**Введение**

В нашем обществе улучшение жилищных условий и создание комфортного пространства становятся важной задачей для многих, особенно когда речь идет о ремонте квартир. Специализированные компании и онлайн-платформы предлагают качественные и стильные решения для таких нужд..

С развитием технологий и ростом популярности онлайн-сервисов, люди все чаще обращаются к интернет-ресурсам для поиска услуг по ремонту и отделке квартир. Это создает спрос на удобные и быстрые сайты, где можно найти информацию о ремонтных услугах, различных стилях и материалах, удовлетворяющих любые предпочтения..

В рамках этого проекта планируется использование современных технологий, таких как HTML, CSS и JavaScript, для создания сайта, предлагающего полный спектр услуг по ремонту квартир в Москве. Основная цель проекта — создать функциональный и привлекательный веб-ресурс с интуитивно понятным интерфейсом и быстрой загрузкой страниц, обеспечивающий пользователям доступ к необходимой информации и услугам.

**1 Анализ задачи**

**1.1 Постановка задачи**

На учебную практику по WEB ­– программированию была поставлена задача по верстке шаблона по макету из Figma на тему: «Ремонт квартир».

Необходимо сверстать главную страницу сайта по макету, представленному в виде figma- ссылки: https://www.figma.com/design/PP43LVeIX6mLjPAQ3NBJtA/MaxProff-%2B?m=auto&t=HWgdidFW1BvVl857-6

Задача: Разработка сайта по продаже мебели с помощью HTML, CSS и JS.

Главная страница должна быть сверстана с учётом просмотра её на различных устройствах с разными разрешениями экрана, то есть иметь адаптивную верстку для просмотра на компьютере при различных разрешениях (ширина по умолчанию - 1440рх, макет соответствует ширине 800 рх), а также на мобильном телефоне (ширина экрана менее 420 рх).

Также должен быть составлен документ, который содержит в себе список элементов правил, подходящих под стилистику конкретного сайта - стайлгайд. Стайлгайд - это набор стандартов и требований, обязательных к соблюдению. Их необходимо использовать в процессе оформления сайта. Ключевое назначение - создание единого стилистического оформительского однообразия. На основании данных из Figma стайлгайд должен включать себя следующие разделы:

**—** логотип;

**—**  цветовая палитра;

**—**  типографика;

**—**  сетка и отступы;

**—**  навигация, меню;

**—**  иконки;

**—**  кнопки, поля ввода;

**—**  компоненты;

**—** пользовательские формы;

**—**  модальные окна, алерты и т.д.

После завершения составления стайлгайда и вёрстки макета необходимо выполнить стилизацию при помощи CSS отдельных элементов макета в соответствии со следующими с критериями:

**—**  hover-эффекты;

**—**  оформление гиперссылок;

**—**  тени;

**—**  градиенты;

**—**  оформление изображений;

**—** оформление(стилизация) пользовательских форм;

**—**  трансформации;

**—**  анимация и переходы;

**—**  декоративные элементы;

**—**  любые другие эффекты.

Далее необходимо добавить в сверстанный шаблон главной страницы интерактивные, динамические элементы. При их разработке необходимо писать код на JavaScript и использовать библиотеки Jquery, Jquery UI в соответствии со следующими критериями:

**—**  работа с окнами сообщений;

**—**  создание динамического меню и организация навигации;

**—**  слайдеры;

**—**  работа с изображениями через JS;

**—**  использование библиотеки Jquery;

**—**  использование библиотеки Jquery UI;

**—**  использование любых возможных способов JS+Jquery.

**1.2 Инструменты разработки**

Для вёрстки данного шаблона макета будет выбраны следующие среды для разработки:

**—**  Microsoft Visual Studio Code с расширением Live server

**—**  WEB Storm.

А Также для дальнейшей работы с проектом потребуется Figma и GitHub.

Microsoft Visual Studio Code – среда разработки. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

WebStorm — мощный редактор кода от JetBrains для кроссплатформенной веб-разработки. Включает отладчик, поддержку Git, подсветку синтаксиса, интеллектуальное автодополнение и анализ кода, а также инструменты рефакторинга. Поддерживает кастомизацию с помощью тем, сочетаний клавиш и файлов конфигурации.

Figma - онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.

GitHub - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

При верстке данного шаблона использовался персональный компьютер со следующими характеристиками:

**—** процессор: AMD Ryzen 5 5600H

* объем ОЗУ 16,00 ГБ;
* объем места на SSD: 512 GB;
* ОС: Windows 11 pro.

**2 Проектирование задачи**

**2.1 Стайлгайд**

**2.1.1 Логотип и сетка**

Логотип - графический знак, эмблема или символ, используемый территориальными образованиями, коммерческими предприятиями, организации и частными лицами для повышения узнаваемости и распознаваемости в социуме. Логотип представляет собой название сущности, которую он идентифицирует, в виде стилизованных букв и/или идеограммы (рисунок 1).



