

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт информационных технологий и управления в технических системах
(полное название института)

кафедра «Информационные системы»
(полное название кафедры)

Пояснительная записка

к расчетно-графической работе
по дисциплине «Веб-технологии»

на тему «Калькулятор расчёта стоимости пошива швейных изделий»

Выполнили: студенты III курса, группы: ПИ/б-20-1-о

Направления подготовки (специальности) 09.03.03

Прикладная информатика
(код и наименование направления подготовки
(специальности))

профиль (специализация) Геоинформационные технологии

Мирзаханова Наима Маратовна, Нехода Дарья Сергеевна
(фамилия, имя, отчество студента)

Руководитель _____
(фамилия, инициалы, степень, звание, должность)

Защита «____» _____ 20 __ г. Оценка _____

Руководитель

(подпись)

(инициалы, фамилия)

2023 __ г.

АННОТАЦИЯ

Цель данной расчетно-графической работы заключается в разработке сайта-калькулятора для расчета стоимости пошива швейных изделий. Для этого были рассмотрены основные этапы создания сайта и выбраны соответствующие технологии, такие как HTML, CSS, JavaScript, PHP и MySQL.

Сайт-калькулятор состоит из пользовательского интерфейса, где пользователь может выбрать нужный тип изделия и указать необходимые параметры, длину и ширину некоторых деталей изделий. Затем происходит автоматический расчет стоимости пошива, основанный на заранее заданных ценах за метр ткани, работу мастера, сложность конструкции и другие параметры.

Результатом курсовой работы является готовый сайт-калькулятор, который обеспечит удобство заказчикам и возможность быстрой оценки стоимости пошива швейных изделий.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ

Разработке подлежит Веб-сайт «MyDress». Разрабатываемый Веб-сайт должен обеспечивать структурированное и наглядное представление услуг по расчету стоимости пошива швейных изделий, а также контактную информацию.

В разрабатываемом Веб-сайте предполагается создание двух независимых интерфейсов пользователей: интерфейс пользователей-посетителей и интерфейс пользователя-администратора и пользователя-модератора сайта.

Для интерфейса администратора/модератора сайта необходима организация авторизованного доступа. Интерфейс пользователя-администратора обеспечивает администратору сайта возможность добавлять, изменять и удалять информацию некоторых разделов сайта. Информация, добавляемая администратором, сохраняется в базе данных и отображается в соответствующих разделах пользовательского интерфейса Веб-сайта.

Интерфейс пользователей-посетителей имеет свободный доступ – и содержит структурированную информацию о компании и ее услугах. Пользовательский интерфейс содержит статические разделы (разделы, для изменения которых необходимо привлечение разработчика или стороннего Веб-программиста) и динамические разделы (содержимое динамических разделов формируется программно по информации добавленной администратором сайта и хранящейся в базе данных).

СПИСОК ИСПОЛНИТЕЛЕЙ

Сайт был разработан двумя разработчиками Мирзахановой Наимой и Неходой Дарьей.

Мирзаханова Наима разработала back-end часть сайта, а именно написал код всех контроллеров и моделей, и установил взаимодействие их друг с другом. Написал «программу-калькулятор» для расчёта стоимости услуг.

Нехода Дарья разработал front-end часть сайта, а именно написал html-разметку для всех представлений страниц сайта, так же установил для всего сайта каскадную таблицу стилей, и разработал весь дизайн.

Так же каждый из разработчиков затронул 25% зоны ответственности другого, то есть Мирзаханова Наима сделал 25% front-end разработки, а Нехода Дарья сделал 25% back-end разработки.

СОДЕРЖАНИЕ

АННОТАЦИЯ	2
ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ	3
СПИСОК ИСПОЛНИТЕЛЕЙ	4
ВВЕДЕНИЕ.....	6
1 ПОСТАНОВКА ЗАДАЧИ.....	7
2 АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	8
3 ПРОЕКТИРОВАНИЕ БД ВЕБ-САЙТА.....	9
4 ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ.....	11
5 РАЗРАБОТКА РАЗДЕЛА АДМИНИСТРАТОРА.....	12
5.1 Проектирование интерфейса раздела администратора.....	12
5.2 Разработка программных модулей раздела администратора	12
6 РАЗРАБОТКА РАЗДЕЛА ПОЛЬЗОВАТЕЛЯ.....	14
6.1 Проектирование интерфейса раздела пользователя.....	14
6.2 Разработка программных модулей раздела пользователя	14
7 ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО САЙТА	15
7.1 Анализ кроссбраузерности сайта	15
7.2 Тестовые примеры работы	17
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	19
ПРИЛОЖЕНИЕ А МИГРАЦИИ ДЛЯ СОЗДАНИЯ БД САЙТА	20
ПРИЛОЖЕНИЕ Б ИСХОДНЫЙ КОД РАЗРАБОТАННЫХ ПРОГРАММНЫХ МОДУЛЕЙ.....	22

ВВЕДЕНИЕ

В современном мире всё больше людей привлекает идея создания своего бизнеса, в том числе и в области швейных услуг. Одним из ключевых аспектов успешного ведения такого бизнеса является точный расчет стоимости пошива изделий. Для удобства клиентов и оптимизации работы мастеров возникает необходимость в создании калькулятора расчета стоимости пошива швейных изделий. В данном проекте мы будем разрабатывать сайт, где клиенты могут выбрать необходимое изделие, указать его параметры, а в ответ получить точную стоимость пошива.

WEB сайт представляет собой систему, ориентированную в основном на клиента. Она должна предоставлять услуги по расчету стоимости пошива определенного изделия, а также контактную информацию для связи с производителем. В то же время, система должна быть ориентирована на модераторе и предоставлять удобный интерфейс для менеджмента заказов.

Цель данной расчетно-графической работы – создание сайта по оказанию услуг, связанных с пошивом одежды, обеспечивающего удобство, безопасность и интуитивно понятный интерфейс, а также безошибочную, хорошо организованную работу модератора сайта.

1 ПОСТАНОВКА ЗАДАЧИ

Исходя из основных целей системы, она должна предложить следующие возможности:

- при первом посещении сайта пользователь должен понять, для чего нужен сайт, получить общее впечатление о нем, выяснить свои потребности по отношению к сайту. Для этого у него должен быть список всех услуг, доступных для него;
- если пользователь не желает пользоваться услугами магазина, он может покинуть сайт, не оставляя никаких записей о себе в базе данных сайта;
- для просмотра информации об услугах необходимо несколько страниц, которые будут находиться в каталоге;

Пользователь должен сразу видеть основные услуги, интерфейс сайта должен быть удобным и приятным для глаз. Перемещение по сайту должно быть интуитивно понятным и не вызывать у пользователя дискомфорта. Главная задача – удержать посетителя на сайте, заставить его просмотреть каталог предлагаемых магазином услуг, помочь заказчику в выборе необходимой услуги из имеющихся.

