Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

КАФЕДРА БИЗНЕС-ИНФОРМАТИКИ И МОДЕЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ

**КУРСОВАЯ РАБОТА**

Создание Web-приложения с применением современных Web-технологий по тематике «Консалтинг».

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Никитин

подпись, дата

Студент УБ23-08Б, 432323950 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Симахин подпись, дата

Красноярск 2024

**Содержание**

[**Введение** 2](#_Toc187184201)

[**1 Теоретическая часть** 4](#_Toc187184202)

[1.1 HTML 4](#_Toc187184203)

[1.2 CSS 5](#_Toc187184204)

[1.3 JavaScript 6](#_Toc187184205)

[1.4 PHP 8](#_Toc187184206)

[1.5 MySQL 9](#_Toc187184207)

[**2 Практическая часть** 10](#_Toc187184208)

[2.1 Требования к выполнению работы 10](#_Toc187184209)

[2.2 Создание формы регистрации и личного кабинета 12](#_Toc187184210)

[2.3 Главная страница 16](#_Toc187184212)

[2.4 Функции администратора 21](#_Toc187184216)

[2.5 Стиль сайта 26](#_Toc187184217)

[**Заключение** 31](#_Toc187184218)

[**Список использованных источников** 34](#_Toc187184230)

# **Введение**

Современные веб-технологии играют ключевую роль в развитии бизнеса, позволяя автоматизировать процессы, повышать удобство взаимодействия с клиентами и предоставлять доступ к услугам в режиме онлайн. Веб-приложения становятся важным инструментом для организаций, которые стремятся улучшить взаимодействие с клиентами и предложить качественный сервис. В данном проекте рассматривается разработка веб-приложения, предназначенного для управления консалтинговыми услугами, с использованием современных технологий.

Целью данной курсовой работы является разработка веб-приложения, обеспечивающего удобную и функциональную платформу для управления консалтинговыми услугами. Приложение должно включать регистрацию и авторизацию пользователей, разграничение ролей (пользователи и администраторы), управление услугами и данными пользователей, а также адаптивный дизайн для работы на различных устройствах.

Для достижения поставленной цели были сформулированы следующие задачи:

* 1. Изучить современные веб-технологии, такие как HTML, CSS, JavaScript, PHP и MySQL, для их применения в проекте.
  2. Разработать архитектуру приложения, включающую основные страницы и функциональные модули.
  3. Реализовать функционал авторизации и регистрации пользователей с разграничением прав доступа.
  4. Создать личный кабинет для пользователей, позволяющий редактировать данные профиля и просматривать услуги.
  5. Реализовать административный интерфейс для управления пользователями и услугами, включая их добавление, изменение и удаление.
  6. Разработать адаптивный дизайн приложения для обеспечения удобства использования на различных устройствах.
  7. Протестировать работу приложения на соответствие требованиям и исправить выявленные ошибки.

Объектом исследования является процесс разработки веб-приложения для управления консалтинговыми услугами, в частности, его функциональная и техническая реализация с использованием современных веб-технологий.

Методом исследования в рамках данной работы является проектирование и практическая реализация веб-приложения. Используется подход инкрементальной разработки, при котором каждый этап (проектирование, реализация, тестирование) проходит с последовательной детализацией и проверкой функциональности. Для анализа и проверки результатов применялись методы функционального тестирования, позволяющие убедиться в корректной работе системы в соответствии с требованиями.

Данная курсовая работа представляет собой практическое исследование процесса разработки веб-приложения, начиная с теоретического изучения технологий и заканчивая реализацией и тестированием. Результаты работы могут быть полезны для создания аналогичных приложений, а также послужить основой для дальнейшего изучения и совершенствования технологий веб-разработки.

# **1 Теоретическая часть**

## 1.1 HTML

HTML (HyperText Markup Language) — это стандартный язык разметки, используемый для создания структуры веб-страниц. С помощью HTML веб-разработчики задают, как будут выглядеть заголовки, абзацы, изображения, ссылки, таблицы и другие элементы на странице. HTML был разработан в конце 1980-х годов Тимом Бернерс-Ли, и с тех пор он стал основой для создания веб-документов в Интернете.

Современный HTML представлен версией HTML5, которая была официально опубликована в октябре 2014 года. Она значительно улучшила функциональность языка, добавив новые элементы и атрибуты для мультимедиа, форм и структурирования контента. HTML5 поддерживает аудио, видео, холсты для рисования, а также новые семантические элементы, которые делают структуру страницы более понятной для браузеров, поисковых систем и разработчиков.

HTML состоит из тегов, которые обрамляют контент. Теги бывают парными, например, Заголовок, где — это открывающий тег, а — закрывающий, и одиночными, например, , используемыми для вставки изображений. Атрибуты тегов, такие как id, class, href, src, alt, предоставляют дополнительную информацию о элементах. Например, атрибут alt описывает содержимое изображения для случаев, когда оно не может быть отображено.

Пример базовой структуры HTML-документа (Рис. 1):

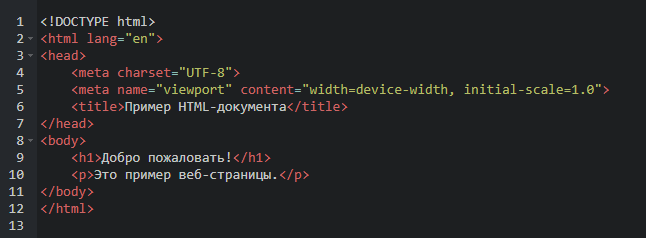


Рисунок 1 – Базовая структура HTML

Этот документ начинается с декларации, которая указывает, что используется HTML5. Затем идет раздел, содержащий метаинформацию о странице, и, где располагается основной контент.

## 1.2 CSS

CSS (Cascading Style Sheets, или каскадные таблицы стилей) — это язык описания внешнего вида HTML-документов. Он используется для управления стилем и компоновкой веб-страниц, включая цвет, шрифты, расположение элементов и адаптивность.

CSS был разработан для разделения структуры и стиля веб-страниц, что позволяет разработчикам легко изменять внешний вид сайта, не затрагивая его содержание. Он состоит из правил, которые определяют стиль элементов. Каждое правило состоит из селектора и блока деклараций. Например (Рис. 2):

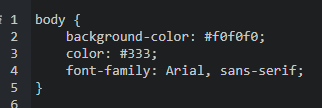


Рисунок 2 – Селектор body

В этом примере селектор body применяет стили ко всей странице. Фон задается с помощью background-color, текст окрашивается с помощью color, а шрифт определяется через font-family.

CSS предоставляет множество возможностей, включая:

* Блочная и строчная модель: определяет, как элементы располагаются на странице.
* Макет и выравнивание: с помощью flexbox и grid можно создавать сложные компоновки.
* Медиа-запросы: позволяют адаптировать дизайн под разные устройства и размеры экранов.
* Анимации: делают интерфейсы более интерактивными с помощью ключевых кадров и переходов.