Рисунок 1 - Логотип

Модульная сетка сайта для персонального компьютера (ширина 1440рх), мобильного телефона (ширина 420 рх) представлена в приложении А.

**2.1.2 Цветовая палитра сайта**

Цветовая палитра:

* Основной цвет: #6CAC3D (Зеленый)
* Вторичный цвет: #2D4872 (Синий)
* Акцентный цвет: #DC3545 (Красный)
* Фоновый цвет: #FFFFFF (Белый)

**2.1.3 Типографика**

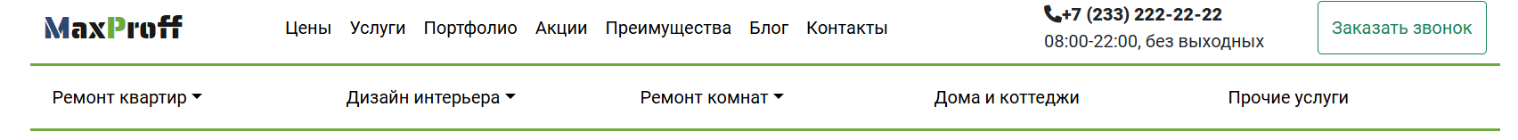
Типографика - сильнейший инструмент для выражения посыла в веб дизайне. С его помощью вы можете объединить текстовую и визуальную составляющие, что поможет вам достучаться до посетителя. Типографика для шаблона макета «Шторы» представлена ниже:

* Основной шрифт: Roboto;
* Дополнительный шрифт: Saira Stencil One;
* Размер шрифта: 8-28px;
* Интервал между строками в среднем составляет: 5-18px;
* Цвет: #000000 (черный), # 6CAC3D (зеленый), #FFFFFF (белый), #2D4872 (Синий)

**2.1.4 Навигация, меню**

Важнейшей составляющей любого сайта является навигация и меню. Частая ошибка при создании дизайна сайта – это сложная или запутанная навигация, которая отталкивает клиента, что в последствии может принести убыток.

Простое правило: чем проще ориентироваться на сайте, тем быстрее пользователь найдёт необходимую информацию.

****В шапке сайта пользователь без трудностей может перейти к различным разделам сайта, а также имеет возможность заказать звонок.

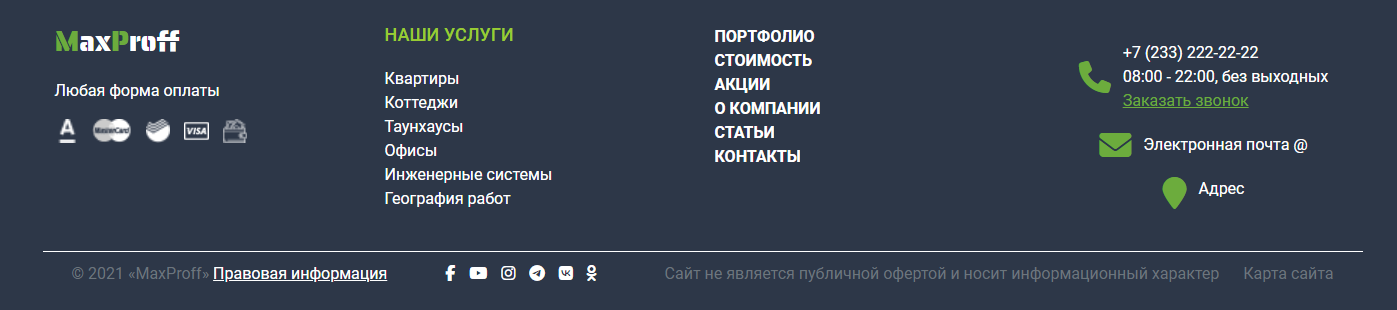


Рисунок 2 – Шапка сайта

Подвал данного макета схож с шапкой, а также имеет более подробное описание различных разделов (рисунок 3).

Рисунок 3 – Подвал

**2.1.5 Иконки**

Иконка — это максимально упрощённое изображение объекта или действия. Иконки идеально вписываются в современные тренды веб-дизайна. Помимо этого, у иконок есть ещё одно неоспоримое преимущество перед - отсутствие языкового барьера. Иллюстрации понимают всё, поэтому иконки используют в операционных системах, в интерфейсах, на сайтах и так далее.

Элементы блока шапки сайта представлены на рисунке 4.

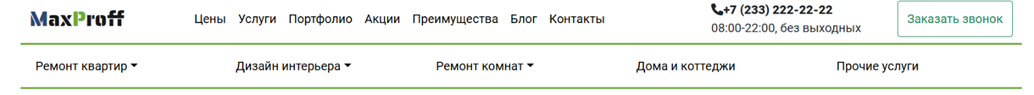


Рисунок 4 – Шапка сайта

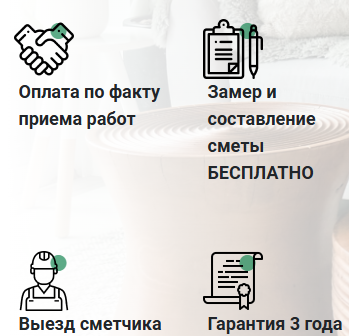
 Элементы блока «РЕМОНТ КВАРТИР В МОСКВЕ» представлены на рисунке 5.