2 АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Данный сайт предназначен для облегчения процесса расчета стоимости пошива различных видов швейных изделий.

Анализ области применения показал, что разработанный сайт сможет использоваться в различных сферах, в которых требуется, пошив швейных изделий. Это могут быть магазины одежды, индивидуальные мастерские, школы и учебные заведения, где проходят курсы по шитью, а также потребители, желающие заказать, пошив изделий для персонального использования.

Кроме того, сайт может быть полезен для мастеров, которые занимаются не только пошивом изделий на заказ, но и созданием одежды для продажи в своих магазинах или на онлайн-платформах. Они могут использовать разработанный калькулятор для установления цен на свои товары с учетом затрат на материалы, время и услуги мастера.

Целевой аудиторией являются люди, желающие заказать, пошив изделия, и мастера, занимающимся швейными услугами.

3 ПРОЕКТИРОВАНИЕ БД ВЕБ-САЙТА

Для сайта необходимо было разработать БД Веб-сайта. В ходе выполнения задания было создано две диаграммы: ER- в нотации П.Чена (см. рисунок 3.1). Были отображены все сущности и их атрибуты. Независимые сущности были отображены прямоугольниками с полным названием данной сущности. Все атрибуты заключены в овал. Также были подчеркнуты все ключевые атрибуты. Зависимые сущности отличаются от независимых тем, что наименование сущности предварительно заключены в ромб, а внешние ключи – в пунктирные овалы. Были отображены связи между сущностями, подкрепленные словесным описанием отношений между ними.

В диаграмме отсутствует конкретность в ключах и связях между данными ключами. Решением данной проблемы является построение последующих уровней логической модели.

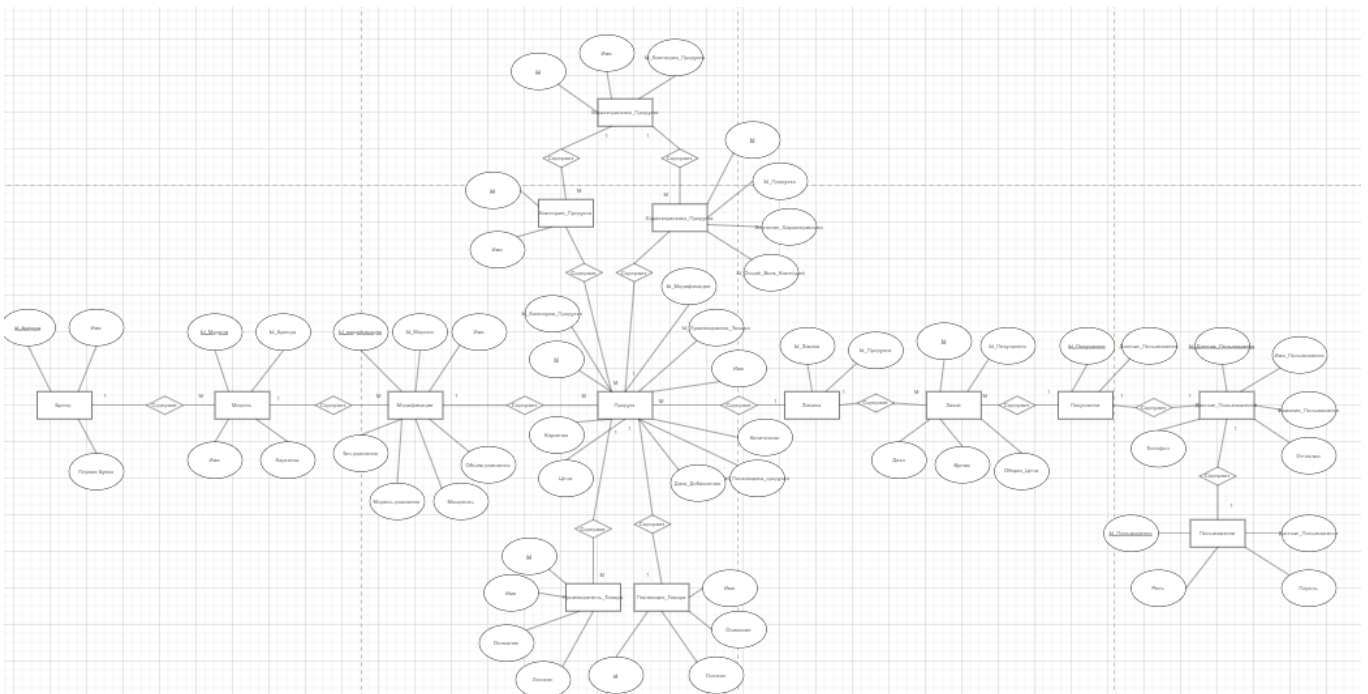


Рисунок 2.1 – Диаграмма Чена

Для установки ясности в связях между ключами всех сущностей была построена модель, основанная на ключах (см. рисунок 3.2).