Пример использования медиа-запроса (Рис. 3):

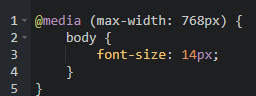


Рисунок 3 – Медиа-запрос

Этот код уменьшает размер шрифта на устройствах с шириной экрана меньше 768 пикселей, что делает сайт адаптивным.

## 1.3 JavaScript

JavaScript — это высокоуровневый язык программирования, используемый для создания интерактивности на веб-страницах. Он позволяет добавлять такие функции, как обработка событий (например, нажатия кнопок), валидация форм, анимации, взаимодействие с сервером без перезагрузки страницы (AJAX) и многое другое.

JavaScript впервые появился в 1995 году благодаря Брендану Эйху и стал неотъемлемой частью веб-разработки. С тех пор язык значительно эволюционировал, получив современные возможности, такие как классы, модули, промисы и async/await для асинхронного программирования.

Пример простого JavaScript-кода (Рис. 4):

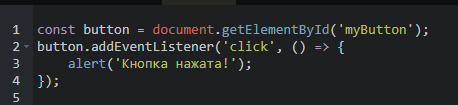


Рисунок 4 – Пример JavaScript-кода

Этот код добавляет обработчик события на кнопку с id "myButton", чтобы при нажатии на неё отображалось сообщение.

JavaScript тесно интегрируется с DOM (Document Object Model), что позволяет изменять содержимое страницы на лету. Пример (Рис. 5):



Рисунок 5 – DOM

Этот код изменяет текст элемента с id "title".

JavaScript также используется в связке с библиотеками и фреймворками, такими как jQuery, React, Angular и Vue.js, которые упрощают разработку и масштабирование приложений. Кроме того, с помощью Node.js язык стал популярным для серверной разработки.

## 1.4 PHP

PHP (Hypertext Preprocessor) — это язык серверного программирования, созданный для генерации динамических веб-страниц. Он позволяет взаимодействовать с базами данных, обрабатывать формы, управлять сессиями и выполнять другие серверные задачи.

PHP был разработан в 1994 году Расмусом Лердорфом и быстро стал одним из самых популярных языков для веб-разработки благодаря своей простоте и интеграции с HTML. Код PHP исполняется на сервере, а результат отправляется клиенту в виде HTML.

Пример PHP-кода (Рис. 6):



Рисунок 6 – Пример PHP-кода

Этот код подключается к базе данных, выполняет запрос и выводит результаты на экран.

PHP поддерживает множество функций, таких как работа с массивами, строками, файлами и регулярными выражениями. Для построения сложных приложений используются фреймворки, такие как Laravel, Symfony и CodeIgniter, которые предоставляют структуры для маршрутизации, работы с базами данных и обработки запросов.

## 1.5 MySQL

MySQL — это система управления реляционными базами данных (СУБД), которая используется для хранения и управления данными. Она является одной из самых популярных СУБД благодаря своей производительности, масштабируемости и открытой лицензии.

MySQL была разработана шведской компанией MySQL AB и впервые выпущена в 1995 году. Она поддерживает язык SQL (Structured Query Language) для взаимодействия с данными.

Пример создания таблицы в MySQL (Рис. 7):

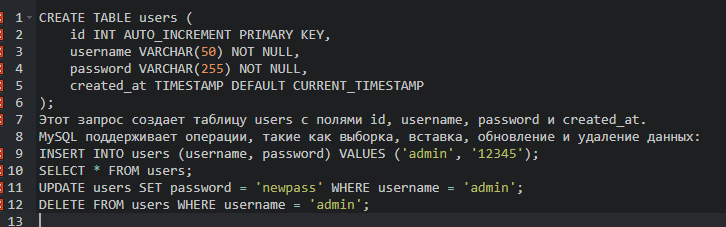


Рисунок 7 – Пример создания таблицы

Для взаимодействия с PHP используется расширение PDO или MySQLi. Например (Рис. 8):

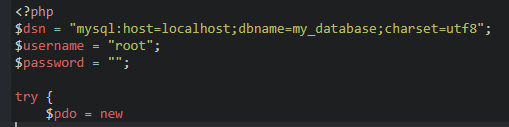


Рисунок 8 – Расширение PDO

# **2 Практическая часть**

## 2.1 Требования к выполнению работы

Перед тем, как приступить к разработке Web-приложения, необходимо разобраться с тем, что от нас требуется. К данной работе предъявлено 8 требований:

* + 1. Должна быть предусмотрена авторизация и регистрация пользователей.
    2. Зарегистрированный пользователь имеет право редактировать свои данные и просматривать свои услуги.
    3. Администраторы имеют право редактировать пользователей и добавлять услуги.
    4. Неавторизированные пользователи имеют право просматривать список услуг.
    5. Сайт должен быть выдержан в определенном стиле и корректно открываться на любом разрешении.
    6. Общий функционал может быть продуман самостоятельно.
    7. Важно продемонстрировать не большой объем функционала, а минимальные требования к проекту. Проект может максимально простым, но при этом содержать все необходимые функции.
    8. Отдельные бонусы предусмотрены за доп. функционал и красивый дизайн.

После ознакомления с требованиями к работе, необходимо настроить базовую конфигурацию для корректной работы приложения. Для этого воспользуемся сборкой локального веб-сервера XAMPP. После скачивания файла с официального сайта и его установки, все необходимые программы будут загружены. После установки мы открываем приложение XAMPP и запускаем модули “Apache” и “MySQL”. Далее необходимо открыть локальную папку с файлами и работать с программой по дистрибутиву xampp/htdocs.

Далее необходимо создать базу данных на платформе phpmyadmin. Для этого создадим файл database(db).sql, в которой кратко опишем, что будет находится в базе данных (Рис. 9).

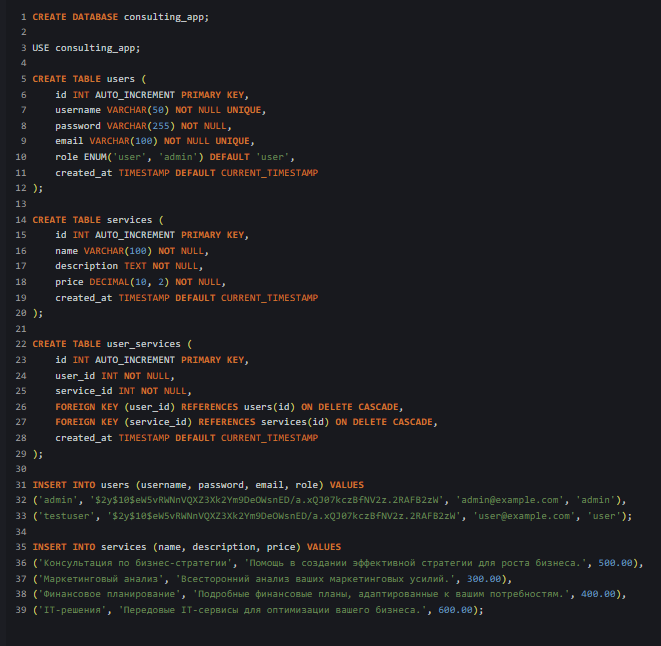


Рисунок 9 – Создание базы данных.

