



МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖИ РЕСПУБЛИКИ КРЫМ
ГБОУВО РК «КРЫМСКИЙ ИНЖЕНЕРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ ФЕВЗИ ЯКУБОВА»

Факультет экономики, менеджмента и информационных технологий
Кафедра прикладной информатики

Направление подготовки 09.03.03 Прикладная информатика
Профиль «Прикладная информатика в информационной сфере»

КУРСОВОЙ ПРОЕКТ

по дисциплине «Программирование и поддержка веб-приложений»

НА ТЕМУ

«Разработка веб-приложения «Сервис по ремонту бытовой техники»

Малыгин
02.03.2022
Малыгин

Студента IV курса
группы И-2-18
очной формы обучения
Малыгина Дмитрия
Владимировича

Малыгин

(подпись)

Руководитель:
к.т.н., доц. Абдураимов Л. Н.

Абдураимов
(подпись)

Симферополь – 2021

Задание
к курсовому проекту по дисциплине
«Программирование и поддержка веб-приложений»
студента группы И-2-18
Матыгина Дмитрия Владимировича

(Ф.И.О. студента полностью в родительном падеже)

Тема: **Разработка веб-приложения «Сервис по ремонту бытовой техники»**

Цель данной работы – приобретение студентами практических навыков по проектированию и разработке веб-приложений.

Пояснительная записка должна включать следующие разделы:

ВВЕДЕНИЕ

ГЛАВА 1. ОСОБЕННОСТИ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ

1.1. Общая характеристика, классификация веб-сайтов (веб-приложений)

1.2. Этапы создания веб-сайтов (веб-приложений)

1.3. Средства разработки веб-сайтов (веб-приложений)

1.4. Системный анализ предметной области

1.5. Проектирование базы данных

Выводы по главе 1

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

2.1. Разработка базы данных

2.2. Реализация компонентов веб-сайта (веб-приложения)

2.3. Разработка административной панели

2.4. Установка веб-сервера

2.5. Разворачивание веб-сайта (веб-приложения)

2.6. Тестирование продукта

Выводы по главе 2

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЯ

Дата получения задания: 13.09.21 г.

Студент













(подпись)

Дата сдачи на кафедру: 02.03.22

Научный руководитель

(подпись)

Календарный план выполнения и защиты курсового проекта

Параграф	Дата сдачи		Оценка	Подпись руководителя
	Плановая	Фактическая		
Общая характеристика, классификация веб-сайтов (веб-приложений)	27.09.21	27.09.21	отн	
Этапы создания веб-сайтов (веб-приложений)	04.10.21	04.10.21	отн	
Средства разработки веб-сайтов (веб-приложений)	11.10.21	11.10.21	отн	
Системный анализ предметной области	25.10.21	25.10.21	отн	
Проектирование базы данных	01.11.21	18.11.21	отн	
Разработка базы данных	08.11.21	13.12.21	отн	
Реализация компонентов веб-сайта (веб-приложения)	22.11.21	29.12.21	отн	
Разработка административной панели	29.11.21	24.01.22	отн	
Установка веб-сервера	03.12.21	15.02.22	отн	
Разворачивание веб-сайта (веб-приложения)	10.12.21	15.02.22	отн	
Тестирование продукта	15.12.21	01.03.22	отн	

Дата защиты курсового проекта 02.03.22 г.

Оценка

хорошо

Научный руководитель

(подпись)

к.т.н., доц. Абдурайимов Л. Н.

(Ф.И.О. руководителя)

АННОТАЦИЯ

Малыгин Д. В. Разработка веб-приложения «Сервис по ремонту бытовой техники»

В курсовом проекте представлена технология разработки веб-приложения для сервиса по ремонту бытовой техники. Приложение разработано с использованием языка программирования Python на базе технологий Apache (HTTP Server), JavaScript, MySQL. Это приложение может успешно применяться в любых коммерческих проектах, направленных на ремонт бытовой техники.

Ключевые слова: JavaScript, Apache (HTTP Server), MySQL, веб-приложение, Python.

ANNOTATION

Malygin D. V. Development of a web application "Service for the repair of household appliances"

The course project presents the technology for developing a web application for a home appliance repair service. The application was developed using the Python programming language based on Apache (HTTP Server), JavaScript, MySQL technologies. This application can be successfully used in any commercial projects aimed at repairing household appliances.

Keywords: JavaScript, Apache (HTTP Server), MySQL, web application, Python.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В ОБЛАСТИ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ	7
1.1. Общая характеристика, классификация веб-сайтов	7
1.2. Этапы создания веб-сайтов.....	8
1.3. Сравнительный анализ сервисов по ремонту бытовой техники	11
1.4. Системный анализ предметной области	14
1.5. Проектирование базы данных	15
Выводы к главе 1	16
ГЛАВА 2. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ СЕРВИСА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ.....	17
2.1. Реализация программного продукта	17
2.2. База данных программного продукта.....	21
2.3. Описание интерфейса приложения	26
Вывод к главе 2	29
ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	31

ВВЕДЕНИЕ

Актуальность темы. В современном мире сложно представить квартиру или дом без многих бытовых приборов, разнообразие которых крайне велико. Стоимость домашней техники очень высокая и, как правило, полную замену устройств проводят не так часто, когда техника полностью выходит из строя. Во время работы такие устройства нуждаются в ремонте и обслуживании.

Спрос на услуги сервиса по ремонту бытовой техники будет всегда. Поэтому наличие у мастерской собственного веб-сайта просто необходимо. Ведь такое приложение позволит предоставлять клиентам различную информацию. Как о новых услугах, так и об эффективности существующих.

Так же на сайте легко публиковать информацию, которую невозможно разместить ни в одном другом виде традиционной рекламы. Веб-сайт – это инструмент, с помощью которого можно привлечь потенциальных клиентов и упростить начальный процесс заказа ваших услуг.

Потенциальные клиенты с помощью каталогов и поисковиков, по ключевым словам, найдут сайт, ознакомятся с услугами и условиями и могут захотеть стать вашими клиентами.

Цель курсового проекта – разработать веб-приложение на тему «Сервис по ремонту бытовой техники».

Для достижения цели были поставлены следующие задачи:

1. Дать общую характеристику, классификацию веб-сайтов;
2. Провести системный анализ предметной области;
3. Разработать веб-приложение «Сервис по ремонту бытовой техники».

Практическая значимость полученных результатов заключается в разработке веб-сервиса, предназначенного для использования в компаниях, занимающихся ремонтом бытовой техники. Предоставляемый приложением функционал позволяет обрабатывать заявки на ремонт бытовой техники.

Объектом исследования является процесс разработки веб-приложений.

Предметом исследования – разработка веб-приложения для сервиса по ремонту бытовой техники.

Структура и объем курсового проекта. Курсовой проект состоит из введения, 2 глав, выводов к главам, заключения, списка использованных источников.

В введении рассмотрена актуальность веб-приложения для сервиса по ремонту бытовой техники, поставлены цели и задачи, определены предмет и объект исследований, выделена практическая значимость курсового проекта.