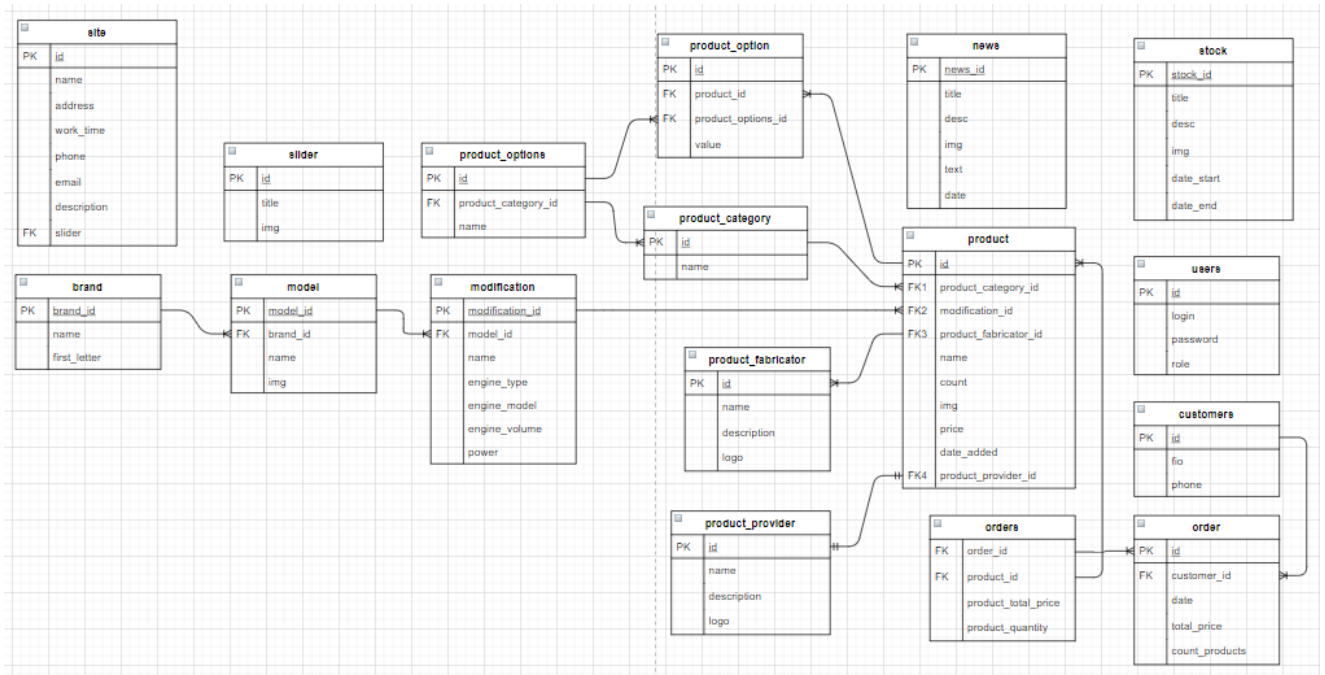


Рисунок 3.2 – Модель на ключах

На данном этапе соблюдается нормальная форма: отсутствуют сущности которые можно было бы разделить на дополнительные сущности, отсутствуют повторяющиеся значения, каждый не ключевой атрибут таблицы зависит только от первичного ключа и не зависит от других не ключевых атрибутов, в каждой сущности есть лишь один атрибут который может однозначно идентифицировать сущность.

4 ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ

В ходе разработки были использованы следующие инструменты:

- PHPStorm;
- OpenServer
- Laravel;
- PHPMyAdmin.

PHPStorm в сравнении с другими редакторами выигрывает по функциональности и в удобстве. Данный редактор является одним из лучших на сегодняшний день.

OpenServer к несомненным плюсам программы можно отнести удобный графический интерфейс пользователя, возможность просмотра логов, а также разнообразие функций для администрирования и управления компонентами.

Laravel это идеальное решение для быстрого и грамотного создания безопасного и надёжного веб-проекта, при этом всегда оставаясь на пике технологий веб-разработки.

PHPMyAdmin позволяет выполнять не сложные запросы без их написания, прост в обучении и удобен в использовании. Обладает всем необходимым функционалом для создания и поддержки БД веб-сайта.

5 РАЗРАБОТКА РАЗДЕЛА АДМИНИСТРАТОРА

5.1 Проектирование интерфейса раздела администратора

Для администратора должна быть возможность редактировать и удалять отзывы на сайте, а также удалять и добавлять новых менеджеров контента для своего сайта. Все эти функции должны быть в одном месте для удобства навигации на сайте.

Для этого необходимо разработать страницу отзывов с возможностью редактирования только для администратора и менеджера контента. А также страницу, доступную только для администратора, с возможностью редактирования модераторов. Разработать отдельную страницу – «Панель администратора». На данной панели разместить кнопки для перехода на страницу для редактирования отзывов и модераторов. Также добавить навигацию в меню сайта, объединенную в один выпадающий список.

Весь остальной функционал такой же, как и для рядового пользователя.

5.2 Разработка программных модулей раздела администратора

Для реализации навигации по страницам, доступным для администратора, в шаблон меню был добавлен следующий код:

```
@if(session('isAdmin') == 1)
    <style>
        .dropdown {
            position: relative;
            display: inline-block;
            min-width: 200px;
        }

        .dropdown-content {
            display: none;
            position: absolute;
            background-color: #000;

            min-width: 200px;
            box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
```

```

        z-index: 1;
    }

    .dropdown-content a {
        color: black;

        text-decoration: none;
        display: block;
    }

    .dropdown-content a:hover {background-color:
rgba(221, 221, 221, 0.11);}

    .dropdown:hover .dropdown-content {display:
block;}

    .dropdown:hover .dropbtn {background-color:
#882bbd;}
</style>

<div class="dropdown">
    <a class="nav-link menubase
dropbtn">Администрирование</a>
    <div class="dropdown-content">
        <a class="nav-link nav-admin" style="margin-
left: 0;" href="/adminpanel">Панель администратора</a>
        <a class="nav-link nav-admin" style="margin-
left: 0;" href="/reviews">Редактор отзывов</a>
        <a class="nav-link nav-admin" style="margin-
left: 0;" href="/moders_editor">Редактор модераторов</a>
    </div>
</div>

    <a class="nav-link ms-5" href="/reviews"></a>
    <a class="btn btn-light ms-2"
href="/adminlogout">Админ: {{session('nickname')}} - выйти</a>
@endif

```

Для страницы отзывов был добавлены функционал для редактирования с использованием технологии AJAX:

```

@if(session('isModer') == 1 | session('isAdmin') == 1)
    <div class="nav-masthead">
        <a
href="/reviews_editor/{{ $d['id'] }}"></a>
        <a
href="/reviews_editor/{{ $d['id'] }}/delete"></a>
    </div>
@endif

```

Код для редактирования отзывов и модераторов представлен в приложении В.

6 РАЗРАБОТКА РАЗДЕЛА ПОЛЬЗОВАТЕЛЯ

6.1 Проектирование интерфейса раздела пользователя

Авторизованный пользователь имеет возможность рассчитать стоимость различных швейных изделий, оставить отзывы, просмотреть ткани.

Для удобства ориентирования на сайте должна быть навигация по его страницам.

6.2 Разработка программных модулей раздела пользователя

Так как меню одинаково на каждой странице, меню и основной стиль сайта был добавлен в шаблон. Код шаблона представлен в приложении Б.

На странице отзывов был добавлен следующий код:

```
<div class="container">
    <textarea class="form-control mt-3"
    id="reviewtext" name="reviewtext">

    </textarea>

    <script>
        tinymce.init({
            selector: 'textarea',
            plugins: 'anchor autolink charmap
codesample emoticons image link lists media searchreplace table visualblocks
wordcount checklist mediaembed casechange export formatpainter pageembed
linkchecker allychecker tinymcespellchecker permanentpen powerpaste advtable
advcode editimage tinycomments tableofcontents footnotes mergetags
autocorrect typography inlinecss',
            toolbar: 'undo redo | blocks
fontfamily fontsize | bold italic underline strikethrough | link image media
table mergetags | addcomment showcomments | spellcheckdialog allycheck
typography | align lineheight | checklist numlist bullist indent outdent |
emoticons charmap | removeformat',
            tinycomments_mode: 'embedded',
            tinycomments_author: 'Author
name',
            mergetags_list: [
                { value: 'First.Name', title:
'First Name' },
                { value: 'Email', title:
'Email' },
            ]
        });
    </script>
</div>
```

7 ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО САЙТА

7.1 Анализ кроссбраузерности сайта

Кроссбраузерность – это правильная верстка сайта с помощью, которой страницы сайта одинаково отображаются в различных браузерах. Реализация происходит при помощи HTML и CSS. Для правильного отображения сайта одновременно в Internet Explorer, FireFox, Opera, Safari, Chrome, причем самых разных версий (от более ранних до самых новых), веб-дизайнер обязательно должен позаботиться о кроссбраузерности проекта, сайта с первой секунды работы над ним.