После импортируем данную базу данных в PhpMyAdmin. На рисунке 10 видно 3 таблицы: **users**, **user\_services** и **services**.

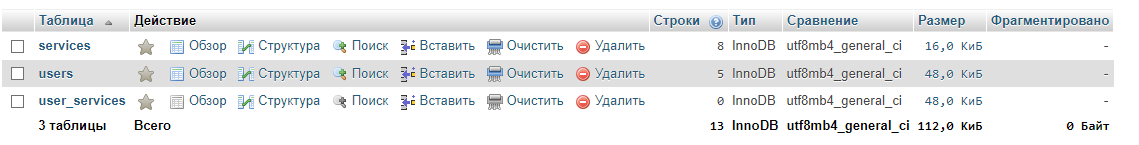


Рисунок 10 – Таблицы в базе данных PhpMyAdmin.

Все требования проекта были изучены и была создана база данных. Теперь можно приступить к созданию сайта.

## 2.2 Создание формы регистрации и личного кабинета

Создание формы регистрации и личного кабинета — это одна из ключевых частей разработки веб-приложения. Она обеспечивает возможность пользователям зарегистрироваться в системе, хранить персональные данные и взаимодействовать с приложением через их личный кабинет. В этой части подробно опишем процесс реализации, начиная от регистрации нового пользователя до создания функционального личного кабинета.

1. **Создание HTML-формы для ввода данных**

В HTML-коде форма регистрации содержит такие поля, как имя пользователя, пароль и их подтверждение. Пример из проекта (Рис. 11):

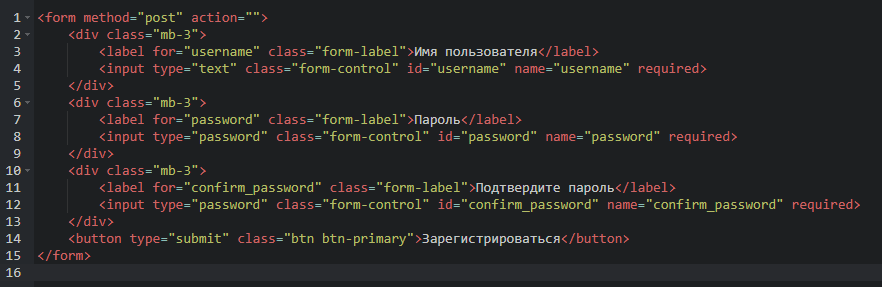


Рисунок 11 – Создание HTML-формы

1. **Проверка данных на сервере**

После отправки формы данные передаются на сервер, где они проверяются и обрабатываются. Пример серверной обработки из кода (Рис. 12):

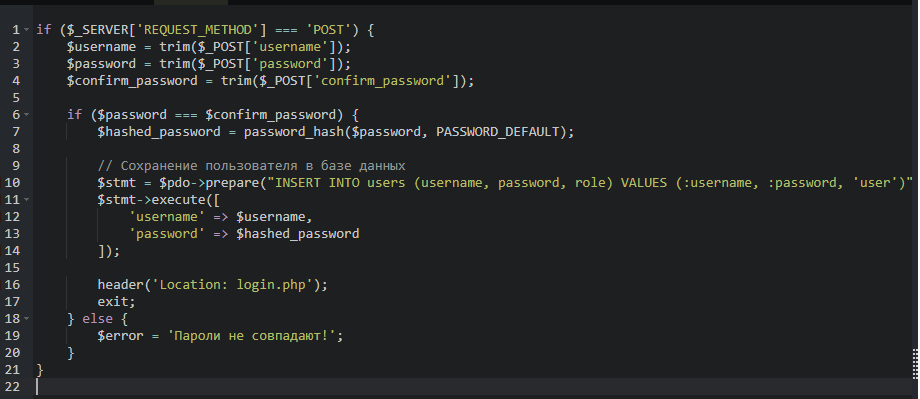


Рисунок 12 – Серверная обработка

В этом коде осуществляется проверка на совпадение паролей, а также используется password\_hash() для безопасного хранения паролей.

1. **Сохранение данных в базе данных**

Регистрационные данные сохраняются в таблице users. Таблица имеет следующие поля:

* id — уникальный идентификатор пользователя.
* username — имя пользователя.
* password — хэш пароля.
* role — роль пользователя (например, user или admin).

После регистрации пользователь может войти в систему и получить доступ к личному кабинету, где хранятся его персональные данные. Реализация личного кабинета включает следующие этапы:

1. **Проверка авторизации пользователя**

При попытке доступа к личному кабинету система проверяет, авторизован ли пользователь. Если нет, его перенаправляют на страницу входа. Пример проверки авторизации (Рис. 13):

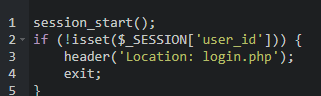


Рисунок 13 – Проверка авторизации

1. **Отображение персональных данных**

После авторизации система извлекает данные пользователя из базы и отображает их в личном кабинете. Пример кода для вывода информации (Рис. 14):

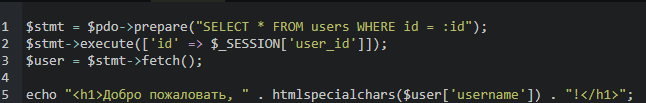


Рисунок 14 – Вывод информации

1. **Редактирование данных**

Пользователь может изменять свои данные, такие как имя пользователя или пароль. Для этого используется форма редактирования (Рис. 15):

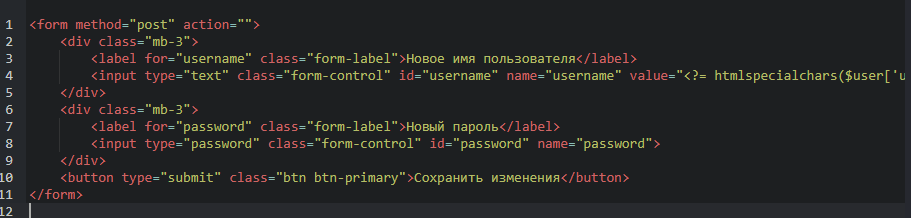


Рисунок 15 – Форма редактирования

При отправке формы изменения сохраняются в базе данных (Рис. 16):