В первой главе рассмотрены виды веб-приложений по шаблону разработки, а после был определён тип разрабатываемого веб-приложения для сервиса по ремонту бытовой техники.

Вторая глава содержит этап проектирования в парадигмах UML и программная реализация веб-приложения для сервиса по ремонту бытовой техники.

В заключении сделаны выводы по курсовому проекту и приведен план дальнейшего развития веб-приложения для сервиса по ремонту бытовой техники.

Объем курсового проекта составляет 30 страниц, из них 24 страница основного текста, 18 рисунков, 5 листингов. Список использованных источников содержит 12 наименований.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В ОБЛАСТИ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ

1.1. Общая характеристика, классификация веб-сайтов

Веб-сайт (от английского «web-site»: «web» – паутина, сеть; «site» – место, местоположение) – это структурированная специальным образом информация, которая размещена на сервере и является открытой пользователям сети как для свободного, так и для авторизируемого или ограниченного доступа.

Совокупность всех сайтов составляет Всемирную паутину, объединяющую в единое целое части информации всего мирового сообщества – объединения планетарного масштаба и базу данных. Протокол HTTP (Hypertext Transfer Protocol — протокол передачи гипертекста) специально был разработан для того, чтобы обеспечить клиенту прямой доступ к сайтам на серверах [1].

Сайт является набором определенных файлов – страниц – определенного формата. Страницы сайтов – это файлы с текстом, который содержит команды на языке HTML (Hypertext Markup Language – язык разметки гипертекста). Данные файлы при загрузке посетителем на компьютер обрабатываются посредством браузера и выводятся на средство отображения компьютера (монитор, экран смартфона, экран КПК и т.д.) Возможности языка HTML позволяют править, различать его функциональные элементы, создавать гиперссылки, а также включать мультимедийные элементы: изображения, звукозаписи и т.д.

Существует большое количество признаков, по которым можно произвести классификацию сайтов. Классификация сайтов является необходимой, так как зачастую веб-мастеру приходится сталкиваться с непониманием заказчика того, какой сайт должен получиться в итоге. Поэтому целесообразно выделение базовых признаков каждого вида (типа). Кроме того, у сайтов разного типа различная, специфическая структура[3].

В первую очередь, как правило, выделяют сайты коммерческие и некоммерческие. Конечной целью функционирования коммерческого сайта всегда является прибыль, в то время как некоммерческие сайты выполняют, в основном, информационную функцию. Коммерческим можно назвать не только сайт, предназначенный для осуществления продаж либо предоставления услуг, но и промо-сайты, популяризирующие бренд, товар или фирму. Некоммерческие сайты чаще всего создаются общественными организациями, политическими партиями, государственными организациями. Также некоммерческими сайтами можно назвать ресурсы, предназначенные для обмена мнениями и организации общения людей[2].

Основные виды сайтов:

- Блог
- Форум
- Wiki-ресурс
- Flash-сайт
- Доска объявлений
- Визитка
- Сервис
- Портал
- Интернет-магазин
- Биржа
- Простой сайт

1.2. Этапы создания веб-сайтов

Следование всем этапам процесса создания сайтов позволяет правильно организовать работу всех специалистов, что гарантирует конечное качество проекта.

Основные этапы создания веб-сайта:

1. Анализ целевой аудитории и конкурентов.

Первый этап разработки — аудит и анализ. В первую очередь собирается информация о деятельности предприятия, оценивается текущая стратегия

продвижения, фронт проведенных работ. Отдельно внимание уделяется целевой аудитории и основным конкурентам. Для создания качественного сайта важно изучить конкурентов в нише, а также основных потребителей продукта [6].

2. Постановка целей.

После анализа обязательно проводится формирование целей и задач проекта. Важно сразу определить для чего будет использоваться ресурс: для продаж, информирования потребителей и получения обратной связи, работы с репутацией бренда и т. д.

3. Разработка технического задания.

Немаловажный этап разработки сайта — создание проектной документации и технического задания. Во время написания ТЗ важно учитывать каждый аспект работы и прописывать все процессы — качество работы специалистов и проекта в целом напрямую зависит от проработки технического задания. На данном этапе определяется общая концепция проекта, планируется его функциональность, общий стиль и особенности дизайна.

4. Проектирование и прототипирование.

После проработки проектной документации создается прототип сайта. Основная задача прототипа — разработать интуитивно понятный и функциональный интерфейс, удобный для конечного пользователя. При этом требуется помнить, что от принятого прототипа зависит конечный результат работы и жизнеспособность сайта в целом[3].

5. Проработка макетов дизайна.

Следующим этапом разработки сайта является создание макетов страниц согласно ТЗ и готового прототипа. Дизайнеры отрисовывают указанное в ТЗ количество вариаций главной страницы, категорий, каталога и товарных карточек. Кроме того, уделяется внимание дизайну каждого элемента на страницах: от блоков и форм поиска до слайдеров с фото.

6. Вёрстка и интеграция с CMS.

Готовые макеты отправляются верстальщикам и программистам. Основная задача верстки — обеспечить адаптивность страниц ко всем веб-браузерам и мобильным гаджетам. Важно, что веб-сайт корректно отображался на всех популярных форматах экранов, независимо от их диагонали и разрешения.

Далее проводится работа над валидностью верстки и возможностью ее дальнейшей доработки при расширении проекта. Исходный код проверяется на наличие ошибок, подстраивается под современные стандарты качества и интегрируется в выбранную систему управления сайтом (CMS) [4].

7. Настройка основных модулей.

Одним из основных этапов создания сайта считается программирование — добавление в проект всех заложенных опций и функций. Задача программистов: разработать и внедрить заложенную в ТЗ функциональность, настроить основные модули, обеспечить корректную работу ресурса.

8. Интеграция с учётными системами и сервисами.

Отдельно проводится внедрение сторонних IT-продуктов для расширения функциональности сайта. Разработчики настраивают синхронизацию веб-сайта с CRM, подключают программу складского учета и бухгалтерию. Дополнительно проводится интеграция мессенджеров, подключаются онлайн-консультанты, настраиваются модули почтовой рассылки.

9. Наполнение контентом SEO.

Следующие этапы разработки сайта — базовая оптимизация и работа с контентом. Согласно рекомендациям SEO-специалиста, разрабатывается структура и посадочные страницы сайта, внедряются коммерческие факторы и т.д. Также на данном этапе возможно проведение базовой SEO-оптимизации: распределение веса страниц сайта, добавление и оптимизация изображений, заполнение meta-данных.

10. Тестирование и запуск[10].

Порядок создания интернет-сайта предполагает проведение тестирования перед релизом. На локальной копии проекта или тестовом сервере оценивается работа основных функций, скорость загрузки страниц, удобство серфинга. Любые ошибки устраняются, код и структура при необходимости редактируются.

11. Поддержка и сопровождение.

После релиза сайт успешно собирает трафик в интернете и генерирует продажи. Однако для поддержания конкурентоспособности важно регулярно развивать проект. Услуга поддержки и сопровождения веб-сайта предполагает проведение технических работ или консультации сотрудников компании клиента[9].

