HTML

<main>

<div class="container-fluid">

<div class="row">

<nav id="sidebarMenu" class="col-md-3 col-lg-2 d-md-block bg-light sidebar collapse">

<div class="position-sticky pt-3">

<ul class="nav flex-column" id="ul\_menu">

</ul>

</div>

</nav>

<div class="col-md-9 ms-sm-auto col-lg-10 px-md-4">

<div class="row" id="load\_products">

</div><!-- /.row -->

</div>

</div>

</div>

</main>

<script src="js/js3.js"></script>

<script src="js/pisk.js"></script>

На странице подробнее присутствует подробная информация о выбранной сфере работ и возможность отправить её на страницу заявок. код разметки страницы HTML показан в листинге 3.

Листинг 3. Страница «Информация» HTML

<main><div class="container-fluid">

<div class="row">

<nav id="sidebarMenu" class="col-md-3 col-lg-2 d-md-block bg-light sidebar collapse">

<div class="position-sticky pt-3">

<ul class="nav flex-column" id="ul\_menu">

</ul>

</div>

</nav>

<div class="col-md-9 ms-sm-auto col-lg-10 px-md-4">

<div class="row" id="load\_products">

</div><!-- /.row -->

</div>

</div>

</div>

</main>

<script src="js/js3.js"></script>

<script src="js/pisk.js"></script>

Заявка

На страницы заявок, оформляются заявка и отправляется админу, либо удаляется, представлен код на листинге 4.

Листинг 4. Страница «Заявка» HTML

<div id="list\_tovar\_korzina">

</div>

</div>

<form class="form-inline" id="zacaz">

<div class="form-group mb-2">

<div class="alert alert-danger" role="alert" id="message\_zacaz"></div>

<label for="staticEmail2"  id="summ" class="sr-only">Итоговая сумма заказа</label>

<input disabled type="text" readonly class="form-control-plaintext" id="all\_summ\_korzina" value="0">

</div>

<button  type="submit" class="btn btn-primary mb-2" id="add\_all\_orders" >Оформить заказ</button>

</form>

<br>

</div>  
</div>

Полный код программного продукта можно увидеть в приложении 1.

## **2.3 Тестирование и отладка**

Тестирование — это процесс выполнения программы, целью которой является выявление ошибок в системе.

Отладка - процесс поиска, локализации и исправления ошибок в программе.

Тестирование — это процесс выполнения ПО системы или компонента в условиях анализа или записи получаемых результатов с целью проверки (оценки) некоторых свойств тестируемого объекта.

Тестирование — это процесс анализа пункта требований к ПО с целью фиксации различий между существующим состоянием ПО и требуемым (что свидетельствует о проявлении ошибки) при экспериментальной проверке соответствующего пункта требований.

Тестирование — это контролируемое выполнение программы на конечном множестве тестовых данных и анализ результатов этого выполнения для поиска ошибок.

Шаги процесса задаются тестами.

Каждый тест определяет:

* · Свой набор исходных данных и условий для запуска программы.
* · Набор ожидаемых результатов работы программы.

Целью проектирования тестовых вариантов является систематическое обнаружение различных классов ошибок при минимальных затратах времени и стоимости.

Тестирование обеспечивает:

* Обнаружение ошибок.
* Демонстрацию соответствия функций программы ее назначению.
* Демонстрацию реализации требований к характеристикам программы.
* Отображение надежности как индикатора качества программы.

Цель тестирования противоположна целям других этапов разработки. Его целью является нахождение ошибок. Успешным считается тест, нарушающий работу ПО. Все остальные этапы разработки направлены на предотвращение ошибок и недопущение нарушения работы программы.

Тестирование никогда не доказывает отсутствия ошибок. Отсутствие ошибок может указывать как на безупречность программы, так и на неэффективность или неполноту тестов.

Тестирование не повышает качества ПО - оно указывает на качество программы, но не влияет на него.