Рисунок 5 –«РЕМОНТ КВАРТИР В МОСКВЕ»

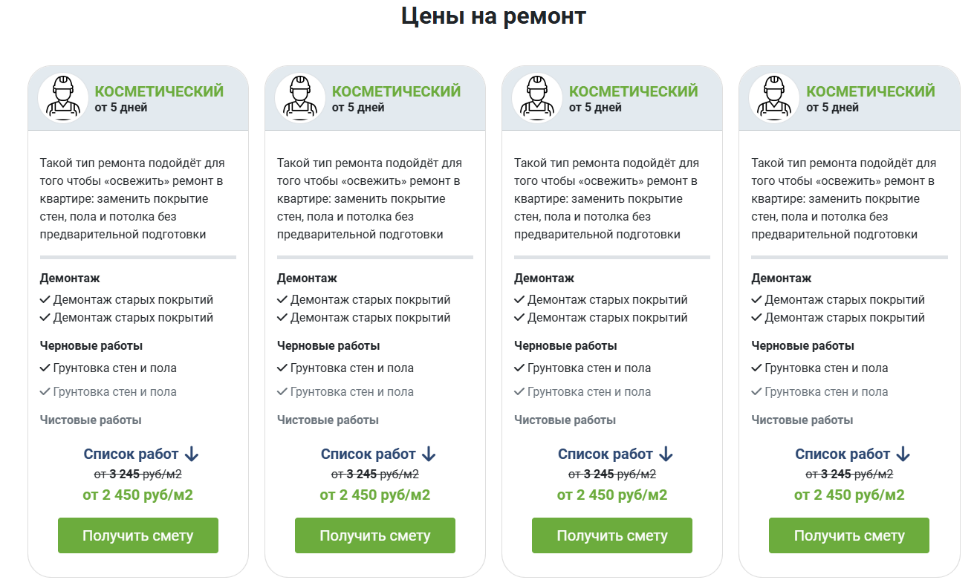
 Элементы блока «ЦЕНЫ НА РЕМОНТ» представлены на рисунке 6.

Рисунок 6 - «ЦЕНЫ НА РЕМОНТ»

**2.1.6 Кнопки и поля ввода**

Интерактивные элементы пользовательского интерфейса создают основу мощных и удобных цифровых продуктов. Внимательный подход к деталям помогает создать положительный опыт взаимодействия с продуктом в целом. Кнопка является одним из самых распространенных элементов любого интерфейса.

Кнопка «Заказать звонок» в шапке(рисунок 7).

Рисунок 7 - Кнопка «Заказать звонок»

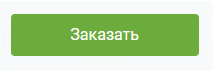
 Кнопка «Заказать», которая представлена на форме (рисунок 8).

Рисунок 8 - Кнопка «Заказать»

Радио-кнопки на сайте, а также поле для ввода, «ползунок», – (рисунок 9 и 10).

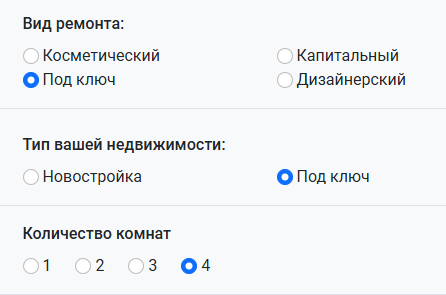


Рисунок 9 – Радиокнопки на форме



Рисунок 10 – Радиокнопки в блоке «Примеры наших работ»

 Вкладки (рисунок 11)

Рисунок 11- Вкладки

**2.1.7 Компоненты**

Кнопка «Наверх» (рисунок 12).

Рисунок 10 - Кнопка «Наверх»

Компонент Яндекс-карта (рисунок 13).

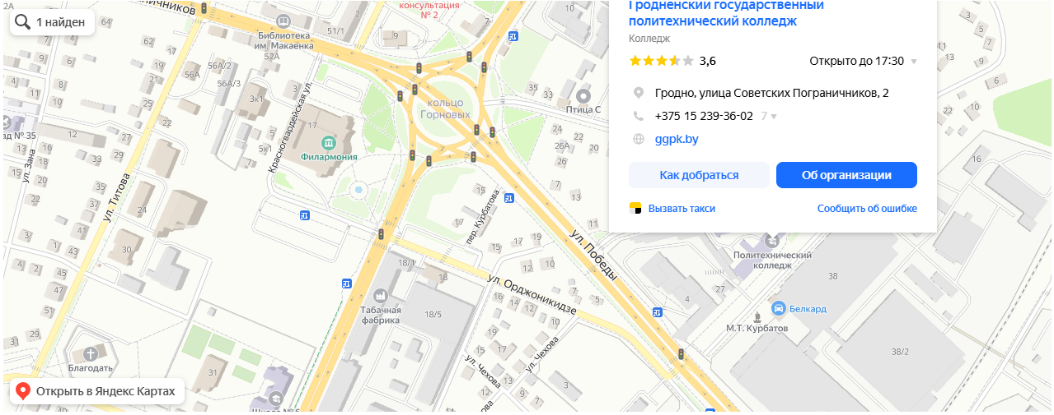


Рисунок 11 - Компонент Яндекс-карта

Компонент подсказок на форме (рисунок 14).