Каждый этап напрямую влияет на конечный результат проекта, его функциональность и эргономику.

1.3. Сравнительный анализ сервисов по ремонту бытовой техники

Для принятия решения о выборе программного обеспечения были проведены обзор и анализ известных веб-сервисов.

1. Алиса-сервис

«Алиса-сервис» — авторизированный сервисный центр, команда которого больше 20 лет занимается ремонтом электронной и бытовой техники.

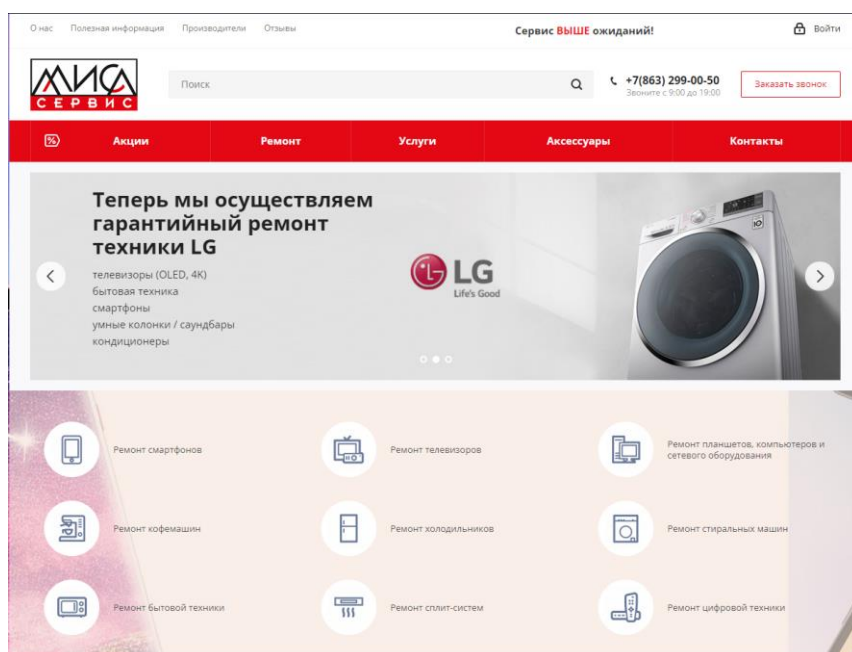


Рис. 1.1. Сервисный центр «Алиса-сервис»

«Алиса-сервис» является официальным центром по ремонту оборудования от Samsung, LG, Philips, Lenovo, Intel, Delonghi, Braun, Tefal. Также здесь берутся за восстановление девайсов других фирм.

В сервис можно обратиться, чтобы восстановить смартфон, планшет, компьютер, ноутбук, телевизор, магнитофон, кофеварку или кофемашину, пылесос, утюг, стиральную машину и многое другое.

Компания дает гарантию на оказанные услуги до 90 дней, а все возникающие вопросы всегда решает в пользу клиента.

При ремонте габаритной техники вы можете заказать ее забор и доставку курьерами сервиса.

Также стоит отметить, что в пространстве имеется собственный склад запасных частей, который регулярно пополняется новыми деталями.

2. Remteh

Сервис специализируется на предоставлении широкого спектра услуг в сфере ремонта бытовой техники, а также различного профессионального оборудования. Компания работает в сфере ремонта бытовой техники уже не первый год.

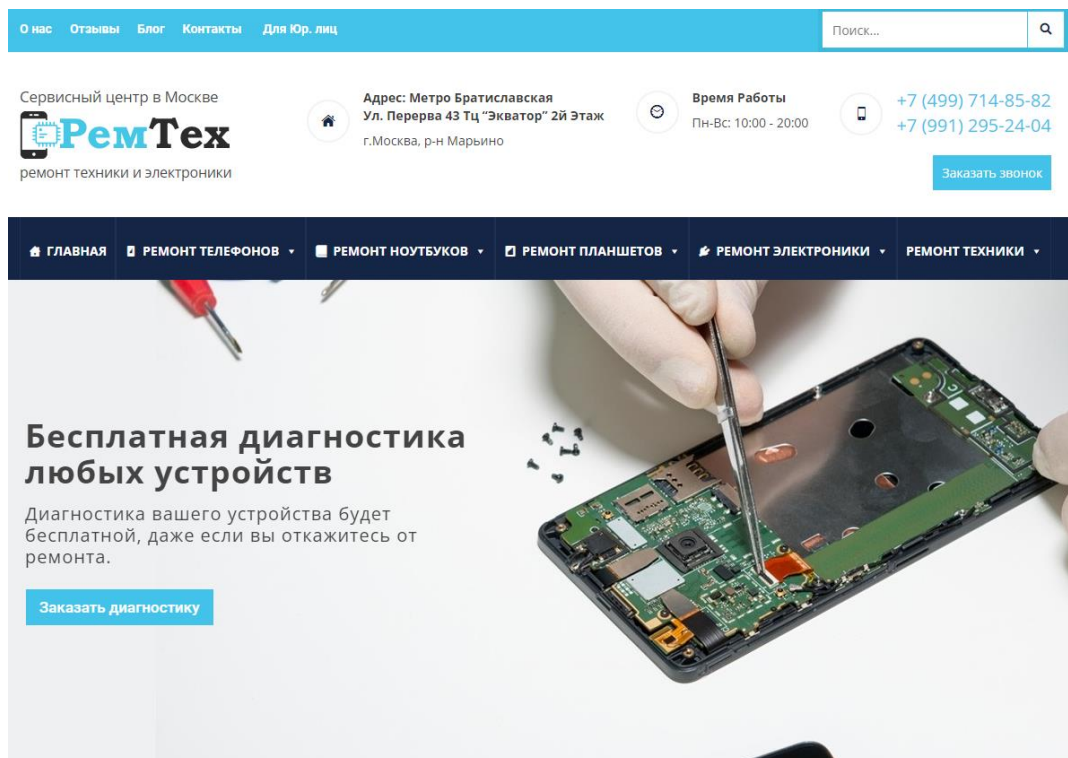


Рис. 1.2. Сервисный центр «Remteh»

Преимущества:

1. Широкий спектр услуг.
2. Исключительное качество всех работ, проводимых нашими специалистами (как ремонтниками, так и монтажниками).
3. Использование только самых современных инструментов для проведения ремонтных и монтажных работ.
4. Мы производим как гарантийный, так и постгарантийный ремонт бытовой техники в Симферополе.
5. Работаем на равных условиях как с физическими, так и юридическими лицами.
6. Предоставление бесплатных консультаций клиентам по вопросам, связанным с ремонтными работами, монтажными работами, а также сервисным обслуживанием техники.
7. Удобная система оплаты наших услуг.
8. Возможность наличного и безналичного расчета.
9. Индивидуальный подход ко всем нашим клиентам.
10. Умение идти навстречу заказчику в любых непредвиденных ситуациях.