Виды тестирования.

Тестирование - самая популярная методика повышения качества, подкрепленная многими исследованиями и богатым опытом разработки коммерческих приложений. Существует множество видов тестирования: одни обычно выполняют сами разработчики, а другие - специализированные группы. Виды тестирования перечислены ниже:

* Блочным тестированием называют тестирование полного класса, метода или небольшого приложения, написанного одним программистом или группой, выполняемое отдельно от прочих частей системы.
* Тестирование компонента - это тестирование класса, пакета, небольшого приложения или другого элемента системы, разработанного несколькими программистами или группами, выполняемое в изоляции от остальных частей системы.
* Интеграционное тестирование — это совместное выполнение двух или более классов, пакетов, компонентов или подсистем, созданных несколькими программистами или группами.
* Регрессивным тестированием называют повторное выполнение тестов, направленное на обнаружение дефектов в программе, уже прошедшей этот набор тестов.
* Тестирование системы - это выполнение ПО в его окончательной конфигурации, интегрированного с другими программными и аппаратными системами.

Фазы тестирования.

Реализация тестирования делится на три этапа:

* Создание тестового набора (test suite) путем ручной разработки или автоматической генерации для конкретной среды тестирования (testing environment).
* Прогон программы на тестах, управляемый тестовым монитором (test monitor, test driver) с получением протокола тестирования (test log).
* Оценка результатов выполнения программы на наборе тестов с целью принятия решения о продолжении или остановке тестирования.

Тестирование проводилось в рамках предприятия ООО ЧОП «Палладин-нк» сотрудниками фирмы. В ходе его проведения были выявлены и в дальнейшем в процессе отладки устранены проблемы отображения представлений данных.

## **2.4 Руководство пользователя**

При переходе на сайт «Техник-Сервис» открывается главная страница, на которой находится верхнее ссылочное меню для навигации по сайту показано на рисунке 6.