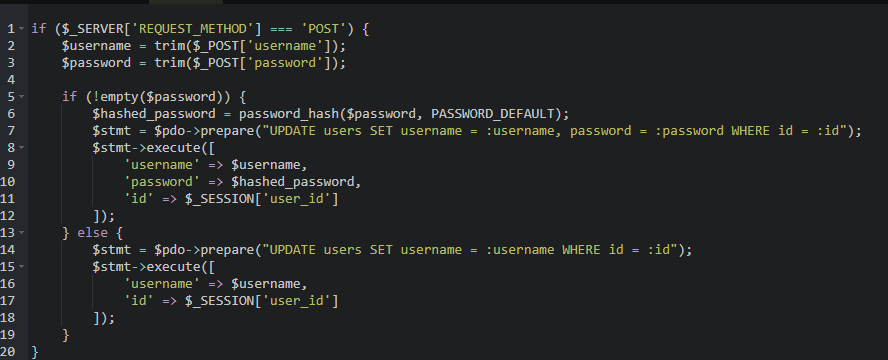


Рисунок 16 – Сохранение изменений в базе данных

Для удобства пользователя личный кабинет стилизован с использованием Bootstrap. Пример оформления (Рис. 17):

### 

Рисунок 17 – Оформление личного кабинета

Форма регистрации и личный кабинет (Рис. 18) и (Рис. 19) реализованы с использованием современных технологий PHP, MySQL и Bootstrap. Они обеспечивают безопасность данных, удобство для пользователей и адаптивный интерфейс. Данный функционал — это фундаментальная часть приложения, которая позволяет пользователю взаимодействовать с системой.