3. Мультисервис

Компания рада предложить комплекс услуг по монтажу, ремонту и обслуживанию бытовой техники с выездом на дом. Высокое качество работ в короткие сроки. Принимают в работу мелкую и крупную бытовую технику - от простой до самой сложной, вне зависимости от года выпуска и завода-изготовителя.



Рис. 1.3. Сервисный центр «Мультисервис»

Преимущества:

1. Цены ниже рыночных.
2. Краткие сроки выполнения работы
3. Квалифицированный персонал
4. Ремонт техники прямо на дому

1.4. Системный анализ предметной области

Веб-сайт мастерской по ремонту бытовой техники должен отображать информацию об клиентах, исполнителях, и о предоставляемых услугах. Данное приложение призвано в некоторой степени автоматизировать деятельность мастерской.

Веб-приложение должно позволить пользователям отслеживать перемещения, заказов между структурными подразделениями, работниками фирмы. Собирать статистическую информацию о работниках сервисного центра, о количестве выполненных заказов, их стоимости, а также другую информацию [5].

Система позволит повысить как качество предоставляемых предприятием услуг, так и скорость их предоставления.

Также каждому заказу будет присваиваться определённый статус, он изменяется в зависимости от операций, произведённых над ним. Количество статусов не ограничено, администратор мастерской вправе сам решать, какой статус присвоить. Однако изначально будут обязательные статусы такие как: Принят, В ремонте, На согласовании, Готов, Выдан.

1.5. Проектирование базы данных

Проектирование баз данных — процесс создания схемы базы данных и определения необходимых ограничений целостности.

Основные задачи:

- Обеспечение хранения в БД всей необходимой информации.
- Обеспечение возможности получения данных по всем необходимым запросам.
- Сокращение избыточности и дублирования данных.
- Обеспечение целостности базы данных.

Концептуальное проектирование базы данных - это процесс создания модели используемой на предприятии информации, не зависящей от любых физических аспектов ее представления [4].

Логическое проектирование базы данных – это процесс создания модели используемой на предприятии информации на основе выбранной модели организации данных, но без учета типа целевой СУБД и других физических аспектов реализации.

Физическое проектирование является третьим и последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных [12].

Нормализация — это процесс организации данных в базе данных. Это включает создание таблиц и установление связей между этими таблицами в соответствии с правилами, предназначенными как для защиты данных, так и

для того, чтобы сделать базу данных более гибкой за счет устранения избыточности и непоследовательной зависимости.

Избыточные данные пустая трата дискового пространства и создает проблемы с обслуживанием. Если данные, которые существуют в нескольких местах, должны быть изменены, данные должны быть изменены точно так же во всех расположениях. Изменение адреса клиента гораздо проще реализовать, если эти данные хранятся только в таблице Клиентов и нигде в базе данных [11].

Выводы к главе 1

В первой главе курсового проекта было рассмотрено веб-приложение для сервисного центра по ремонту бытовой техники, а также основные методы и приемы в исследуемой теме. Был проведен анализ классификаций и характеристик веб-сайтов. Проведен сравнительный анализ подобных систем. Выявлены основные задачи веб-приложения. Описаны программные продукты, такие как: «Алиса-сервис», «Мультисервис», «Remteh». Проведено сравнение их преимуществ и недостатков. Помимо этого, проанализировали предметную область. А также в этой главе было проведено проектирование базы данных проекта.

ГЛАВА 2. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ СЕРВИСА ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ

2.1. Реализация программного продукта

Для реализации веб-сервиса необходимо установить среду разработки Visual Studio Code. На следующем этапе необходимо установить основную библиотеку Flask для языка программирования Python. На рисунке 2.1 изображена основная структура проекта.

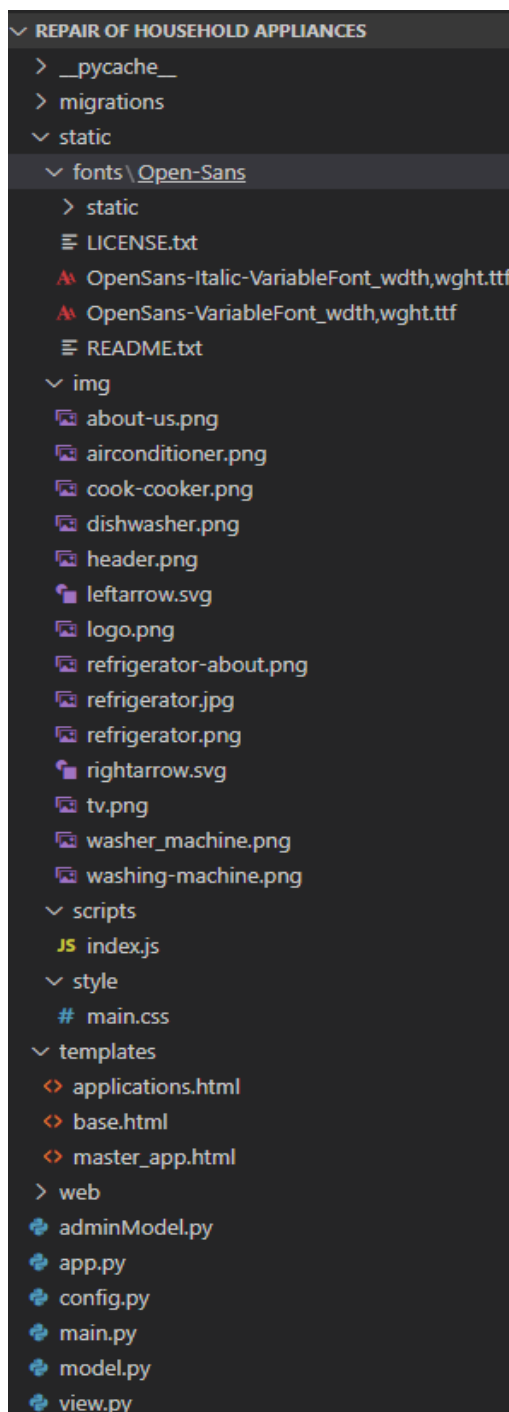


Рис. 2.1. Структура проекта

По данной структуре видно, что проект состоит из скриптов на языке программирования Python, на них написан backend проекта. Помимо это в проекте находятся HTML файлы, стили CSS, и клиентские скрипты JS.

Рассмотри каждый из серверных скриптов:

- `app.py` – основной файл проекта в нём происходит инициализация всех основных переменных.
- `config.py` – хранит настройки сервера
- `main.py` – служит для запуска проекта
- `model.py` – здесь лежат классы выступающие в качестве таблиц для базы данных
- `view.py` – отвечает за маршрутизацию по страницам проекта, здесь же хранятся все функции, например: регистрация, авторизация, выгрузка данных в базу данных и т. д.
- `adminModel.py` – содержит единственный класс позволяющий менять внешний вид административной панели.

В листинге 2.1 файла `view.py` вызывается функция `index()`. Данная функция предназначена в первую очередь для отображения файла “`base.html`” – главной страницы проекта. Также внутри этой функции реализованы методы для авторизации и регистрации пользователей.