Рисунок 12 – Подсказка

Всплывающие сообщения alert (рисунок 15).

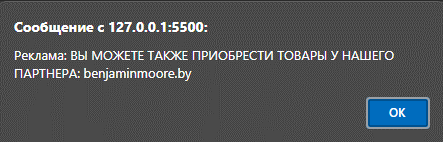


Рисунок-15 Всплывающее сообщение

**2.1.8 Пользовательские формы**

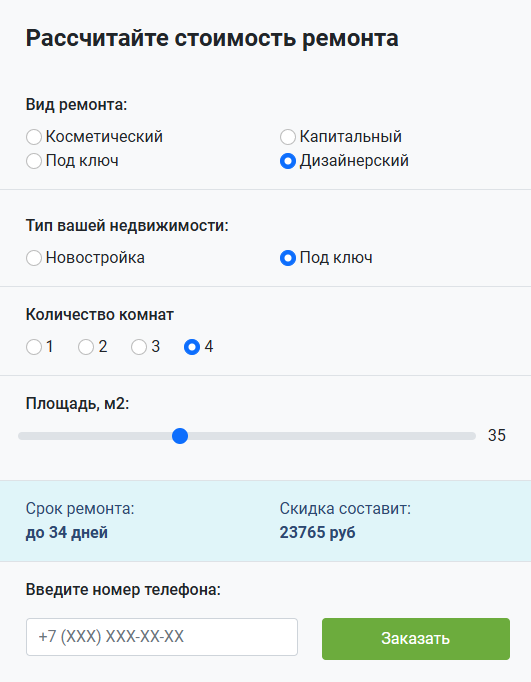
Пользовательской формой является форма «Рассчитайте стоимость ремонта» (рисунок 16).

Рисунок 16 - Пользовательская форма

**3 Реализация**

**3.1 Руководство программиста**

Общие требования:

Сайт должен быть разработан с помощью HTML, CSS и JS.

Сайт должен быть совместим с популярными браузерами и устройствами.

Сайт должен обеспечивать безопасность и защищенность данных клиентов.

Структура проекта:

Использовать HTML5 для создания структуры страницы.

Использовать семантические теги для улучшения доступности и поисковой оптимизации. Пример кода HTML:

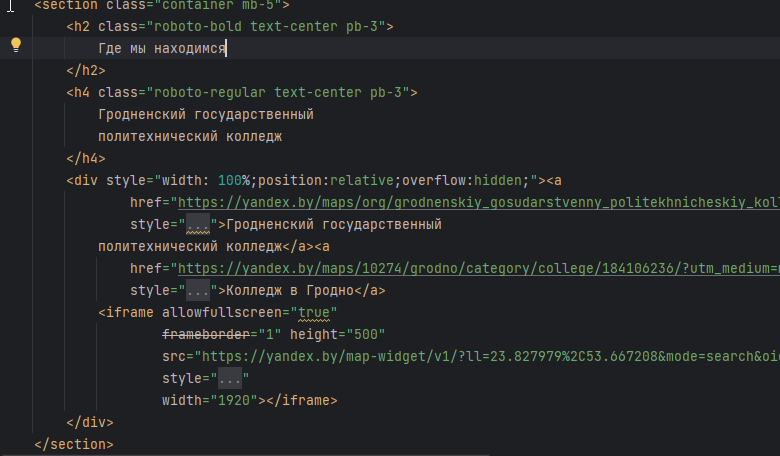


Рисунок 13 - Пример кода HTML

CSS

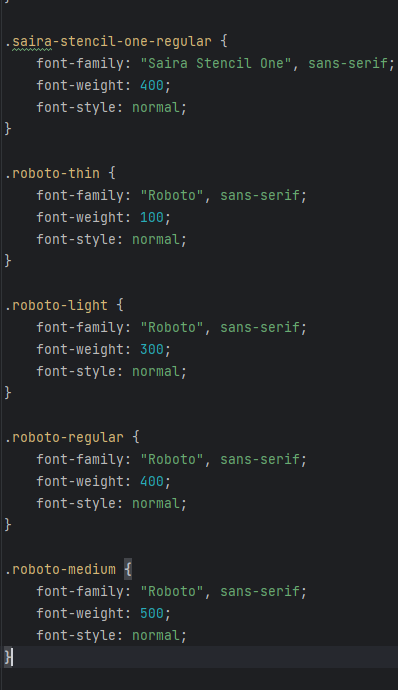
Использовать CSS3 для создания стилей и дизайна сайта. Пример кода CSS:

Рисунок 14 - пример кода CSS

JS

Использовать JavaScript для создания интерактивных элементов и функциональности сайта.