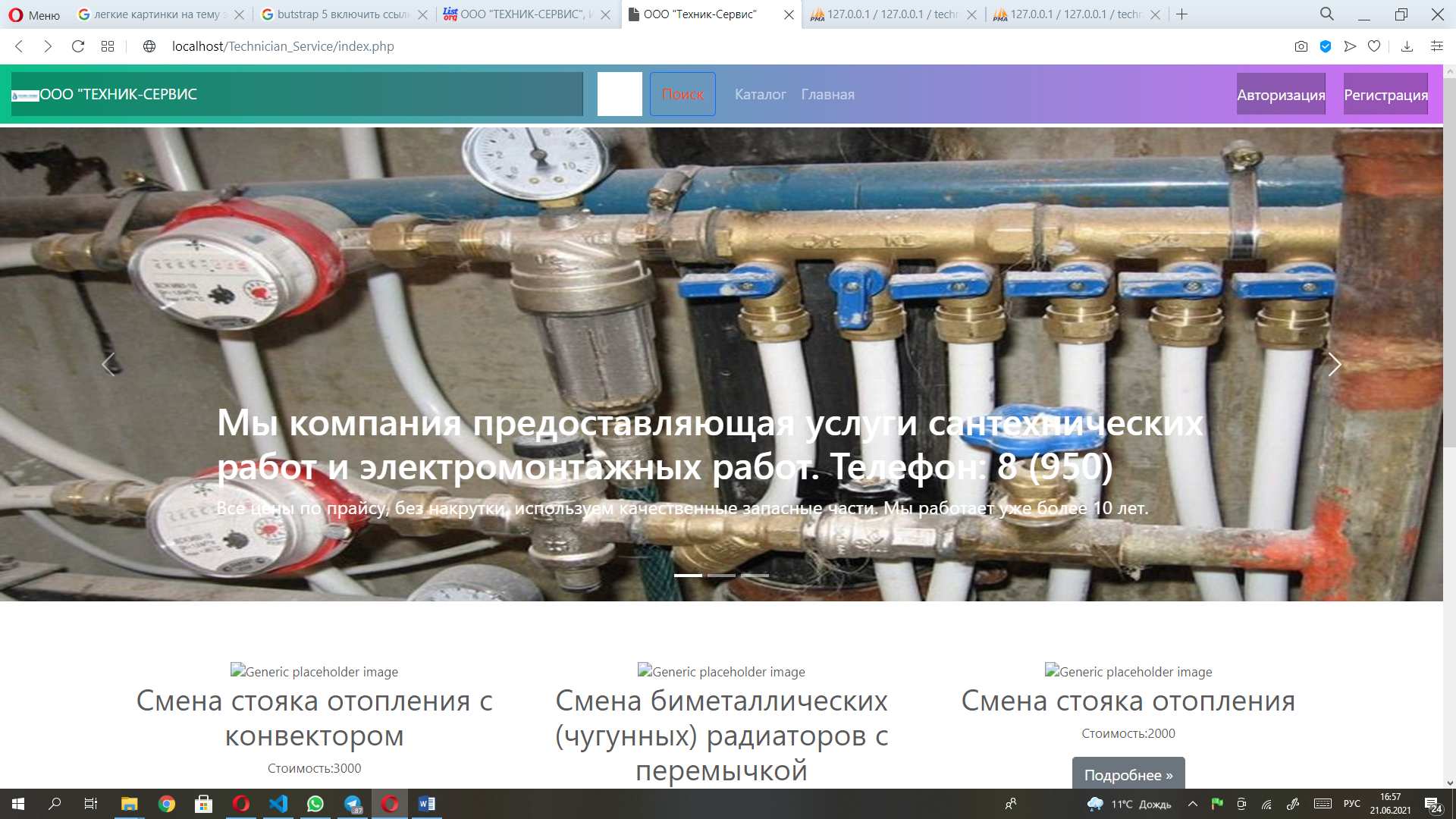


Рисунок 6 – «Главная страница»

Кликая по кнопке «Каталог» осуществляется переход на соответствующую страницу сайта, где находится меню по категориям с прайсом на различные виды работ и переходом на страницу подробнее показано на рисунке 7.