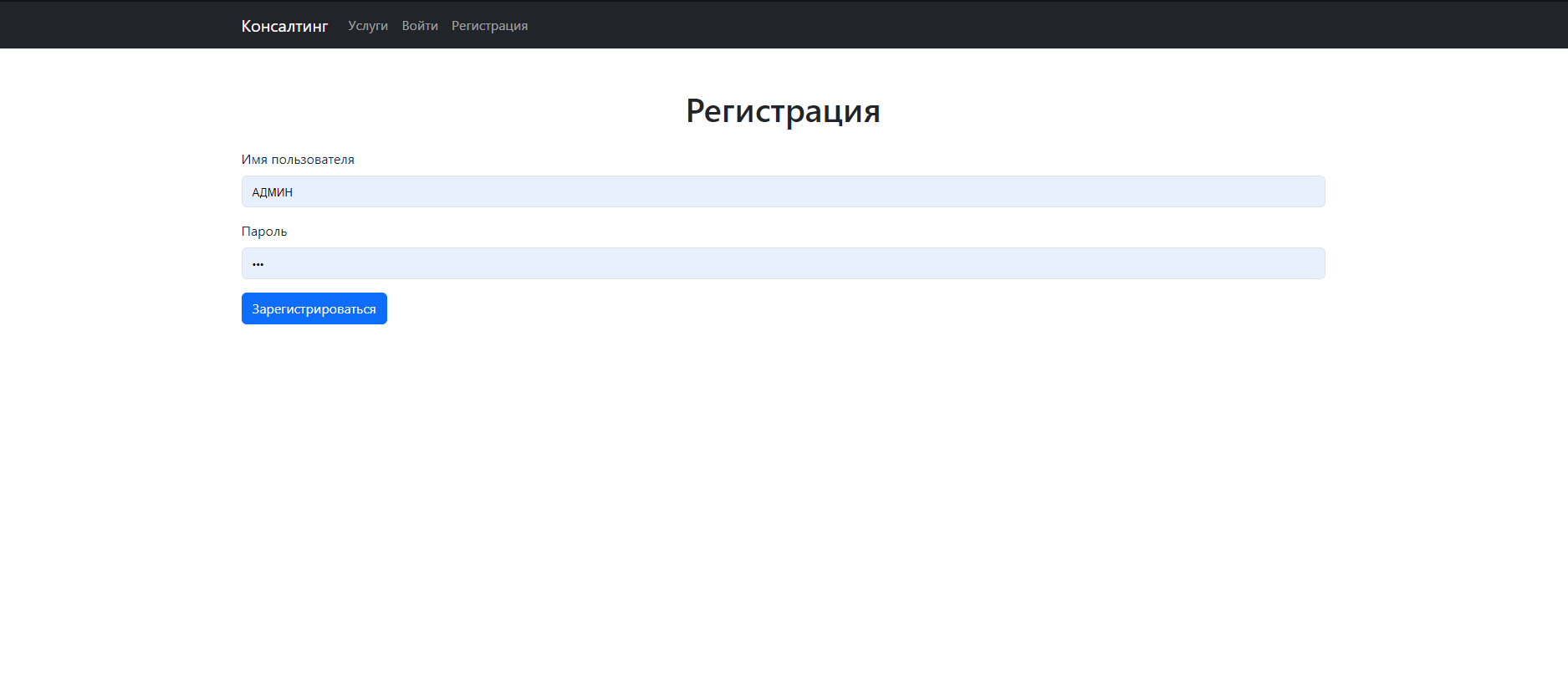


Рисунок 18 – Окно регистрации

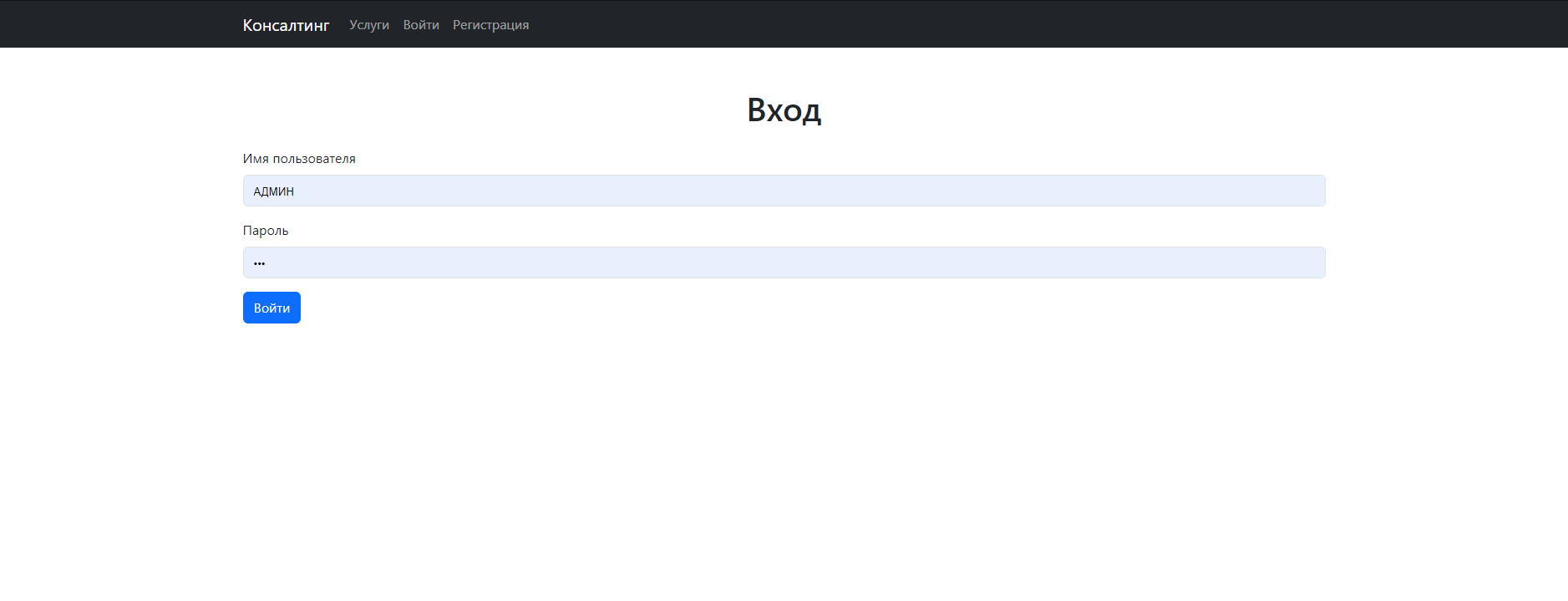


Рисунок 19 – Окно авторизации

## 2.3 Главная страница

Главная страница является отправной точкой веб-приложения. Это лицо сайта, от которого зависит первое впечатление пользователя. Главная страница нашего консалтингового приложения реализована с акцентом на удобство, простоту и информативность, с использованием адаптивного дизайна, современного HTML и CSS. В этом разделе подробно описывается, как была создана и структурирована главная страница, а также как был реализован функционал для разных категорий пользователей.

Главная страница приложения построена с использованием языка HTML5 и имеет следующую логическую структуру:

1. **Шапка (Header)**

Включает навигационное меню, которое позволяет пользователю быстро перемещаться между страницами сайта (например, "Услуги", "Вход", "Регистрация" или "Личный кабинет").

1. **Основной блок (Main Section)**

Представляет собой центральную часть страницы, где отображаются доступные услуги.

1. **Подвал (Footer)**

Содержит контактную информацию, копирайты и ссылки на политику конфиденциальности или условия использования.

Пример HTML-кода главной страницы (Рис. 20):

### 

Рисунок 20 – Код главной страницы

### Навигационная панель (шапка) (Рис. 22 и Рис. 23) обеспечивает удобный переход между различными разделами сайта. Она адаптивна благодаря использованию Bootstrap. Вот пример кода навигации (Рис. 21):

### 

Рисунок 21 – Код навигации

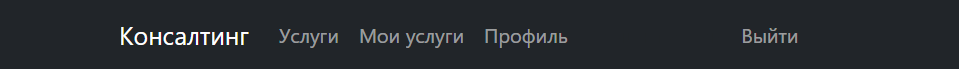


Рисунок 22 – Шапка для обычного пользователя

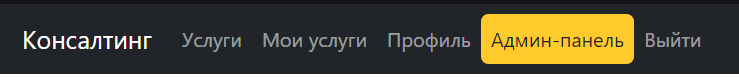


Рисунок 23 – Шапка для администратора

Особенности:

* Ссылки на основные страницы автоматически отображаются в зависимости от авторизации пользователя.
* У администраторов добавлена ссылка на админ-панель.

Для визуального оформления и адаптации под разные устройства используются CSS и Bootstrap. Например:

* Сетка Bootstrap позволяет распределить услуги по колонкам, адаптируя их к ширине экрана.
* Карточки Bootstrap делают вывод услуг эстетичным и удобным.

Пример CSS для улучшения внешнего вида (Рис. 24):

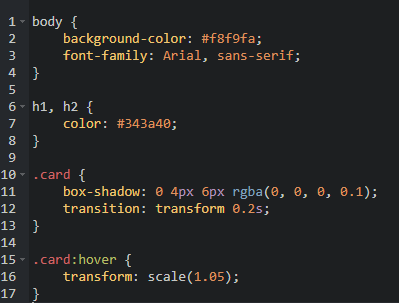


Рисунок 24 – Код CSS

Главная страница предоставляет следующий функционал:

1. **Просмотр списка услуг**

Все услуги, добавленные в базу данных, выводятся в виде карточек с названием, описанием и ценой.

1. **Динамическое содержание**

Если услуг в базе данных нет, отображается сообщение: *"На данный момент услуги отсутствуют."*

1. **Разграничение прав доступа**

* Неавторизованные пользователи могут просматривать услуги, но не заказывать их.
* Авторизованные пользователи имеют доступ к своим услугам и личному кабинету.
* Администраторы могут перейти в панель управления для редактирования услуг.

Данные об услугах берутся из таблицы services, которая имеет следующую структуру:

* id — уникальный идентификатор услуги.
* name — название услуги.
* description — описание.
* price — стоимость.

Пример SQL-запроса для выборки данных (Рис. 25):



Рисунок 25 – SQL-запрос для выборки данных

Этот запрос извлекает все услуги из базы данных и передает их для отображения.

«Подвал» содержит общую информацию и служит для завершения страницы. Пример кода (Рис. 26):

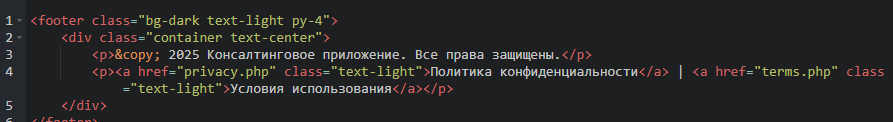


Рисунок 26 – Код «подвала»

## 2.4 Функции администратора

Администратор в консалтинговом веб-приложении играет ключевую роль в управлении платформой. Его функции включают добавление, редактирование и удаление услуг, управление пользователями, а также возможность настройки системы. Раздел описывает, как были реализованы функции администратора, включая логику, интерфейс и взаимодействие с базой данных.

Админ-панель предоставляет администратору удобный интерфейс для выполнения ключевых задач. Доступ к админ-панели имеют только пользователи с ролью "администратор".

Ключевые возможности админ-панели:

* + 1. Управление списком услуг:
    2. Добавление новых услуг.
    3. Редактирование существующих услуг.
    4. Удаление услуг.
    5. Управление пользователями:
    6. Просмотр списка зарегистрированных пользователей.
    7. Редактирование данных пользователей.
    8. Удаление пользователей.
    9. Защита панели от неавторизованного доступа.

Пример кода проверки роли администратора (Рис. 27):

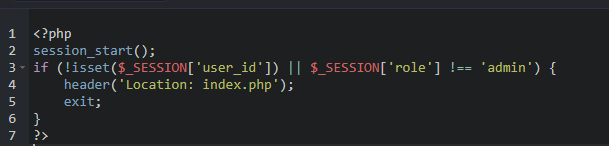


Рис. 27 – Проверка роли

Этот код помещается в начале файлов админ-панели для проверки прав доступа.