Использовать библиотеку jQuery и jQueryUI для упрощения написания JS-кода. Пример кода JavaScript:

Рисунок 19 - Примера кода JS

**3.2 Спецификация файлов проекта**

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 1 – Спецификация файлов проекта | | | |
| **Имя корневой папки** | **Имя папки** | **Имя файла** | **Назначение** |
| Практика по ВЕБ-ПРОГРАММИРОВАНИЮ | css | style.css | Файл с общими стилями страницы |
|  |  | font-awesome.css | Файл со стилями шрифтов и иконок |
|  |  | bootstrap.min.css | Файл со стилями bootstrap |
|  |  | swiper-bundle.min.css | Файл со стилями swiper |
| Практика по ВЕБ-ПРОГРАММИРОВАНИЮ | img/\* | \*.png | Картинки для проекта |
|  |  | \*.svg | Иконки для проекта |
| Практика по ВЕБ-ПРОГРАММИРОВАНИЮ | html | index.html | Файл разметки главной страницы сайта |
| Практика по ВЕБ-ПРОГРАММИРОВАНИЮ | js | script.js | Файл с общими скриптами страницы |
|  |  | jquery-ui.js | Файл со скриптами jquery-ui |
|  |  | jquery-3.6.0.min.js | Файл со скриптами jquery |
|  |  | bootstrap.bundle.min.js | Файл со скриптами bootstrap |
|  |  | swiper-bundle.min.js | Файл со скриптами swiper |
|  |  | popper.min.js | Файл со скриптами popper |
|  |  |  |  |

**4 Тестирование**

Тестирование является одним из самых важнейших этапов при создании программного продукта. Исходя из внедрения программы при определённом наборе тестов, нельзя сделать однозначных выводов о том, что программа будет исправно работать в любой момент использования. Опираясь на данный набор тестов, можно говорить о некоторой степени уверенности в правильности работы программного продукта.

При тестировании программного продукта, в первую очередь нужно обратить внимание на правильную работу страницы. Показывает ли текст, видны ли картинки, нет ли неравномерных отступов, все работают ли пункты меню. Нужно проверить верную работу всех подключённых скриптов для правильной работы сайта. Также необходимо проверить имеется ли адаптация под разные устройства.

Таким образом, основная часть ошибок и недоработок была выявлена и исправлена на этапе реализации проекта. После завершения этапа реализации, было проведено тщательное тестирование.

Программный продукт был протестирован разработчиком – Сенкевичем Виталием Андреевичем. Тестирование проводилось на персональном ноутбуке с операционной системой «Windows 11 Pro» с использованием браузера «Firefox», «Chrome», «Microsoft Edge». Расписание проведения и время, затраченное на тестирование, описано в таблице 2.

Таблица 2 – Отчёт о результатах тестирования функций сайта.

|  |  |  |  |
| --- | --- | --- | --- |
| **ФИО** | **Дата** | **Описание** | **Длительность, ч** |
| Соколовский Э.В | 19.06.2024 | Выполнение тест-кейсов | 3 |
| Соколовский Э.В | 19.06.2024 | Анализ выполнения тест-кейсов | 2 |
| Соколовский Э.В | 20.06.2024 | Повторение выполнения тест-кейсов | 1 |

Далее проводилась статистика по тестированию.

Выявленные по результатам тестирования ошибки представлены ниже (таблица 3).

Таблица 3 - Статистика по выявленным ошибкам

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Статус** | **Количество ошибок** | **Важность** | | | |
| **Критическая** | **Высокая** | **Средняя** | **Минимальная** |
| Проверено | 1 | 1 | 1 | 1 | 1 |
| Исправлено | 1 | 1 | 1 | 1 | 1 |
| Открыто заново | 0 | 0 | 0 | 0 | 0 |
| Найдено | 0 | 0 | 0 | 0 | 0 |
| Отклонено | 0 | 0 | 0 | 0 | 0 |

Таким образом, после проведения тестирования, были выявлены следующие критические ошибки:

- не происходит валидация полей формы.

Опираясь на вышеперечисленное, можно сделать вывод о стабильности работы программного продукта и его основного функционала. Найденные ошибки были устранены, соответственно, тестирование прошло успешно и программный продукт характеризуется как реализованный и готовый к работе.