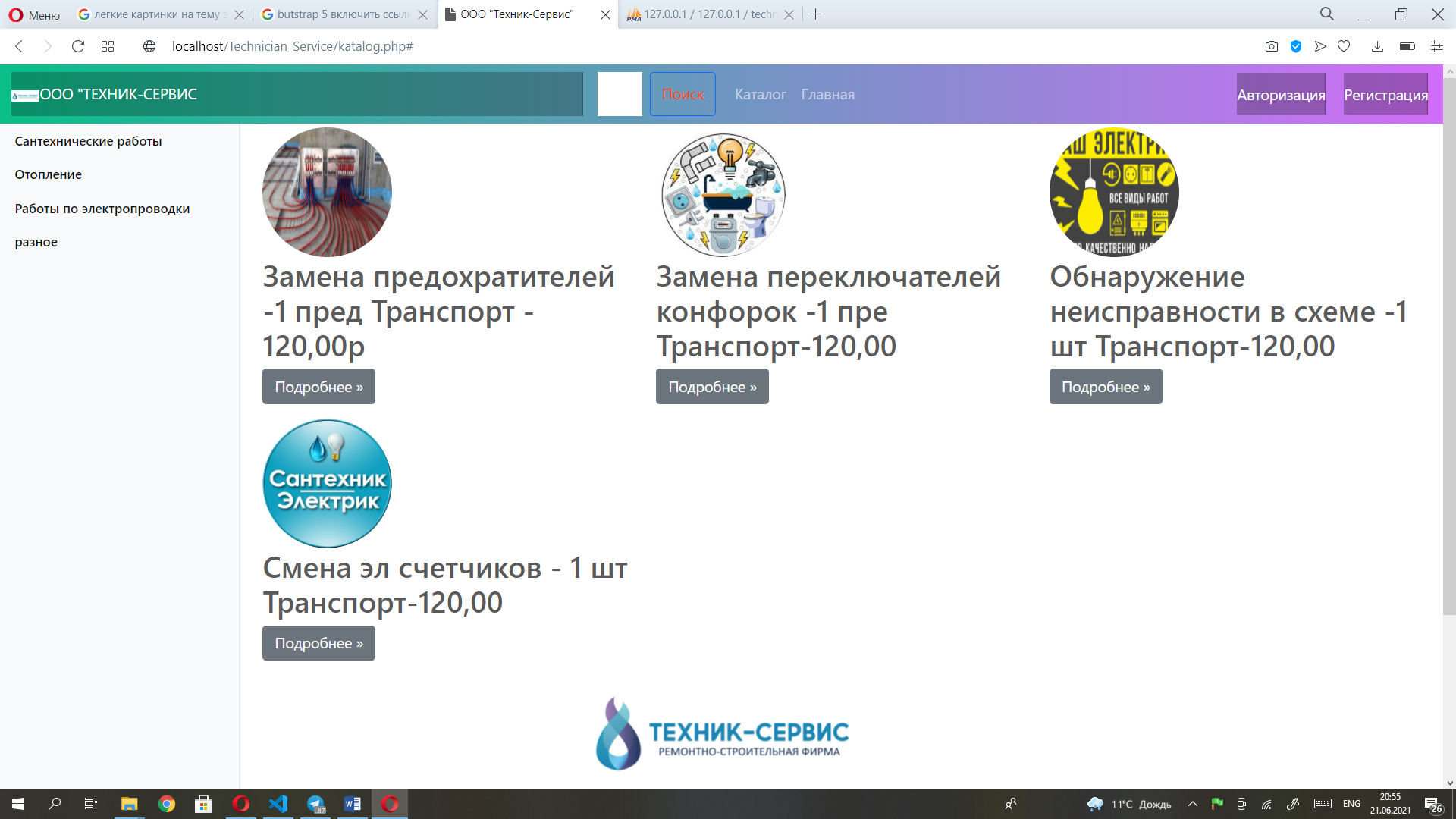


Рисунок 7 – «Каталог»

Переходя по кнопке «Подробнее» открывается одноимённая страница показано на рисунке 8.