Листинг 2.1

Маршрутизация

```
@app.route('/', methods=["POST", "GET"])
def index():
    service = Services.query.all()
    service_name_list = []
    service_img_list = []
    for i in service:
        service_name_list.append(i.name)
        service_img_list.append(i.img_name)
    if request.method == 'POST':
        if request.form['check_login'] == 'registration':
            name = request.form['name']
            email = request.form['email']
```

```

        address = request.form['address']
        telephone = request.form['telephone']
        pssw = generate_password_hash(request.form['pssw'])
        if name == "" or email == "" or address == "" or telephone ==
"" or pssw == "":
            flash("Вы заполнили не все поля")
        else:
            user =
Users.query.filter_by(email=request.form['email']).first()
            if user is None:
                user = Users(name = name, email = email, address =
address, telephone = telephone, passw = pssw, type_user=2)
                db.session.add(user)
                db.session.commit()
                return redirect('/#popup')
            else:
                flash("Пользователь с таким email уже
зарегистрирован")
        if request.form['check_login'] == 'login':
            email = request.form['login_email']
            pssw = request.form['login_pssw']
            if email and pssw:
                user = Users.query.filter_by(email=email).first()
                if check_password_hash(user.passw, pssw):
                    rm = True if request.form.get('remember') else False
                    login_user(user, remember = rm)
                    return redirect('/#nav')
            return render_template('base.html', current_user=current_user,
menu_list = service_name_list, img_list = service_img_list)

```

В листинге 2.2 реализован метод очистки сессии авторизации пользователя.

Листинг 2.2.

Функция очистки сессии и возврат на главную страницу.

```

@app.route("/logout")
@login_required
def logout():
    logout_user()

```

```
return redirect(url_for('index'))
```

В листинге 2.3 файла master_app.html реализована выборка всех заявок для каждого мастера.

Листинг 2.3.

Отображение всех заявок для работников.

```
{% extends 'base.html' %}
{% block content %}

<div class="application">
  <table class="app_table">
    {% for i in arr_master_app %}
      <tr>
        <th class="item_table">{{ i.user_name }}</th>
        <th class="item_table">{{ i.user_adress }}</th>
        <th class="item_table">{{ i.user_telephone }}</th>
      </tr>
    {% endfor %}
  </table>
</div>
{% endblock %}
```

База данных разрабатываемого программного продукта реализована в параграфе 2.2.

На следующем этапе реализуется страница отправки заявок пользователями. В листинге 2.4 показан код отображающий форму заявки, а также сам функционал отправки заявок.

Листинг 2.4.

Файл view.py метод отправки заявок.

```
@app.route('/applications', methods=["POST", "GET"])
@login_required
def applications():

    employees = Employees.query.all()

    service = Services.query.all()

    if request.method == 'POST':
        user_name = current_user.name
```

```

user_telephone = current_user.telephone
user_address = current_user.address
vehicle_type = str(request.form.get('service_list'))
master_name = str(request.form.get('master_list'))
id_vehicle_type = Services.query.filter_by(name=vehicle_type).first()

vehicle_model = request.form['model']
problem_description = request.form['problem']

master_app = Employees.query.filter_by(name=master_name).first()

application = Application(user_name = user_name, user_telephone =
user_telephone, user_adress = user_address,
        vehicle_model = vehicle_model, problem_description =
problem_description, master_id = master_app.id,
vehicle_type=int(id_vehicle_type.id))

db.session.add(application)
db.session.commit()

return redirect('/')

return render_template('applications.html', service_list = service, employees
= employees)

```

2.2. База данных программного продукта

База данных разработанного веб-приложения была создана локально в СУБД MySQL с использованием графического веб-интерфейса phpMyAdmin (рис. 2.2 «Создать БД»).

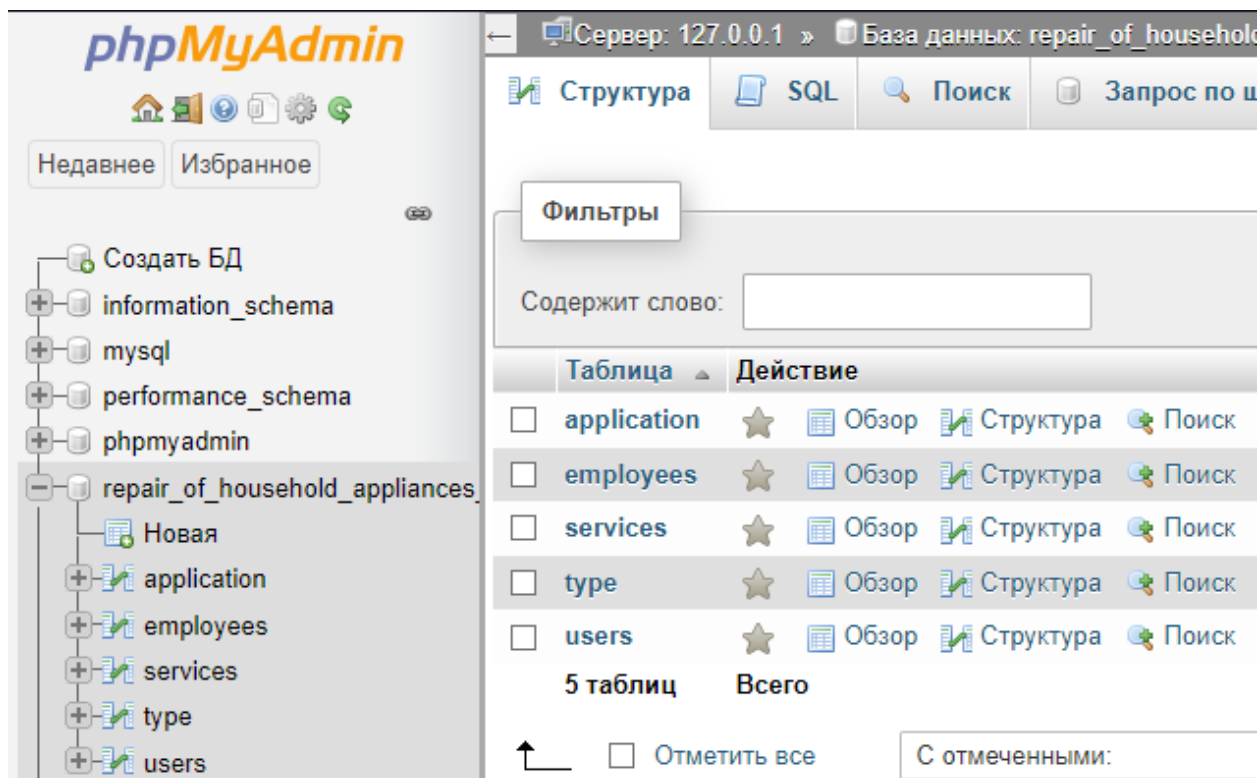


Рис. 2.2 База данных проекта

Далее необходимо задать наименование создаваемой базы данных, выбрать из предложенного списка кодировку и нажать кнопку «Создать».