Веб-сайт поддерживает все основные современные браузеры, а именно: Google Chrome, Mozilla Firefox, Internet Explorer, Opera и т.д. (рисунок 7.1- 7.3).

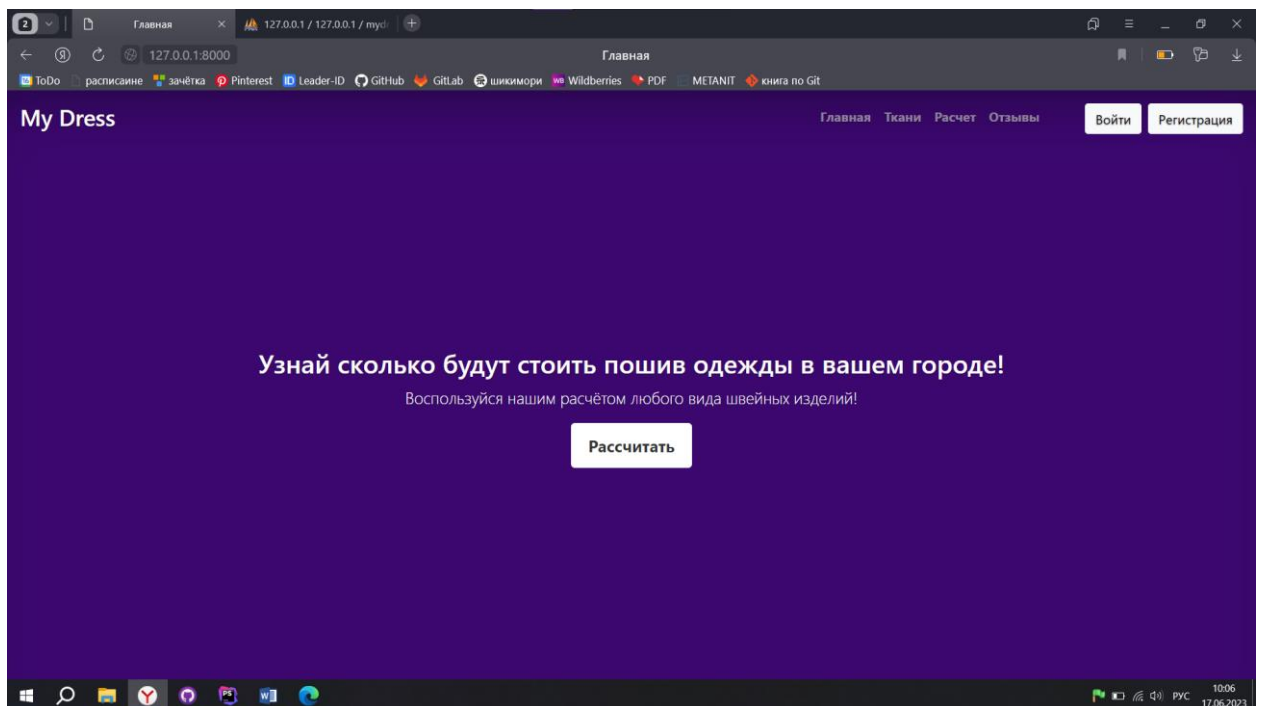


Рисунок 7.1 – Демонстрация кроссбраузерности сайта на примере Яндекс

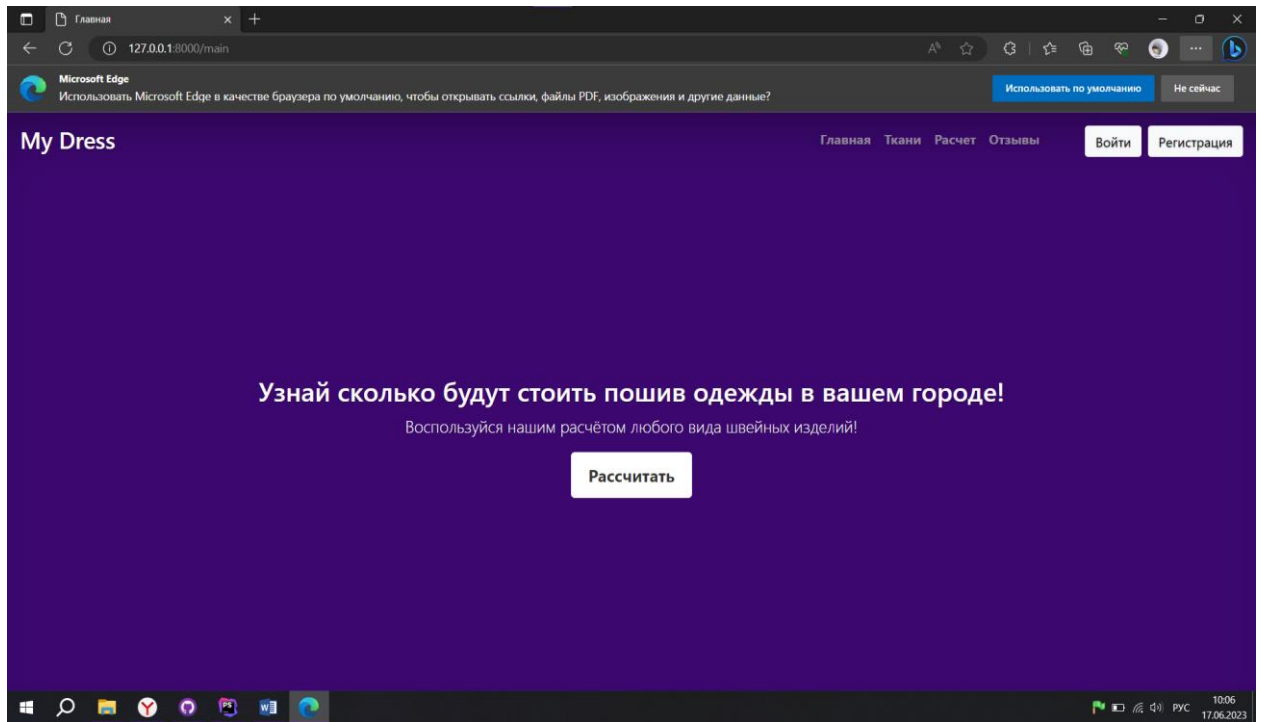