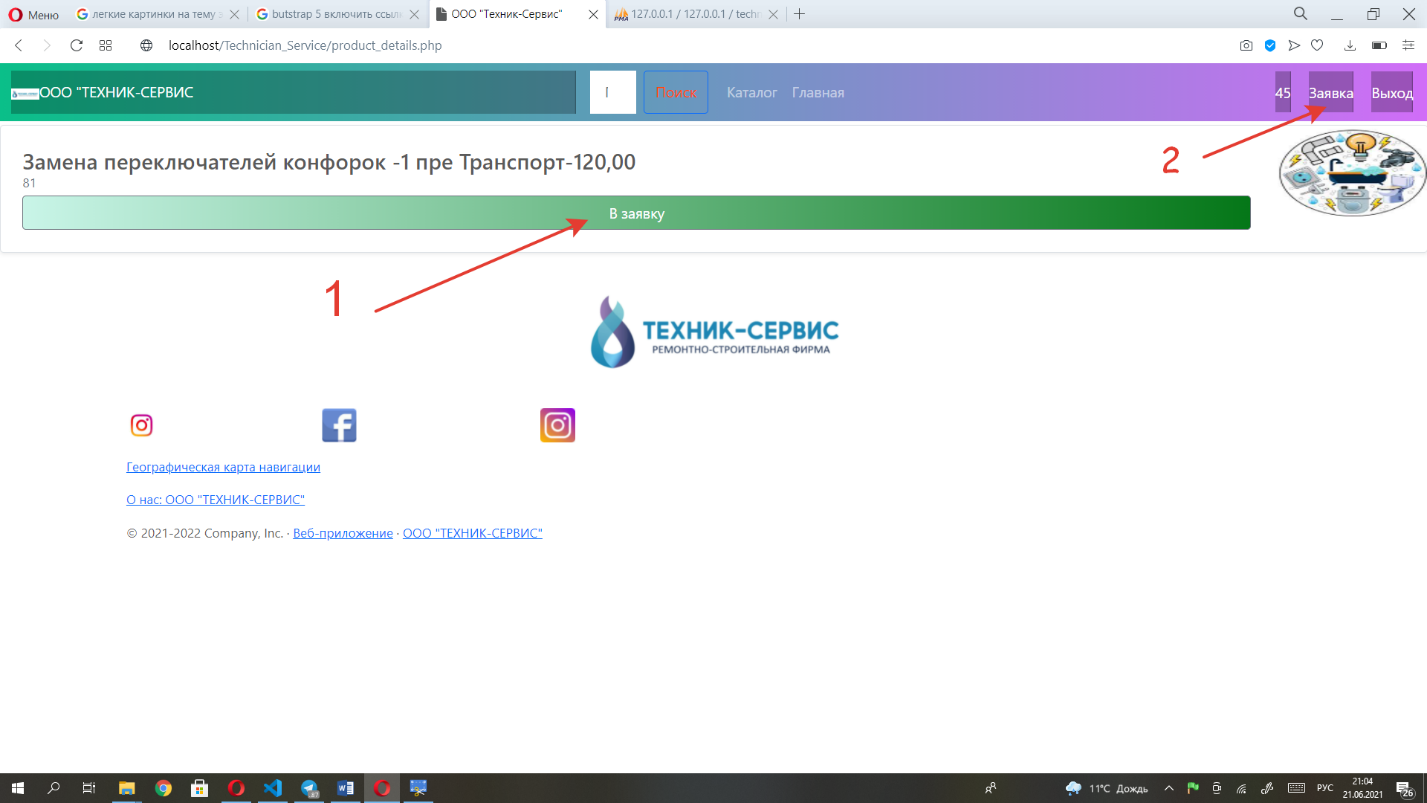


Рисунок 8 – «Подробнее»

При клике по кнопке «в заявку», стрелка 1. Кнопка исчезает, кликаем по ссылке, стрелка 2. вы попадаете на одноимённую страницу, где подтверждаете её либо удаляете показано на рисунке 9.