**5 Применение**

Для того, чтобы открыть сайт необходимо перейти по ссылке:

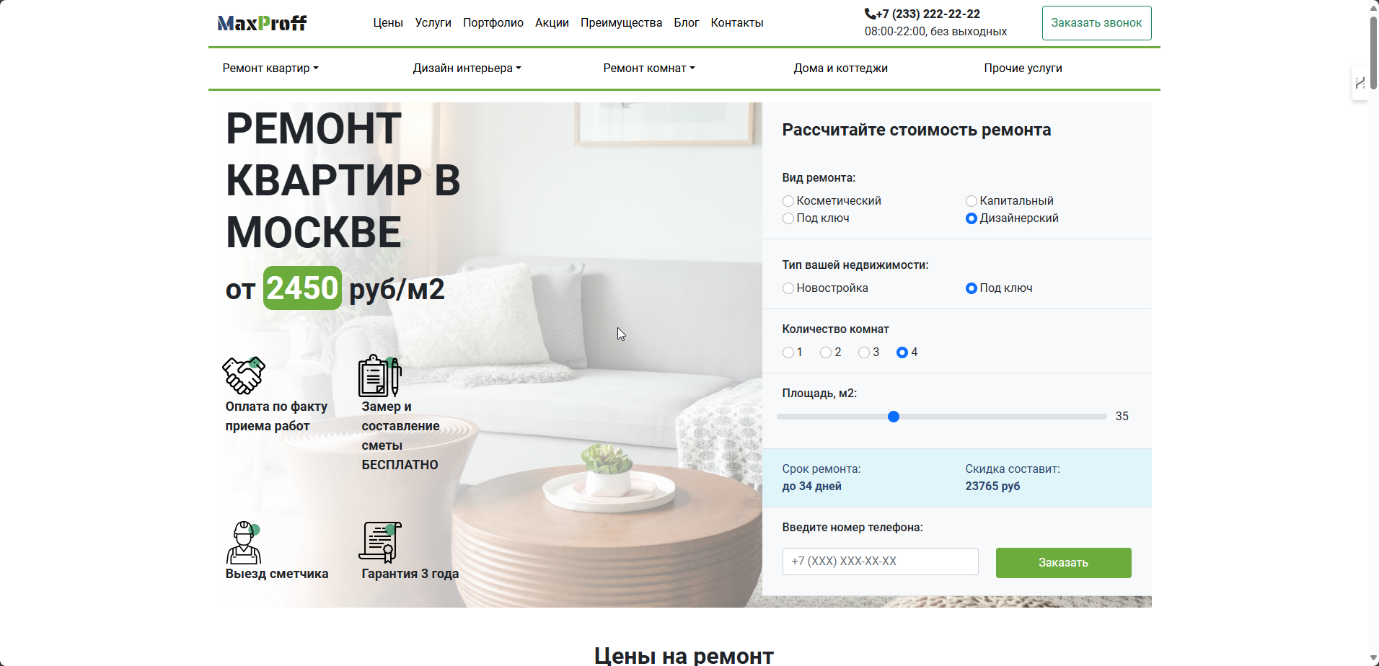
<https://github.com/Eximyx/WEB_PRACTICE>

Рисунок 20 – Главная страница сайта

**Заключение**

Данный проект представляет собой создание современного веб-ресурса, ориентированного на предоставление услуг по ремонту квартир в Москве. Используя передовые веб-технологии, такие как HTML, CSS и JavaScript, был разработан сайт, отвечающий всем требованиям современного пользователя.

Проект успешно достиг своих целей, создав функциональную и привлекательную платформу с интуитивно понятным интерфейсом и быстрой загрузкой страниц. Этот сайт не только предлагает удобный способ поиска и заказа ремонтных услуг, но также помогает пользователям подобрать идеальные решения для улучшения их жилого пространства.

Ожидается, что данный ресурс будет востребован среди целевой аудитории и будет способствовать улучшению качества их жилья. В дальнейшем возможны доработки и улучшения сайта на основе отзывов пользователей и технологических новшеств, что обеспечит его соответствие самым высоким стандартам качества и удобства использования.

.

**Список использованных источников**

1 Сайт с информацией о работе с html [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://htmlacademy.ru/> – Дата доступа: 14.06.2024

2 Знакомство с CSS [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://htmlacademy.ru/courses/41> – Дата доступа: 15.06.2024.

3 Разработка с помощью JS [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://htmlacademy.ru/intensive/javascript> – Дата доступа: 18.06.2024.