Рисунок 7.2 – Демонстрация кроссбраузерности сайта на примере Edge

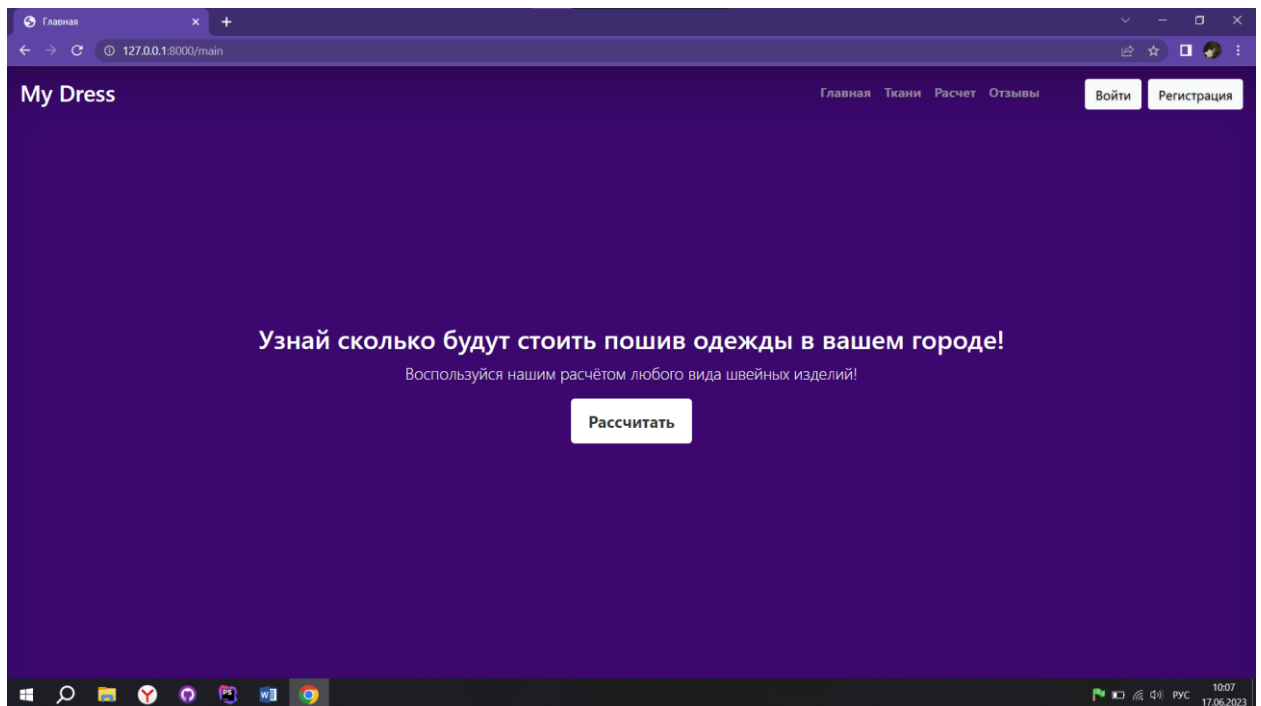


Рисунок 7.3 – Демонстрация кроссбраузерности сайта на примере Chrome

7.2 Тестовые примеры работы

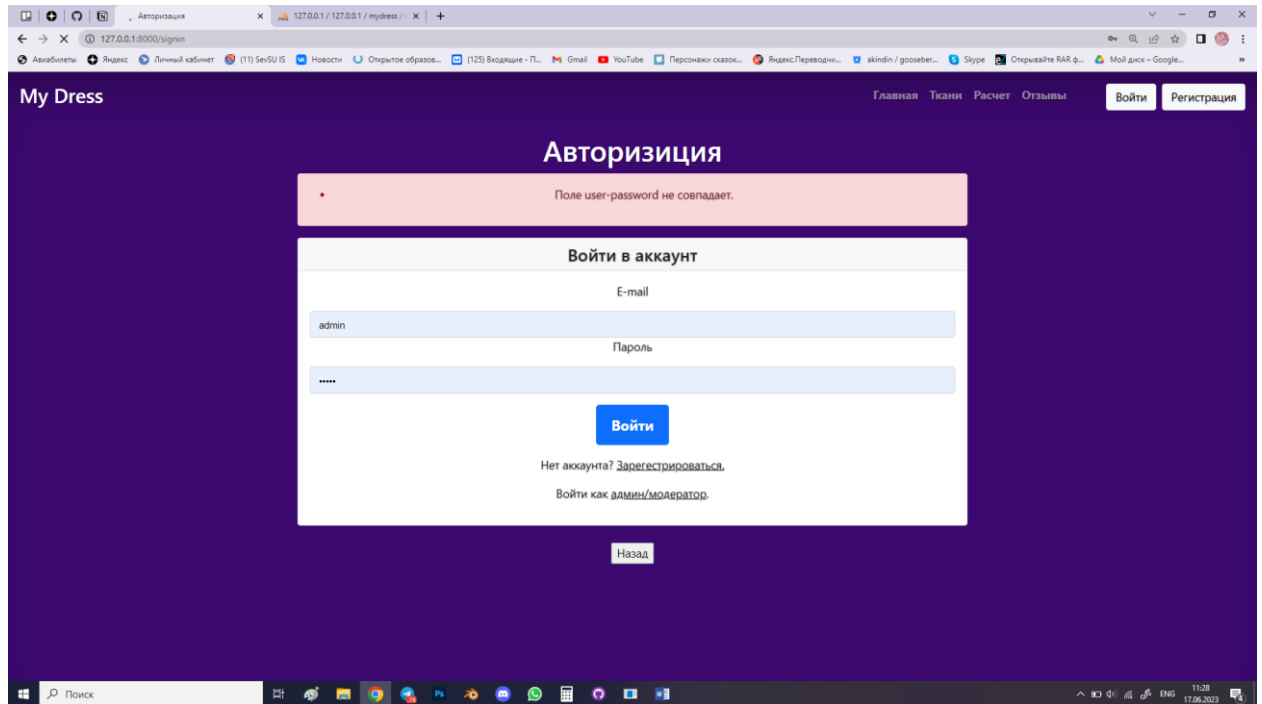


Рисунок 7.4 – Вход на страницу с неправильным паролем.