Управление услугами является одной из основных задач администратора. На странице "Управление услугами" администратор может выполнять следующие действия:

* 1. **Отображение всех услуг**  
     На странице выводится таблица с данными об услугах, включая название, описание, цену и кнопки для редактирования и удаления.

Пример кода для отображения услуг (Рис. 28):

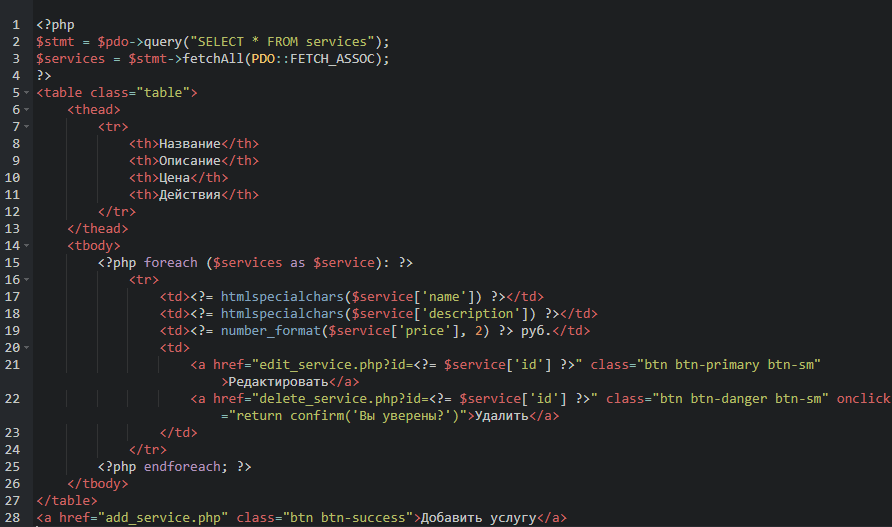


Рисунок 28 – Код для отображения услуг

1. **Добавление услуги**

Администратор переходит на форму добавления услуги, где он заполняет поля "Название", "Описание" и "Цена". Данные отправляются в базу данных с помощью SQL-запроса INSERT.

Пример кода добавления услуги (Рис. 29):



Рисунок 29 – Код добавления услуги

1. **Редактирование услуги**

Редактирование услуги предполагает получение данных текущей услуги из базы данных, отображение их в форме и обновление информации с помощью SQL-запроса UPDATE.

Пример кода для редактирования услуги (Рис. 30):

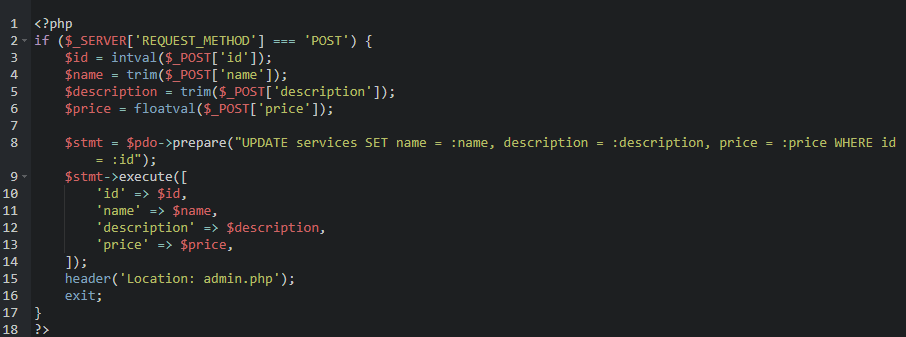


Рисунок 30 – Код для редактирования услуги

1. **Удаление услуги**

Для удаления услуги используется SQL-запрос DELETE.

Пример кода для удаления услуги:

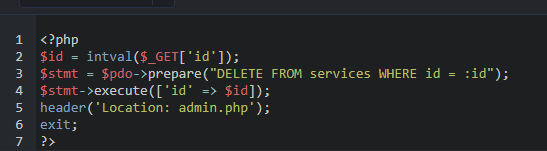


Рисунок 31 – Код для удаления услуги

В админ-панели администратор может просматривать список зарегистрированных пользователей, а также редактировать и удалять их учетные записи.

1. Просмотр списка пользователей  
   Список пользователей отображается в виде таблицы, где указаны их ID, имена пользователей, роли и действия.
2. Редактирование пользователя  
   Редактирование включает изменение имени пользователя и его роли (например, "пользователь" или "администратор").
3. Удаление пользователя  
   Удаление пользователя выполняется через SQL-запрос DELETE.

Админ-панель (Рис. 32) оформлена в стиле Bootstrap для удобства использования. Также были добавлены следующие механизмы защиты:

1. Проверка роли пользователя (доступ только администраторам).
2. Использование подготовленных запросов для защиты от SQL-инъекций.
3. Валидация всех пользовательских данных перед добавлением в базу.