В разработанном проекте в файле app.py реализована инициализация базы данных (рис. 2.3).

```
from flask import Flask
from config import Configuration
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager, current_user
from flask_admin import Admin, AdminIndexView

app = Flask(__name__)
app.config.from_object(Configuration)

db = SQLAlchemy(app)
login_manager = LoginManager(app)
admin = Admin(app)
```

Рис. 2.3. Файл app.py

Для полноценной реализации проекта необходимо было создать следующие таблицы в базе данных (рис 2.4:

- application
- employees

- services
- type
- users

Фильтры								
Содержит слово:								
Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано		
<input type="checkbox"/> application	★ Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8mb4_general_ci	48.0 Киб	-		
<input type="checkbox"/> employees	★ Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8mb4_general_ci	16.0 Киб	-		
<input type="checkbox"/> services	★ Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8mb4_general_ci	32.0 Киб	-		
<input type="checkbox"/> type	★ Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8mb4_general_ci	32.0 Киб	-		
<input type="checkbox"/> users	★ Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8mb4_general_ci	48.0 Киб	-		
5 таблиц	Всего	16	InnoDB	utf8mb4_general_ci	176.0 Киб	0 Байт		

Рис. 2.4. Таблицы базы данных.

На рисунке 2.5 представлена таблица, хранящая в себе данные о пользователях.

id	name	email	address	telephone	passwd	type_user
2	Дмитрий	dmalygin228@gmail.com	Курчатова 20	+79788354388	pbkdf2:sha256:150000\$Hb0IQGN4\$58634e5d791b2fb15049...	1
3	Дмитрий	malygin.d.i.2.18@gmail.com	Курчатова 20	+79788354389	pbkdf2:sha256:150000\$IdEX26Z\$b53f77795dba4c7da0e3...	3
4	Иван	fotahet883@healteas.com	Курчатова 20	+79788354385	pbkdf2:sha256:150000\$mcSKv6z0\$516cb736820ad2600557...	3

Рис. 2.5. Таблица users.

На рисунке 2.6 представлена таблица, хранящая в себе данные об отправленных заявках.

id	user_name	user_telephone	user_address	vehicle_model	problem_description	master_id	vehicle_type
6	Дмитрий	+79788354388	Курчатова 20	Lg	dsfadsfdas	1	1
7	Иван	+79788354385	Курчатова 20	Lg	asdfsdf	1	1

Рис. 2.6. Таблица application.

На рисунке 2.7 представлена таблица, хранящая в себе список сотрудников.

id	name	telephone
1	Дмитрий	+79788354388
2	Иван	+79788354389

Рис. 2.7. Таблица employees

На рисунке 2.8 представлена таблица, хранящая в себе данные об услугах.

id	name	img_name	ymployee_id
1	Холодильники	refrigerator.png	1
2	Стиральные машины	washing-machine.png	2
3	Посудомоечные машины	dishwasher.png	NULL
4	Телевизоры	tv.png	NULL
5	Кондиционеры	airconditioner.png	NULL
6	Духовые шкафы	cook-cooker.png	NULL

Рис. 2.8. Таблица application

На рисунке 2.9 представлена таблица, хранящая в себе данные о типах пользователей.

id	title
1	Админ
2	Клиент
3	Сотрудник

Рис. 2.9. Таблица type

Для реализации всех таблиц базы данных был создан отдельный скрипт model.py он необходим не только для реализации самих таблиц. С помощью данного файла осуществлялась вся работа с бд: от создания таблиц до запросов в саму базу данных (листинг 2.5).

Листинг 2.5.

Файл model.py.

```
from app import db
from flask_login import UserMixin

class Users(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(200), nullable=False)
    email = db.Column(db.String(120), nullable=False, unique=True)
    address = db.Column(db.String(200), nullable=False)
    telephone = db.Column(db.String(200), nullable=False)
    passw = db.Column(db.String(250), nullable=False)
    type_user = db.Column(db.Integer, db.ForeignKey('type.id'))

    def __init__(self, *args, **kwargs):
        super(Users, self).__init__(*args, **kwargs)
```



```

def __repr__(self):
    return '<Users {}>'.format(self.name)

class Type(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(200), nullable=False, unique=True)
    users_id = db.relationship("Users", backref='type', uselist=False)

    def __init__(self, *args, **kwargs):
        super(Type, self).__init__(*args, **kwargs)

    def __repr__(self):
        return '<Type {}>'.format(self.title)

class Services(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(200), nullable=False)
    img_name = db.Column(db.String(200), nullable=False)
    ymployee_id = db.Column(db.Integer, db.ForeignKey('employees.id'))
    application_id = db.relationship("Application", backref='type',
uselist=False)

    def __init__(self, *args, **kwargs):
        super(Services, self).__init__(*args, **kwargs)

    def __repr__(self):
        return '<Services {}>'.format(self.name)

class Employees(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(200), nullable=False)
    telephone = db.Column(db.String(200), nullable=False)

    services = db.relationship("Services")
    application = db.relationship("Application")

    def __init__(self, *args, **kwargs):
        super(Employees, self).__init__(*args, **kwargs)

```

```

def __repr__(self):
    return '<Employees {}>'.format(self.name)

class Application(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_name = db.Column(db.String(200), nullable=False)
    user_telephone = db.Column(db.String(200), nullable=False)
    user_adress = db.Column(db.String(200), nullable=False)
    vehicle_model = db.Column(db.String(200), nullable=False)
    problem_description = db.Column(db.String(200), nullable=False)
    master_id = db.Column(db.Integer, db.ForeignKey('employees.id'))
    vehicle_type = db.Column(db.Integer, db.ForeignKey('services.id'))

    def __init__(self, *args, **kwargs):
        super(Application, self).__init__(*args, **kwargs)

    def __repr__(self):
        return '<Application {}>'.format(self.vehicle_type)

```

2.3. Описание интерфейса приложения

В ходе разработки основной части веб-приложения был спроектирован веб-сервис, обладающий базой данных для хранения информации о пользователях, сотрудниках и заявках.

При посещении сервиса пользователя встречает главная страница, где он может ознакомиться с сферой деятельности сервиса (рис 2.10).