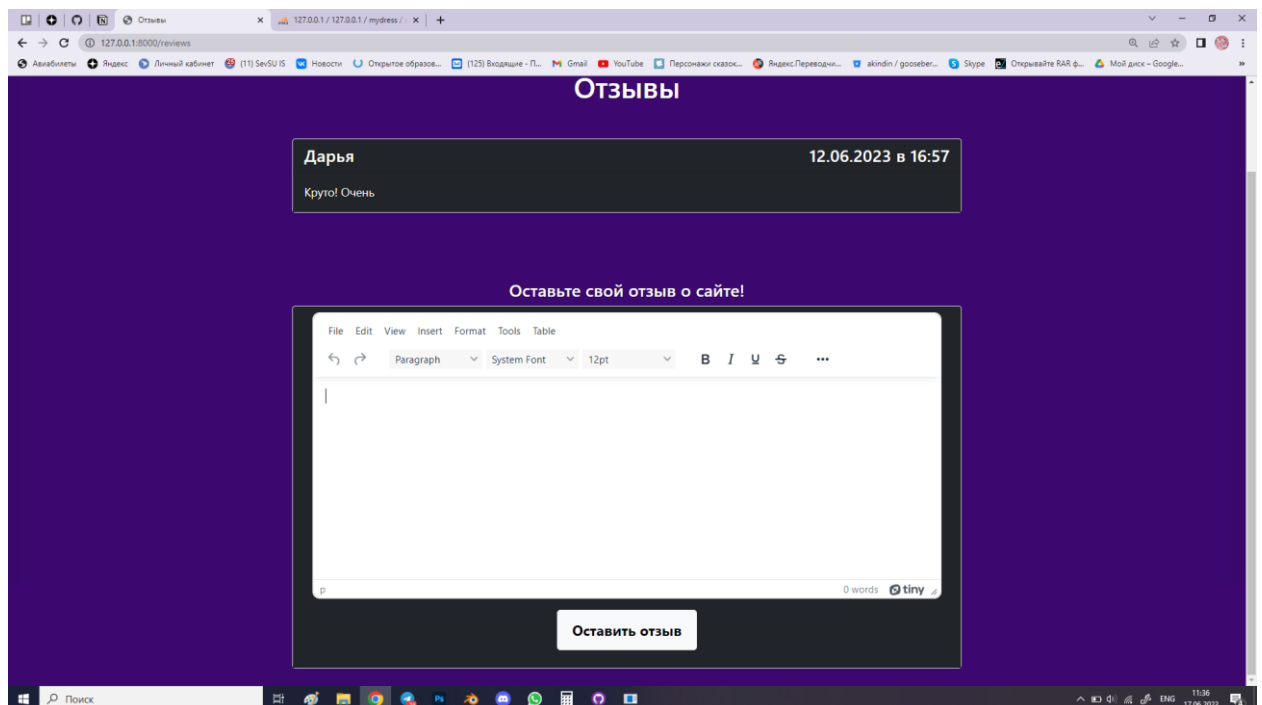


Рисунок 7.4 – Добавление отзывов

ЗАКЛЮЧЕНИЕ

В результате работы был разработан сайт, который дает возможность заказчику просмотреть необходимые услуги и рассчитать стоимость их выполнения. Помимо этого, был реализован весь необходимый функционал:

- реализована навигация с помощью меню на страницах сайта;
- расчет стоимости выполнения услуги;
- публикация отзыва;
- возможность связи с модератором для получения консультации.

Проанализирована предметная область, а также целевая аудитория сайта. Дизайн сайта соответствует ожидаемым предпочтениям целевой группы, времени и целям поиска потенциальных клиентов на сайте. Получены практические навыки работы в команде и в разработке веб-сайта.

Цель курсового проекта была достигнута. Программное средство предоставляет информацию по услугам выполнения швейных работ и примерную стоимость выполнения выбранных работ при заданных параметрах.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. 1 Веб-приложение для работы с базами данных с PHP и MySQL, 2-е издание Дэвида Лейна, Хью Э. Уильямса. О'Райли, май 2003. ISBN: 0-596-00543-1.
2. Список CMS. Обзор cms. Сайт о системах управления сайтом. <http://www.cmslist.ru>.
3. Материал из Википедии – бесплатная энциклопедия о системах управления сайтом. <http://ru.wikipedia.org/wiki/CMS>.
4. Дарахвелидзе П.Г., Марков Е.П. Программирование в Дельфах 7. – С.-Петербург: BHVP, 2003. – 784 с.
5. Калверт С. Delphi 4-я энциклопедия пользователя: Пер. с английского / Чарльз Калверт. – С.: Издательский дом «ДиаСофт», 1999 г. – 800 с.
6. Каратыгин С., Тихонов А., Долголаптев В. Базы данных: простейшие средства обработки информации, электронные таблицы, системы управления базами данных
7. Матросов А.В., Сергеев А.О., Чаунин М.П. HTML 4.0. – Санкт-Петербург: BHVP, 2001. – 672 с.
8. Мещеряков Е.В., Хомоненко А.Д. Публикация баз данных в сети Интернет. – Санкт-Петербург: BHVP Petersburg, 2000. – 560 с.
9. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера: Пособие / Прохоренок Н.А., - 4-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2015. - 768 с. ISBN 978-5-9775-3130-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/943563> (дата обращения: 01.06.2020)

ПРИЛОЖЕНИЕ А**Миграции для создания БД сайта****Миграция для таблицы User**

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Миграция для таблицы Review

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateReviewModelsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
```

```

{
    Schema::create('review_models', function (Blueprint $table) {
        $table->id();
        $table->string('user_email');
        $table->foreign('user_email')->references('email')-
>on('user_models')->onDelete;
        $table->text('review_text');
        $table->string('timestamp');
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('review_models');
}
}

```

Миграция для таблицы Admin

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateAdminModelsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('admin_models', function (Blueprint $table) {
            $table->string('nickname');
            $table->string('password');
            $table->primary('nickname');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('admin_models');
    }
}