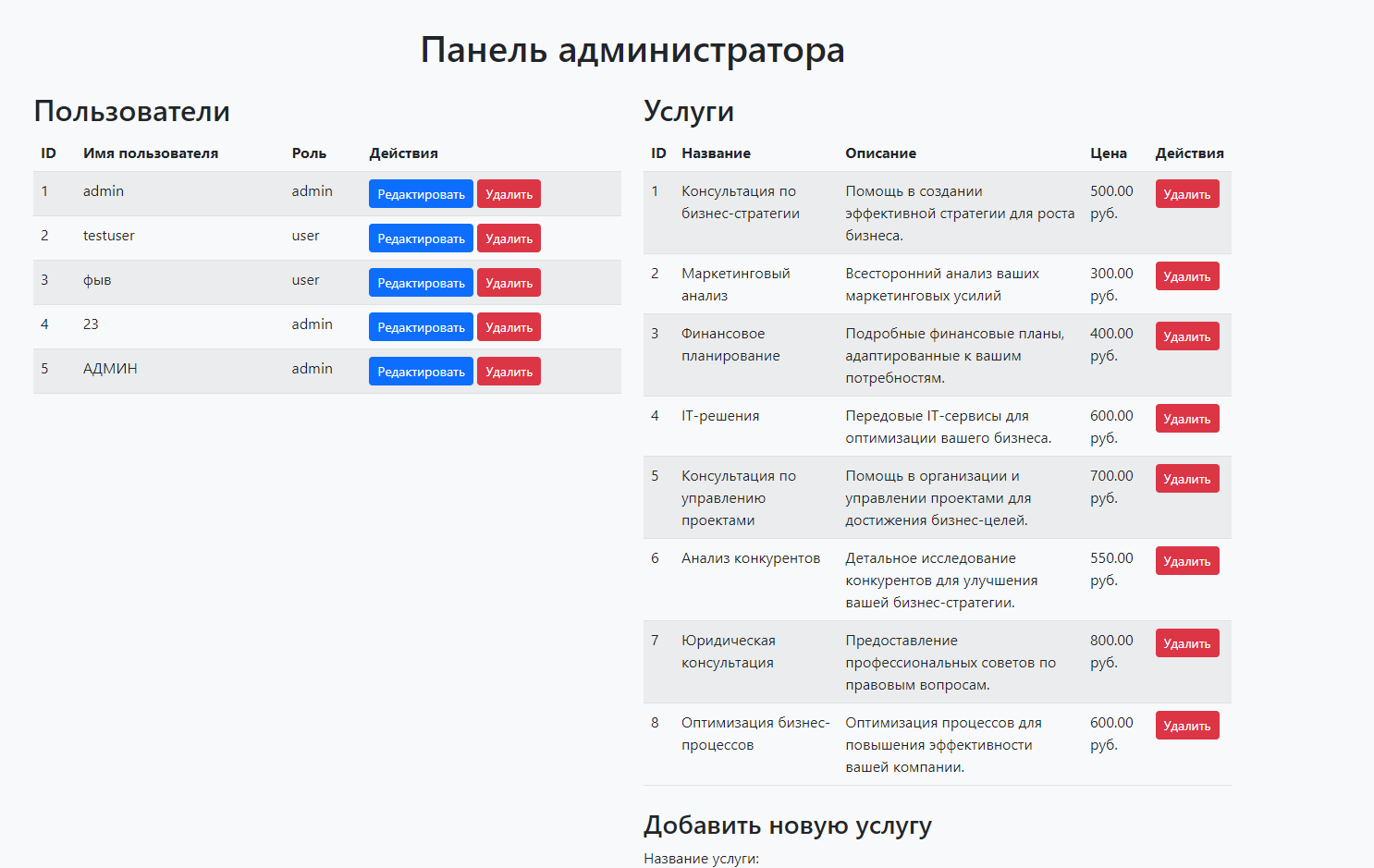


Рисунок 32 – Панель администратора

## 2.5 Стиль сайта

При создании веб-приложения стиль играет важную роль, так как он определяет внешний вид, удобство использования и общее восприятие сайта пользователем. В данном проекте был применён современный подход к оформлению интерфейса, чтобы обеспечить адаптивность, доступность и визуальную привлекательность.

Основные цели оформления:

1. **Простота и минимализм** – дизайн не перегружен лишними элементами, чтобы пользователь мог сосредоточиться на функционале сайта.
2. **Адаптивность** – сайт корректно отображается на различных устройствах, включая настольные компьютеры, планшеты и смартфоны.
3. **Единый стиль** – для всех страниц сайта используется один дизайн, что способствует лучшему восприятию и улучшает навигацию.
4. **Доступность** – элементы интерфейса учитывают стандарты доступности, например, контрастный текст, понятные шрифты и удобные для нажатия кнопки.

Для разработки стиля использовались технологии CSS и библиотека Bootstrap. Bootstrap был выбран благодаря его широкой поддержке адаптивности и готовым компонентам, что позволило быстро создать стильный интерфейс.

**Основные аспекты стиля:**

1. **Цветовая палитра**:

* Основные цвета: тёмно-синий (#343a40) и белый (#ffffff).
* Дополнительные акценты: светло-серый (#f8f9fa) для фона и яркий синий (#0d6efd) для кнопок.
* Цветовая палитра была выбрана для создания современного и профессионального образа, соответствующего тематике консалтинга.

1. **Типографика**:

* Основной шрифт: sans-serif (по умолчанию в Bootstrap используется шрифт семейства Helvetica, Arial или system-ui).
* Размер шрифта: от 14px для обычного текста до 24px для заголовков, с использованием относительных единиц измерения (em, rem) для адаптивности.
* Заголовки выделяются жирным начертанием, что помогает пользователю быстро ориентироваться.

1. **Макет**:

* Использовалась система сеток Bootstrap (Grid), которая позволяет создавать гибкие макеты.
* Сетка из 12 колонок обеспечивает адаптацию под разные размеры экрана.
* Элементы интерфейса располагаются так, чтобы пользователь всегда мог легко найти нужную информацию.

1. **Кнопки и формы**:

* Для кнопок использовался стиль Bootstrap (btn btn-primary, btn-danger), обеспечивающий удобный размер и заметный внешний вид.
* Поля форм снабжены интерактивными эффектами (например, подсветка активного поля ввода), что улучшает пользовательский опыт.

1. **Карточки**:

* Карточки Bootstrap (card) применяются для отображения услуг.
* Карточки имеют отступы, тени и закруглённые углы, что делает интерфейс более "дружественным" визуально.

**Адаптивность дизайна:**

1. **Мобильные устройства**:

* Использование классов Bootstrap (col-sm, col-md, col-lg) позволило адаптировать элементы для разных размеров экрана.
* Навигационная панель (navbar) становится "гамбургер-меню" на маленьких экранах.
* Все шрифты, кнопки и элементы ввода увеличиваются для удобного взаимодействия на устройствах с сенсорными экранами.

1. **Тестирование адаптивности**:

* Сайт был протестирован на разных устройствах (смартфонах, планшетах и ноутбуках) с использованием инструментов разработчика в браузере (Google Chrome DevTools).
* Обеспечена корректная работа интерфейса при любых разрешениях экрана.

**Пример стиля:**

* 1. **Главная страница (index.php)**
* Заголовок "Доступные услуги" оформлен в центре страницы, выделен крупным шрифтом и имеет нижний отступ для визуального разделения.
* Услуги (Рис. 33) отображаются в виде карточек с краткой информацией (название, описание, цена). Карточки равномерно распределяются по ширине экрана.