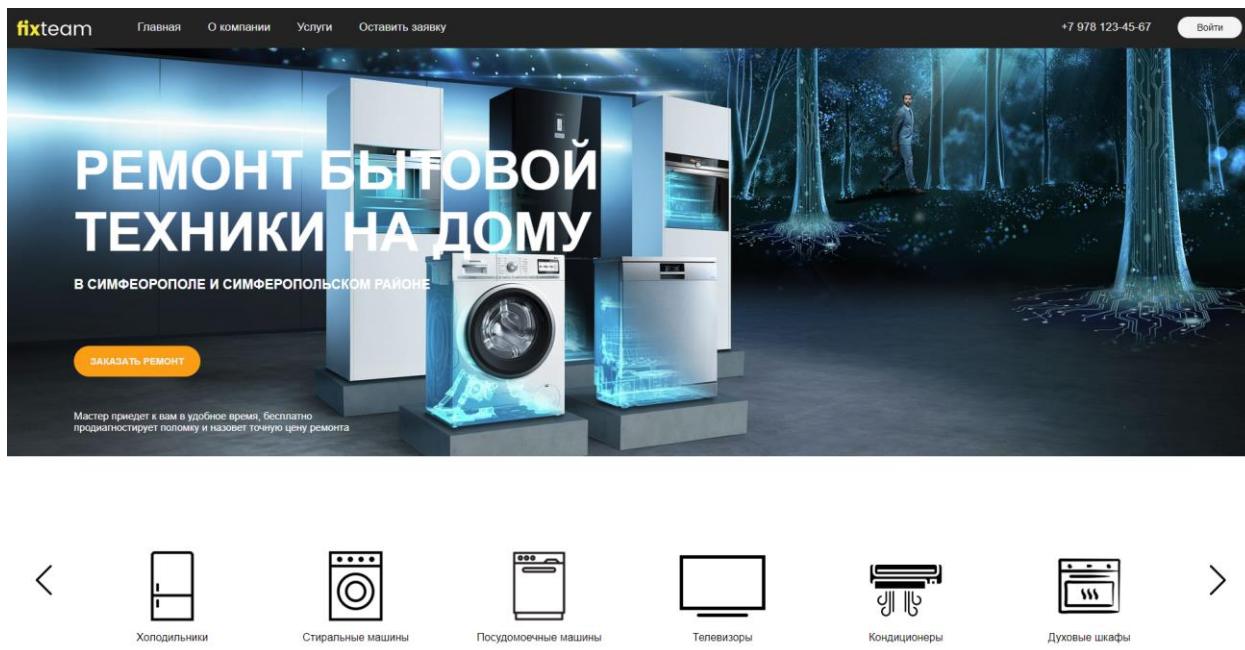


Рис. 2.10. Главная страница проекта.

Далее при желании пользователь может оставить заявку на ремонт своей техники, для этого ему нужно пройти регистрацию (рис 2.11) и авторизацию и (рис 2.12).

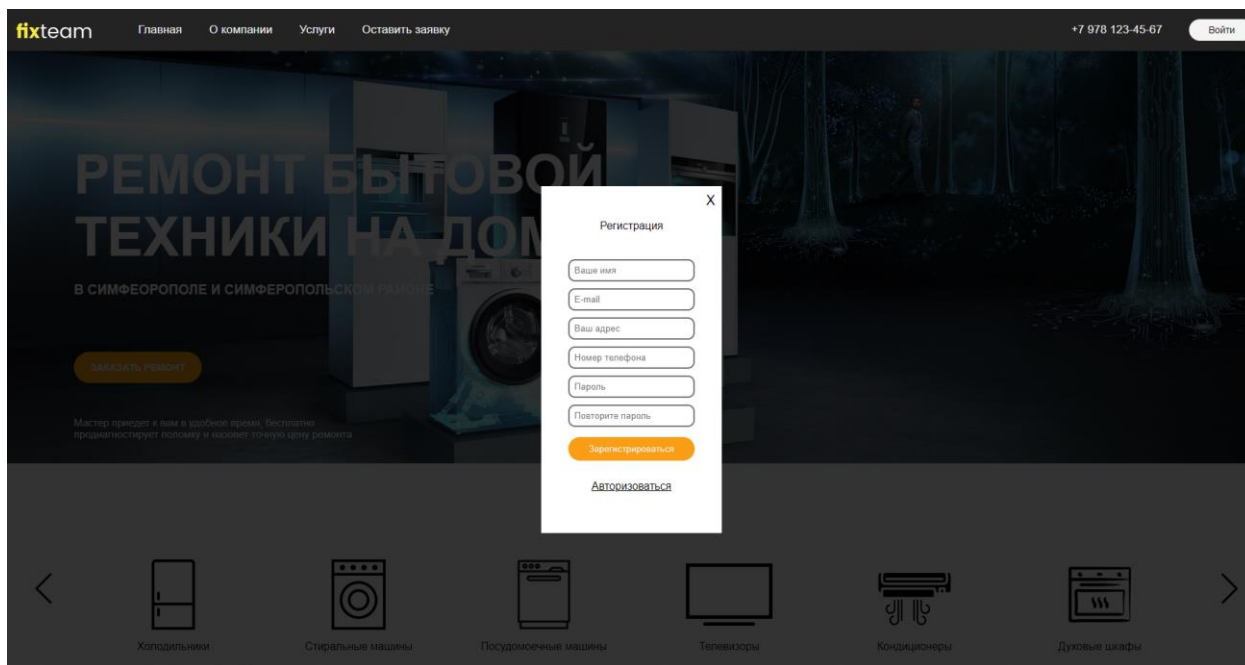


Рис. 2.11 Форма регистрации пользователей.

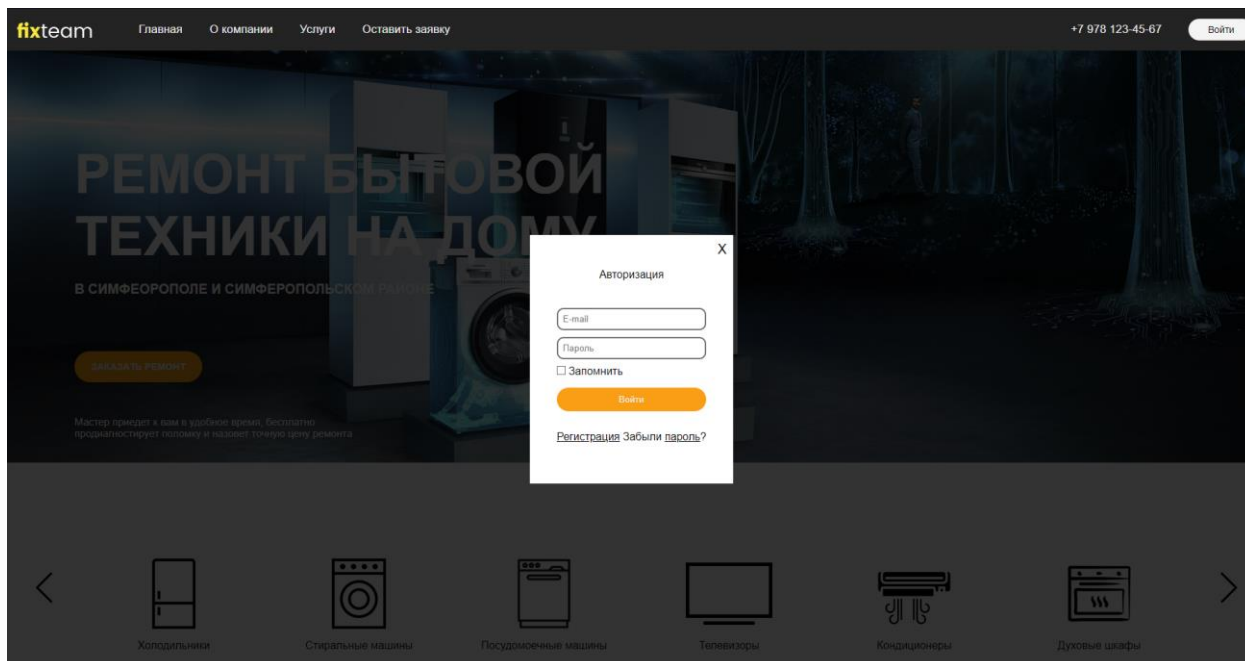


Рис. 2.12. Форма авторизации пользователей.