```

ПРИЛОЖЕНИЕ Б

Исходный код разработанных программных модулей

Код контроллера DBController

```
<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use App\Models\AdminModel;
use App\Models\ModeratorModel;
use App\Models\ReviewModel;
use App\Models\UserModel;
use Illuminate\Http\Request;

class DBController extends Controller
{
    public function CreateUser(Request $request)
    {
        $valid = $request->validate([
            'user-name' => 'required',
            'user-email' => 'required',
            'user-password' => 'required|min:4',
            'password-confirmation' => 'required|min:4',
        ]);

        $user = new UserModel();

        $user->email = $request->input('user-email');
        $user->FIO = $request->input('user-name');
        $user->password = $request->input('user-password');

        $user->save();

        session(['isUser' => 1]);
        session(['username' => $request->input('user-name')]);
        session(['email' => $request->input('user-email')]);

        session(['isUser' => 1]);
        return view('login');
    }

    public function Logout()
    {
        session(['isUser' => 0]);
        return redirect()->route('main');
    }

    public function Adminlogout()
    {
        session(['isModer' => 0]);
        session(['isAdmin' => 0]);
        return redirect()->route('main');
    }
}
```

```

public function Signin(Request $request)
{
    $valid = $request->validate([
        'user-email' => 'required',
        'user-password' => 'required|min:4',
    ]);
    $useremail = UserModel::select("*")->where("email", $request->input('user-email'));

    if($useremail->exists() && UserModel::select("*")->where("password", $request->input('user-password'))->exists()){
        session(['isUser' => 1]);
        session(['username' => UserModel::find($request->input('user-email'))->FIO]);
        session(['email' => $request->input('user-email')]);
    }
    else
    {
        $valid = $request->validate([
            'user-password' => 'confirmed',
        ]);
    }

    return view('signin');
}

public function Adminsignin(Request $request)
{
    $valid = $request->validate([
        'user-nickname' => 'required',
        'user-password' => 'required|min:4',
    ]);

    if ($request->input('type_account') == 0)
    {
        $usernickname = ModeratorModel::select("*")->where("nickname", $request->input('user-nickname'));
        if($usernickname->exists() && ModeratorModel::select("*")->where("password", $request->input('user-password'))->exists()){
            session(['isModer' => 1]);
            session(['nickname' => $request->input('user-nickname')]);
        }
    }
    else
    {
        $usernickname = AdminModel::select("*")->where("nickname", $request->input('user-nickname'));
        if($usernickname->exists() && AdminModel::select("*")->where("password", $request->input('user-password'))->exists()){
            session(['isAdmin' => 1]);
            session(['nickname' => $request->input('user-nickname')]);
        }
    }

    return view('adminsignin');
}

public function Adminpanel()
{
    return view('adminpanel');
}

public function Moderpanel()
{
    return view('moderpanel');
}

```

```

    }

    public function ModeratorEditor()
    {
        $data = [];
        if (ModeratorModel::count() == 0)
        {
            return view('moders_editor', ["empty" => 1]);
        }
        $datas = ModeratorModel::all();
        foreach ($datas as $d)
        {
            $data[$d['nickname']] = [
                'nickname' => $d['nickname'],
                'password' => $d['password'],
            ];
        }
        return view('moders_editor', ["data" => $data]);
    }

    public function CreateModerator(Request $request)
    {
        $valid = $request->validate([
            'nickname' => 'required',
            'password' => 'required|min:4',
        ]);

        $user = new ModeratorModel();

        $user->nickname = $request->input('nickname');
        $user->password = $request->input('password');

        $user->save();

        $data = [];
        if (ModeratorModel::count() == 0)
        {
            return view('moders_editor', ["empty" => 1]);
        }
        $datas = ModeratorModel::all();
        foreach ($datas as $d)
        {
            $data[$d['nickname']] = [
                'nickname' => $d['nickname'],
                'password' => $d['password'],
            ];
        }
        return view('moders_editor', ["data" => $data]);
    }

    public function DeleteModerator($moder)
    {
        ModeratorModel::where('nickname', $moder)->delete();

        return redirect()->route('moders_editor');
    }
}

```

Код контроллера ReviewsController

```
<?php
```

```
namespace App\Http\Controllers;
```



```

use App\Http\Controllers\Controller;
use App\Models\ReviewModel;
use Illuminate\Http\Request;

class ReviewsController extends Controller
{
    public function Reviews()
    {
        //ReviewModel::find(1)->user['FIO'];
        //ReviewModel::find(1)->all();
        $data = [];
        if (ReviewModel::count() == 0)
        {
            return view('reviews', ["empty" => 1]);
        }
        $datas = ReviewModel::find(1)->all();
        foreach ($datas as $d)
        {
            $data[$d['id']] = [
                'id' => $d['id'],
                'fio' => ReviewModel::find($d['id'])->user['FIO'],
                'text' => $d['review_text'],
                'timestamp' => $d['timestamp'],
            ];
        }
        return view('reviews', ["data" => $data]);
    }

    public function AddReview(Request $request)
    {
        $valid = $request->validate([
            'reviewtext' => 'required',
        ]);

        $user = new ReviewModel();

        $user->user_email = session('email');
        $user->review_text = $request->input('reviewtext');
        $user->timestamp = date('d.m.Y \B H:i');

        $user->save();

        $data = [];
        $datas = ReviewModel::find(1)->all();
        foreach ($datas as $d)
        {
            $data[$d['id']] = [
                'id' => $d['id'],
                'fio' => ReviewModel::find($d['id'])->user['FIO'],
                'text' => $d['review_text'],
                'timestamp' => $d['timestamp'],
            ];
        }

        return view('reviews', ["data" => $data]);
    }

    public function EditReview($id)
    {
        $data = [];
        if (ReviewModel::count() == 0)
        {
            return view('reviews', ["empty" => 1]);
        }
    }
}

```

```

    $datas = ReviewModel::findOrFail(1)->all();
    $data['review_text'] = $this->FindInData($datas, $id)['review_text'];
    $data['id'] = $id;

    return view('reviews_editor', ['data' => $data]);
}

public function EditReviewResult(Request $request, $id)
{
    $valid = $request->validate([
        'reviewtext' => 'required',
    ]);

    ReviewModel::where('id', $id)->update(array('review_text' =>
    $request->input('reviewtext')));

    return redirect()->route('reviews');
}

public function DeleteReview($id)
{
    ReviewModel::where('id', $id)->delete();

    return redirect()->route('reviews');
}

public function FindInData($datas, $id)
{
    foreach ($datas as $d)
    {
        if ($d['id'] == $id)
            return $d;
    }
}
}

```

Код представления страницы Layout

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>@yield("page-title")</title>
    <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scri
pt>

    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css
" rel="stylesheet"/>
    <link href="https://getbootstrap.com/docs/4.0/examples/cover/cover.css"
rel="stylesheet">
    <link
href="https://getbootstrap.com/docs/4.0/examples/pricing/pricing.css"
rel="stylesheet">
    <link href="css/app.css" rel="stylesheet">

</head>

<body class="text-center">
    <div class="container-fluid d-flex h-100 p-3 mx-auto flex-column">
        <header class="masthead">
            <div class="inner">

```