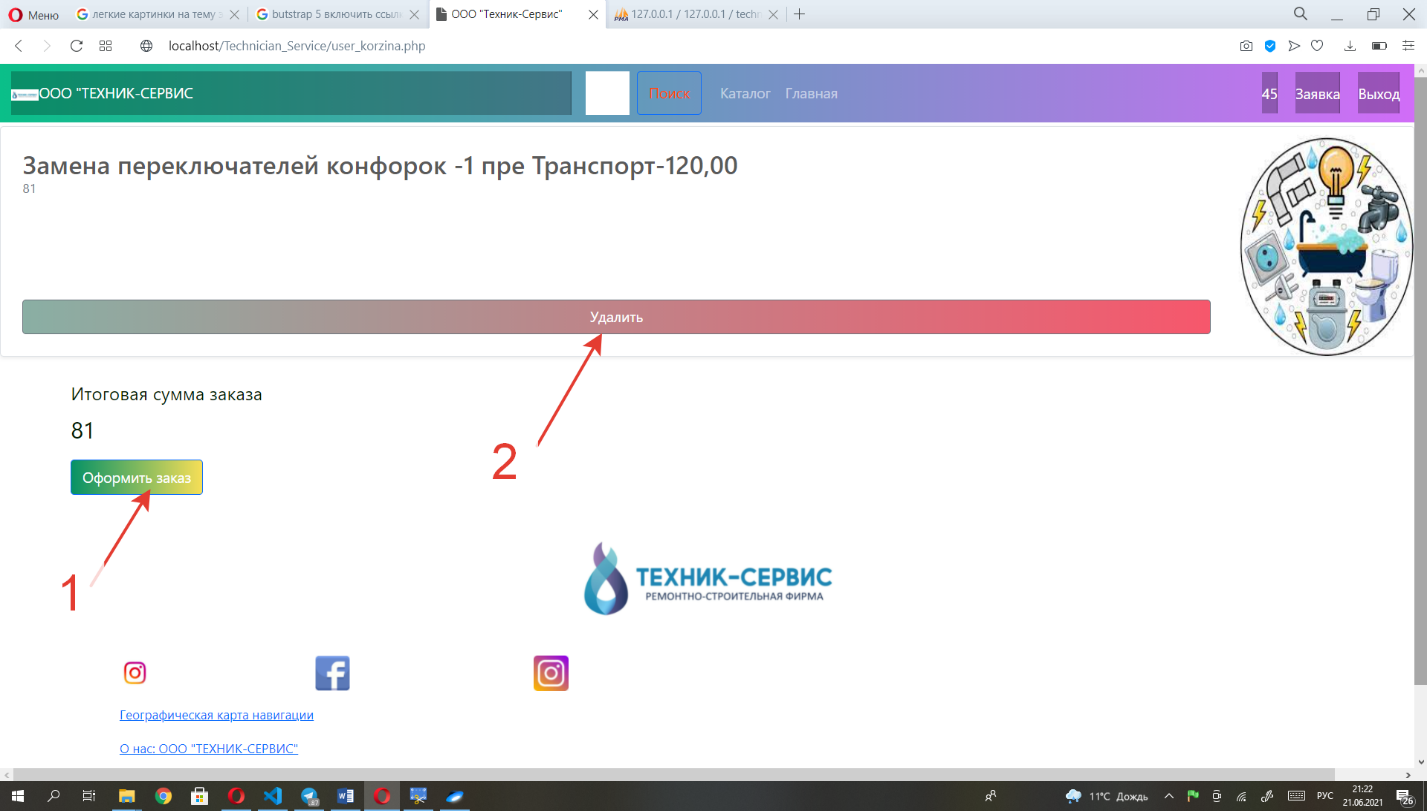


Рисунок 9 – «Заявка»

Страница для руководителя «Админ» показана на листинге 10.