На странице «Оставить заявку» пользователю необходимо заполнить все поля из формы (рис. 2.13), после чего сотрудник свяжется с клиентом при необходимости уточнения вопросов.

Рис. 2.13. Форма для заявки на ремонт техники

Для сотрудников сервиса создана отдельная страница, где отображаются их заявки, с необходимой информацией (рис. 2.14).

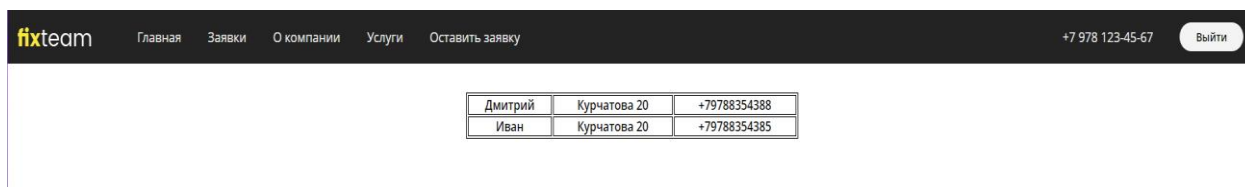


Рис. 2.14. Страница с заявками.

Помимо всего этого существует страница администрирования проекта, доступ к ней имеют только пользователи со статусом «администратор». На этой странице отображается вся информация о клиентах, сотрудниках, услугах, заявках (рис. 2.15).

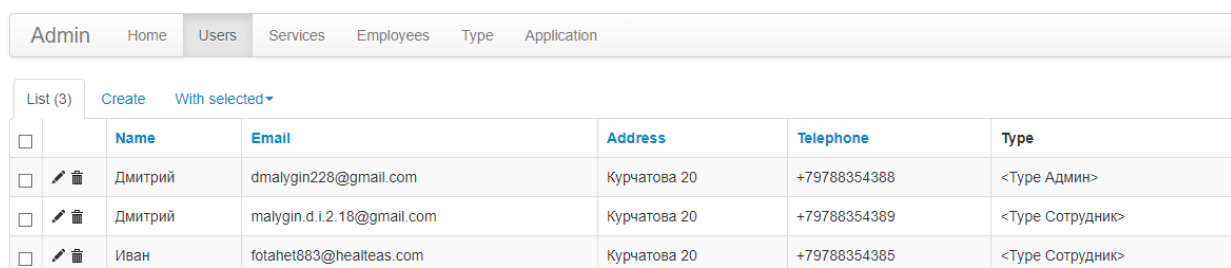


Рис. 2.15. Административная панель сервиса.

Таким образом в данной подглаве была рассмотрена структура веб-сервиса, описаны основные страницы: главная страница, авторизация, регистрация, отправка заявок, страница сотрудника, административная панель.

Вывод к главе 2

Во второй главе был продемонстрирован процесс разработки веб-сервиса по ремонту бытовой техники. Проект состоит из: главной страницы и страниц, представляющей собой административную панель. На главной странице отображается информация о предоставляемых услугах.

Помимо этого, в данной главе были приведены практические аспекты разработки базы данных, и рассмотрены формирующие её объекты, а также описаны основные этапы в разработке приложения.

ЗАКЛЮЧЕНИЕ

Курсовой проект посвящён разработке веб-сервиса для сервиса по ремонту бытовой техники. Разработанный сервис обладает широким функционалом, встроенные инструменты позволяют администратору узнать информацию об определенном сотруднике. Администратор может добавлять новых сотрудников, редактировать заявки, добавлять новые услуги. Кроме того, сервис обладает понятным интерфейсом для обычного пользователя.

В результате проделанной работы необходимо сделать следующие выводы:

- проанализированы требования к разрабатываемому программному приложению на основе обзора существующих сервисов подобного характера;
- выполнены проектирование приложения и выбор программных инструментариев;
- разработан веб-сервис для системы управления персоналом.

В ходе разработки базы, были задействованы все основные средства управления и отображения информации. В дальнейшем планируются улучшение и доработки разработанного приложения. Все поставленные цели были выполнены в полном объеме.

В ходе разработки курсового проекта на тему «Сервис по ремонту бытовой техники» были созданы 3 экрана, а именно: «Главная страница», «Отправка заявок», «Административная панель». Для данного сервиса создана база данных с помощью веб-интерфейса для администрирования «phpMyAdmin». Созданная база данных работает корректно и автоматизирует следующие ключевые моменты работы: отправка заявок, хранение данных о пользователях, сотрудниках, заявках и услугах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. iimag.ru [Электронный ресурс] // iimag.ru – 2020. – Режим доступа: <https://iimag.ru/opredelenie-veb-sajta-vidy-i-klassifikacii-veb-sajtov/>
2. yaklass.ru [Электронный ресурс] // yaklass.ru – 2017. – Режим доступа: <https://www.yaklass.ru/p/informatika/11-klass/sozдание-web-stranits-187010/etapy-sozdaniia-saita-187019/re-b5347e1e-a2db-4bb1-b3a1-fe813d107c43>
3. Шабанов С.В. Применение метаданных в адаптивных информационных системах клиент-серверной архитектуры – / С.В. Шабанов // Международный симпозиум «Надежность и качество». – 2010
4. site-protect.ru [Электронный ресурс] // site-protect.ru– 2020. – Режим доступа: <https://www.site-protect.ru/o-nas/vopros-otvet/sozдание-saitov/poriadok-sroki-razrabotki-saita>
5. wezom.com.ua [Электронный ресурс] // wezom.com.ua – 2011. – Режим доступа: <https://wezom.com.ua/blog/etapy-razrabotki-sajta>
6. Люк Веллинг, Лора Томсон. PHP and MySQL Web Development – / Люк Веллинг, Лора Томсон // – 2010
7. Хоган Брайан. Книга веб-программиста: секреты профессиональной разработки веб-сайтов – / Хоган Брайан // – 2013
8. Марейн Хавербек. Выразительный Javascript – / Марейн Хавербек // – 2013
9. intuit.ru [Электронный ресурс] // iimag.ru – 2017. – Режим доступа: <https://intuit.ru/studies/courses/1001/297/lecture/7409>
10. Дакетт Джон. HTML и CSS: Design and Build Websites – / Дакетт Джон // – 2013
11. lucidchart.com [Электронный ресурс] // lucidchart.com – 2021. – Режим доступа: <https://www.lucidchart.com/pages/ru/проектирование-базы-данных>
12. mstu.edu.ru [Электронный ресурс] // mstu.edu.ru – 2019. – Режим доступа: http://www.mstu.edu.ru/study/materials/zelenkov/ch_5_1.html