**Приложение А**

Модульная сетка

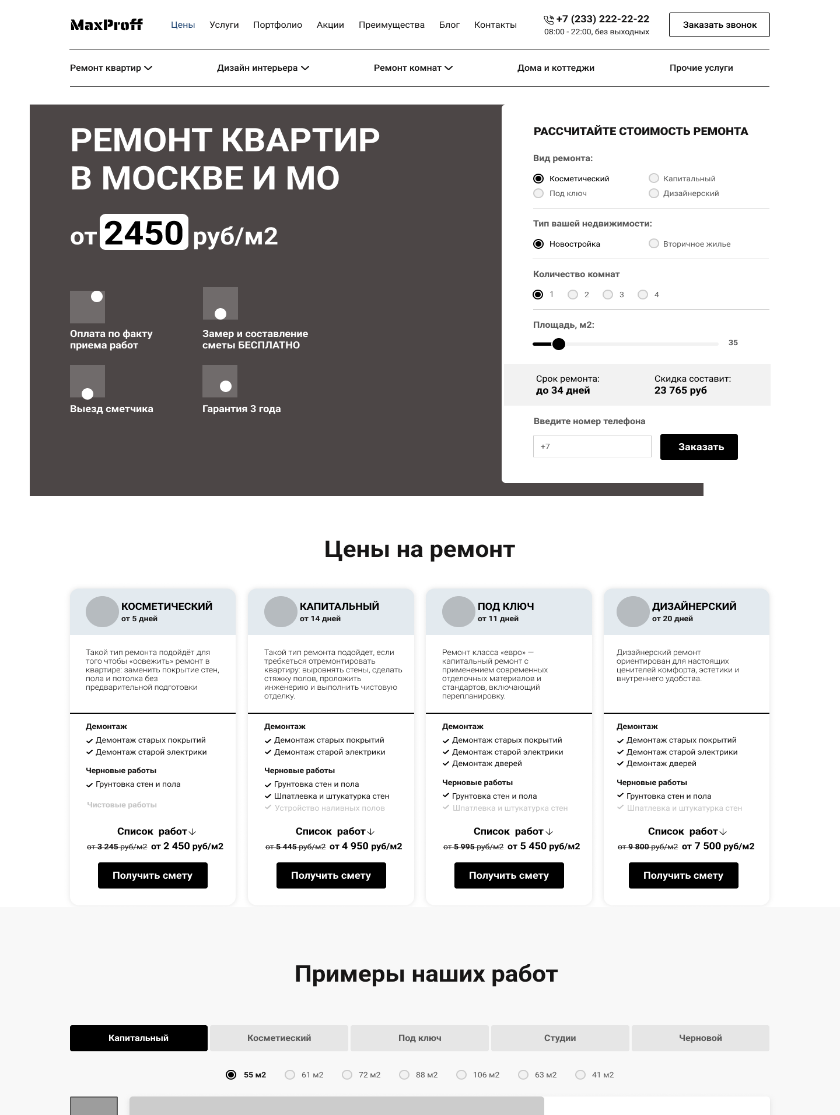
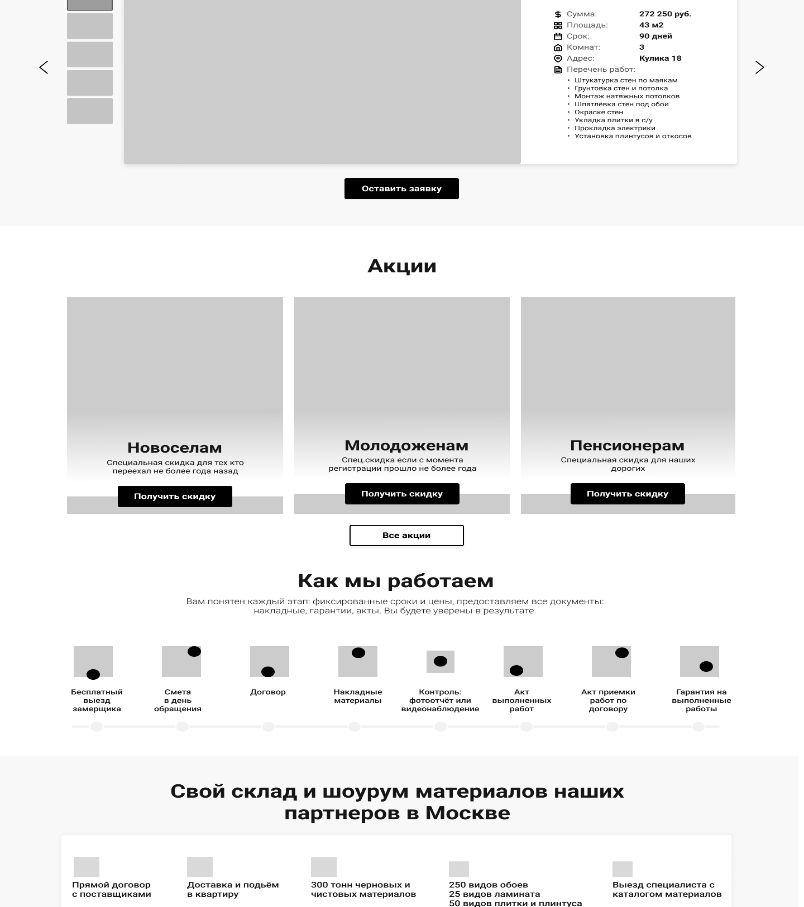


Рисунок А.1 – ПК версия (Часть 1)