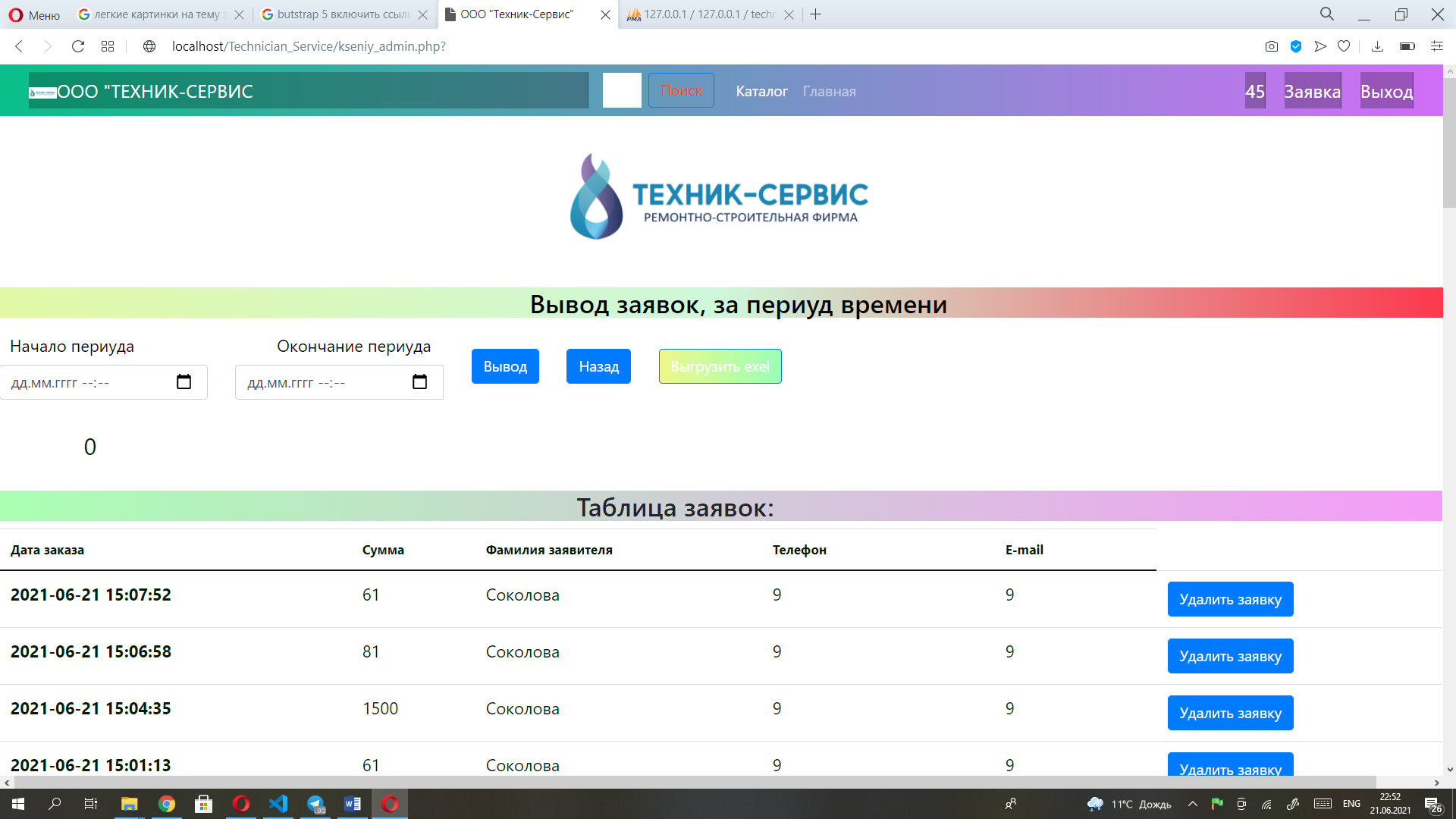


Рисунок 10 – «Админ»

# **ЗАКЛЮЧЕНИЕ**

Во время производственной практики были отработаны и закреплены навыки, изученные в теоретической части учебного процесса. Приобретены умения и навыков практической работы по присваиваемой квалификации и избранной [специальности](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C).

В результате прохождения практики закреплены следующие знания:

* основные виды и процедуры обработки информации, модели и методы решения задач обработки информации (генерация отчетов, поддержка принятия решений, анализ данных, искусственный интеллект, обработка изображений);
* cервисно-ориентированные архитектуры, CRM-системы, ERP-системы;
* объектно-ориентированное программирование;
* спецификации языка, создание графического пользовательского интерфейса, файловый ввод-вывод, создание сетевого сервера и сетевого клиента;
* платформы для создания, исполнения и управления информационной системой;
* основные процессы управления проектом разработки.