```

<h3 class="masthead-brand">My Dress</h3>
<nav class="nav nav-masthead justify-content-center">
  <a class="nav-link" href="/main">Главная</a>
  <a class="nav-link" href="/payment">Расчет</a>
  <a class="nav-link" href="/reviews">Отзывы</a>
  <a class="nav-link ms-5" href="/reviews"></a>
  @if(session('isModer') == 1)
    <a class="nav-link" href="/moderpanel">Панель
модератора</a>
    <a class="nav-link" href="/reviews">Редактор
отзывов</a>
    <a class="nav-link ms-5" href="/reviews"></a>
    <a class="btn btn-light ms-2"
href="/adminlogout">Модератор: {{session('nickname')}} - выйти</a>
  @endif
  @if(session('isAdmin') == 1)
    <a class="nav-link" href="/adminpanel">Панель
администратора</a>
    <a class="nav-link" href="/reviews">Редактор
отзывов</a>
    <a class="nav-link" href="/moders_editor">Редактор
модераторов</a>
    <a class="nav-link ms-5" href="/reviews"></a>
    <a class="btn btn-light ms-2"
href="/adminlogout">Админ: {{session('nickname')}} - выйти</a>
  @endif
  @if(session('isUser') != 1)
    <a class="btn btn-light ms-2" href="signin">Войти</a>
    <a class="btn btn-light ms-2"
href="login">Регистрация</a>
  @endif
  @if(session('isUser') == 1)
    <a class="btn btn-light ms-2" href="/logout">Хорошая
работа, {{session('username')}}</a>
  @endif
</nav>
</div>
</header>

<main role="main" class="inner cover">

  @yield("page-content")
</main>
</div>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scri
pt>
</body>
</html>

```

Код представления страницы Reviews

```

@extends("layout")

@section("page-title") Отзывы @endsection

@section("page-content")
  <h1 class="m-5">Отзывы</h1>
  <div class="container">
    @if($errors->any())
      <div class="alert alert-danger">
        <ul>
          @foreach($errors->all() as $error)
            <li>{{ $error }}</li>
          @endforeach
        </ul>
      </div>
    @endif
  </div>
</section>

```

```

        @endforeach
    </ul>
</div>
@endif
@if(!empty($data))
    @foreach($data as $d)
        <div class="card-deck mb-3">
            <div class="card mb-4 box-shadow">
                <div class="card-header bg-dark text-lg-start">
                    <h4 class="masthead-brand my-0 font-weight-
normal">{{ $d['fio'] }}</h4>
                    <h4 class="nav-masthead my-0 font-weight-
normal">{{ $d['timestamp'] }}</h4>
                </div>
                <div class="card-body bg-dark">
                    <p class="masthead-brand text-lg-
start">{{ $d['text'] }}</p>
                    @if(session('isModer') == 1 |
session('isAdmin') == 1)
                        <div class="nav-masthead">
                            <a
href="/reviews_editor/{{ $d['id'] }}"></a>
                            <a
href="/reviews_editor/{{ $d['id'] }}/delete"></a>
                        </div>
                    @endif
                </div>
            </div>
        </div>
    @endforeach
@endif
@if(session('succes') == 1)
    <div class="alert alert-success">
        Отзыв отправлен
    </div>
@endif
@if(session('isUser') == 1)
    <form action="/reviews" method="post">
        @csrf
        <h4 style="margin-top: 10%">Оставьте свой отзыв о
сайте!</h4>
        <div class="card-deck mb-3">
            <div class="card mb-4 box-shadow">
                <div class="card-header bg-dark">
                    <div class="container">
                        <textarea class="form-control mt-3"
id="reviewtext" name="reviewtext"></textarea>
                    </div>
                    <button type="submit" href="/main" class="btn
btn-lg btn-block btn-light m-3">Оставить отзыв</button>
                </div>
            </div>
        </div>
    </form>
@endif
</div>
@if(session('isUser') != 1)
    <p class="lead">Чтобы оставить отзыв о сайте или о проделанной работе
необходимо <a href="signin">авторизироваться</a></p>
@endif
@endsection

```

Код для редактирования отзывов

```
@extends("layout")

@section("page-title") Редактировать отзыв @endsection

@section("page-content")
    <h1 class="m-5">Редактировать отзыв</h1>
    <form action="/reviews_editor/@if(!empty($data)){{ $data['id'] }}@endif"
method="post">
        @csrf
        <div class="container">
            @if($errors->any())
                <div class="alert alert-danger">
                    <ul>
                        @foreach($errors->all() as $error)
                            <li>{{ $error }}</li>
                        @endforeach
                    </ul>
                </div>
            @endif
            <textarea class="form-control" id="reviewtext"
name="reviewtext">@if(!empty($data)){{ $data['review_text'] }}@endif</textarea>
        </div>
        <button type="submit" href="/main" class="btn btn-lg btn-block btn-
dark m-3">Закончить редактирование</button>
    </form>
@endsection
```

Код для редактирования менеджеров по контенту

```
@extends("layout")

@section("page-title") Редактор модераторов @endsection

@section("page-content")
    <h1 class="m-5">Редактор модераторов</h1>
    <div class="container">
        @if($errors->any())
            <div class="alert alert-danger">
                <ul>
                    @foreach($errors->all() as $error)
                        <li>{{ $error }}</li>
                    @endforeach
                </ul>
            </div>
        @endif
        @if(!empty($data))
            @foreach($data as $d)
                <div class="card-deck mb-3">
                    <div class="card mb-4 box-shadow">
                        <div class="card-header bg-dark text-lg-start">
                            <h4 class="masthead-brand my-0 font-weight-
normal">Имя: {{ $d['nickname'] }}</h4>
                            <h4 style="margin-left: 20%" class="masthead-
brand my-0 font-weight-normal">Пароль: {{ $d['password'] }}</h4>
                            <div class="nav-masthead">
                                <a
href="/moders_editor/{{ $d['nickname'] }}/delete"></a>
                            </div>
                        </div>
                    </div>
                </div>
            @endforeach
        @endif
    </div>
@endsection
```

```

@endif
    @if(!empty($data['succes']) && $data['succes'] == 1)
        <div class="alert alert-success">
            Модератор добавлен
        </div>
    @endif
    <form action="/moders_editor" method="post">
        @csrf
        <h4 style="margin-top: 10%">Добавить модератора</h4>
        <div class="card-deck mb-3">
            <div class="card mb-4 box-shadow">
                <div class="card-header bg-dark text-light">
                    <div class="nav">
                        <input type="text" name="nickname"
class="msthead-brand" style="margin-top: auto; margin-bottom: auto; margin-
right: 10%; margin-left: 10%" placeholder="Имя модератора">
                        <input type="password" name="password"
class="nav-msthead" style="margin-top: auto; margin-bottom: auto; margin-
right: 10%;" placeholder="Пароль модератора">
                        <button type="submit" href="" class="btn btn-
light ms-2 m-3">Добавить</button>
                    </div>
                </div>
            </div>
        </div>
    </form>
</div>
@endsection

```