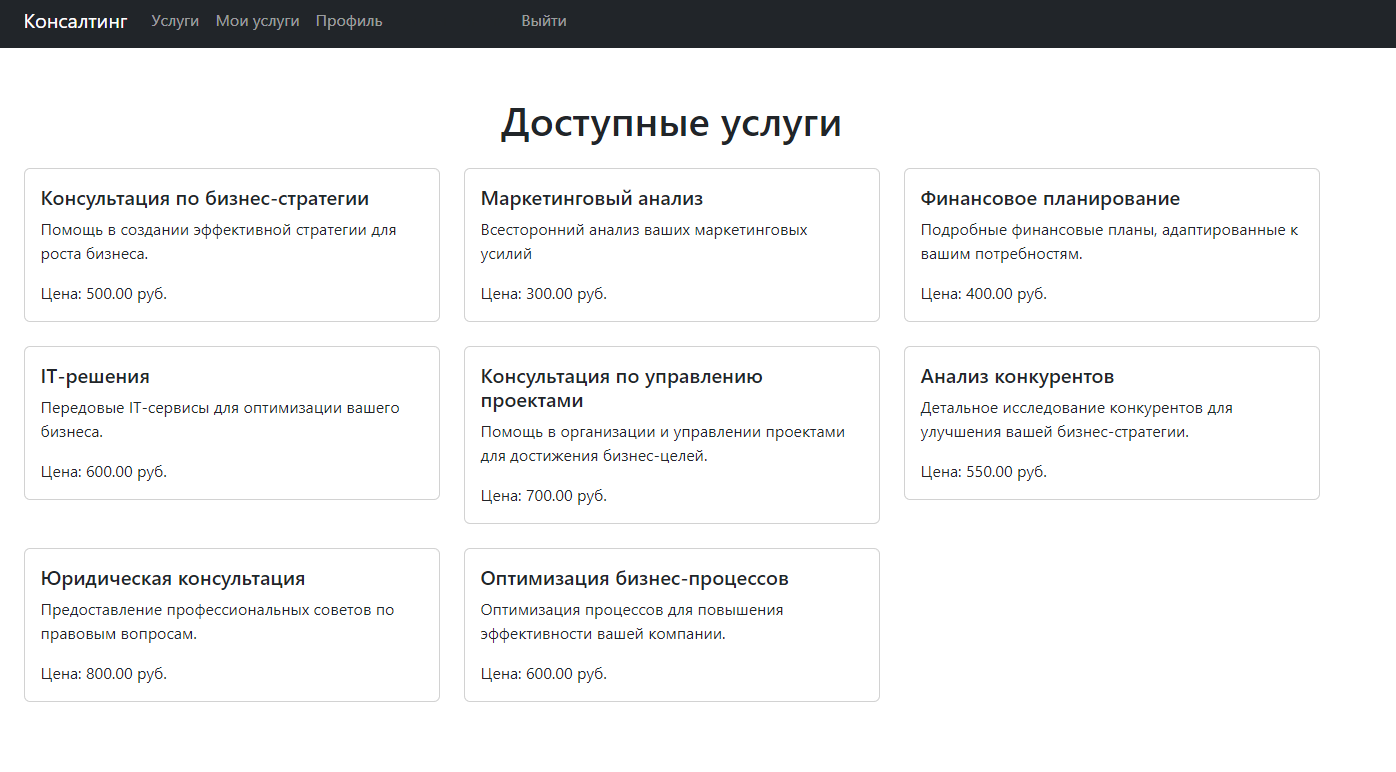


Рисунок 33 – Страница с доступными услугами

* 1. **Панель администратора (admin.php)**
* Данные о пользователях и услугах представлены в таблицах с чёткими границами строк и колонок.
* Используются кнопки для управления данными (удаление, редактирование), каждая из которых снабжена цветовым обозначением (красный для удаления).

Хотя большая часть дизайна базируется на Bootstrap, отдельные элементы были дополнительно стилизованы с помощью пользовательского CSS. Например (Рис. 34):

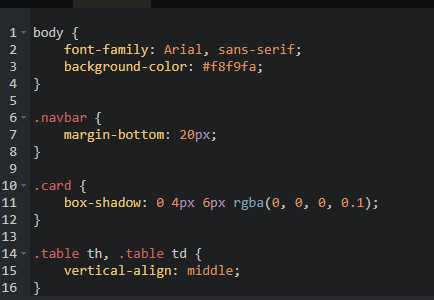


Рисунок 34 – Пользовательский CSS

Применение Bootstrap в сочетании с пользовательским CSS позволило создать современный, удобный и эстетически привлекательный интерфейс. Дизайн полностью отвечает требованиям адаптивности и поддерживает удобство взаимодействия пользователей с сайтом. Такой подход обеспечивает гибкость и универсальность приложения, улучшая общее восприятие.

# **Заключение**

В рамках данной курсовой работы была разработана и реализована система веб-приложения по тематике «Консалтинг» с применением современных технологий веб-разработки. Процесс разработки включал все этапы от планирования структуры до реализации функционала и дизайна.

Основными целями проекта были создание удобного интерфейса для пользователей и администраторов, обеспечение адаптивности приложения и разработка функционала, отвечающего заданным требованиям. В ходе выполнения работы удалось:

Реализовать систему регистрации и авторизации, что позволило разграничить доступ пользователей в зависимости от их роли.

Создать функционал для управления личными данными пользователя и предоставленных услуг.

Реализовать возможности администратора, включая добавление, изменение и удаление пользователей и услуг.

Разработать современный и адаптивный дизайн, который корректно отображается на любых устройствах.

Использовать современные технологии, такие как PHP для серверной логики, MySQL для управления базой данных, HTML и CSS для визуализации, а также Bootstrap для создания адаптивного интерфейса.

Проект соответствует всем заданным требованиям. Функционал охватывает как базовые операции (просмотр услуг, управление аккаунтом), так и административные задачи (управление пользователями и услугами).

Дополнительно была проведена работа по оптимизации дизайна с учётом принципов доступности, что повышает удобство использования приложения для различных категорий пользователей. Адаптивный интерфейс гарантирует, что сайт одинаково удобен как для настольных компьютеров, так и для мобильных устройств.

## Разработка данного приложения продемонстрировала применение ключевых технологий веб-разработки и их взаимодействие. Полученные результаты могут быть полезны для разработки других веб-приложений аналогичного характера, а также как основа для дальнейшего расширения функционала.

## В целом, работа достигла поставленных целей, а созданное приложение представляет собой функциональную, удобную и современную платформу для предоставления консалтинговых услуг.

# **Список использованных источников**

* 1. PHP // https://www.php.net: официальный сайт. URL: https://www.php.net/manual/ru/language.oop5.properties.php (дата обращения: 19.12.2024).
  2. XAMPP // https://blog.skillfactory.ru: официальный сайт. URL: https://blog.skillfactory.ru/glossary/xampp/ (дата обращения: 19.12.2024).
  3. Введение в SQL // https://netology.ru: официальный сайт. URL: https://netology.ru/programs/vvedenie-v-sql-i-rabotu-s-bazoi-dannih (дата обращения: 19.12.2024).
  4. Главное о консалтинге // https://skillbox.ru/media/: официальный сайт. URL: https://skillbox.ru/media/management/glavnoe-o-konsaltinge-kakoy-on-byvaet-i-kak-vybrat-konsaltingovuyu-kompaniyu-dlya-biznesa/ (дата обращения: 19.12.2024).
  5. Основы CSS// https://html5book.ru: официальный сайт. URL: https://html5book.ru/osnovy-css/ (дата обращения: 03.01.2025).
  6. Работа в phpMyAdmin// https://www.reg.ru: официальный сайт. URL: https://help.reg.ru/support/hosting/bazy-dannykh/rabota-v-phpmyadmin (дата обращения: 07.01.2025).