А также отработаны следующие навыки:

* осуществлять математическую и информационную постановку задач по обработке информации, использовать алгоритмы обработки информации для различных приложений;
* решать прикладные вопросы интеллектуальных систем с использованием статических экспертных систем, экспертных систем реального времени;
* использовать языки структурного, объектно-ориентированного программирования и языка сценариев для создания независимых программ, разрабатывать графический интерфейс приложения;
* создавать проект по разработке приложения и формулировать его задачи, выполнять управление проектом с использованием инструментальных средств

Таким образом в результате прохождения производственной практики получен практический опыт:

* использования инструментальных средств обработки информации;
* участия в разработке технического задания;
* формирования отчетной документации по результатам работ;
* использования стандартов при оформлении программной документации;
* программирования в соответствии с требованиями технического задания;
* использования критериев оценки качества и надежности функционирования информационной системы;
* применения методики тестирования разрабатываемых приложений;
* управления процессом разработки приложений с использованием инструментальных средств.

Проект «Веб-сайт» предоставит возможность не только быстрого доступа, но и удобной работы с необходимой информацией, что не вызывает сложностей как у IT-специалиста, так и у рядового пользователя интернета, не смотря на разный спектр возможностей.

По итогам проведённой работы, с последующим тестированием приложений позволяет сделать следующие выводы, что данные «Веб-сайты» помогут сделать деятельность быстрой и коммуникабельной.

Цель достигнута, задачи выполнены.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Джордж Шлосснейгл. Профессиональное программирование на HTML, 2015
2. Уильям Стейнмец, Брайан Вард. 75 готовых решений для вашего web-сайта на HTML, 2016
3. Мишель Е. Дэвис и Джон А. Филипс. Изучаем CSS и MySQL, 2017
4. Николай Прохоренок. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера, 2017
5. James Bender Professional Test Driven Development with C#; [текст] - Москва, 2018. - 384 c.
6. Jesse Liberty Programming C#; [текст] - Москва, 2018. - 688 c.
7. Рихтер, Джефри CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#; Питер - Москва, 2018. - 656 c.

**ПРИЛОЖЕНИЯ**

Листинг 5. Страница «Админ» HTML

<link href="Css/style(t).css" rel="stylesheet">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" />

<div>

<main class="container-fluid nightmode text-center" id="clock2">

<time id="date" class="clocktext"> </time>

<time id="time" class="clocktext"> <span> </span> </time>

</main>

</div>

<!--За промижуток времени-->

<link href="Css/style(t).css">

<link href="Css/style.css">

<div class="peri">

</div>

<div class="col-auto">

<a class="navbar-brand"></a><img class="my\_brend" src="imge/JPEG.jpg"></a>

</div><br>

<form class="row gy-2 gx-3 align-items-center">

<h3 id="lables">Вывод заявок, за периуд времени</h3>

<tr>

<div class="col-auto">

<label class="form-check-labelz">Начало периуда</label><br>

<input class="form-control" type="datetime-local" value="2021-00-00T00:00:00" id="start\_data">

</div>

<div class="col-auto">

<label class="form-check-labele">Окончание периуда</label>

<div class="input-group">

<input class="form-control" type="datetime-local" value="2021-00-00T00:00:00" id="finish\_data"><br>

</div>

</div>

<div class="col-auto">

<!-- <input type="button" class="btn btn-primary" value="Вывод, за промежуток времени " id="Output"> -->

<button type="button" class="btn btn-primary" id="Output" onclick="add\_time();">Вывод</button>

</div>

<div class="col-auto">

<button type="button" class="btn btn-primary" id="return" onclick="tab\_time();">Назад</button>

</div>

<div class="col-auto">

<button class="btn btn-primary" id="Exel" onclick="to\_excel();">Выгрузить exel</button>

</div>

</form><br>

<div class="col-auto">

<input disabled type="text" readonly class="form-control-plaintext" id="all\_summa\_zayvoc" value="0">

<label for="staticEmail2" id="summa" class="sr-only">Итоговая сумма заказа</label>

</div><br>

<div class="pe" id="per\_time">

<h3>Таблица заявок:</h3>

</div>

<link href="Css/style(t).css" rel="stylesheet">

<div class="tabel" id="zacaz\_history">

<td><input type="checkbox" id="inlineCheckbox1" value="true" disabled><br></td>

</div>

<br>