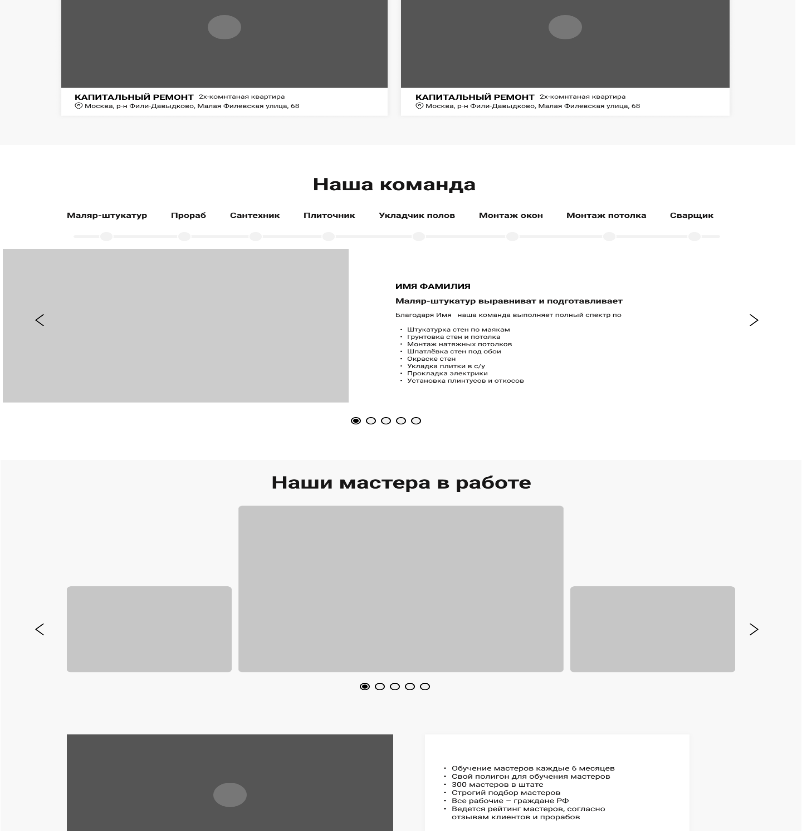


Рисунок А.1 – ПК-версия (Часть 2)

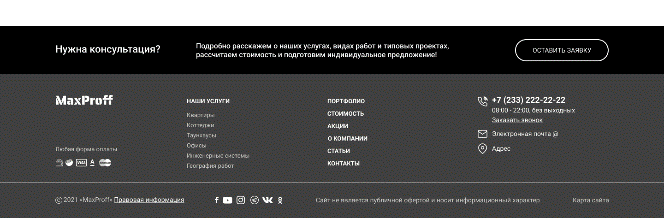
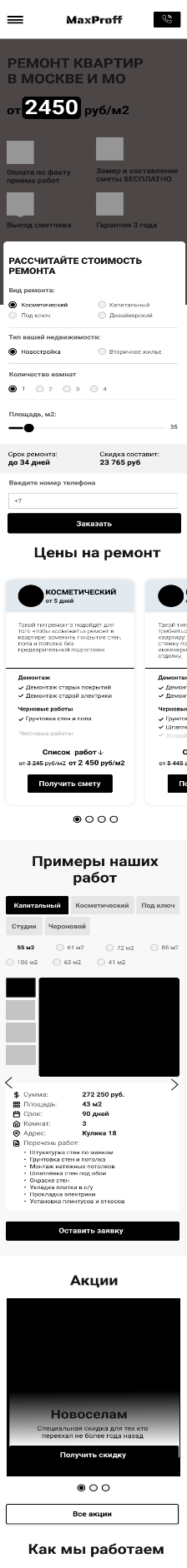
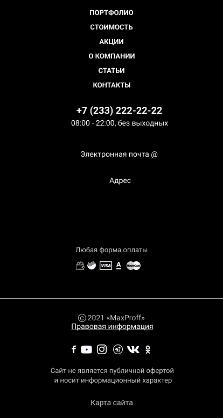
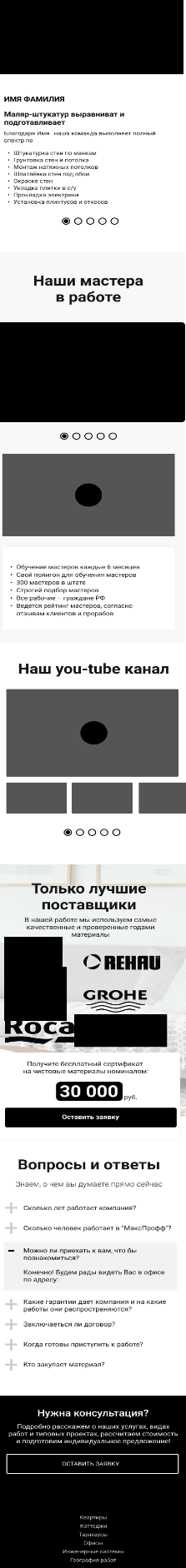
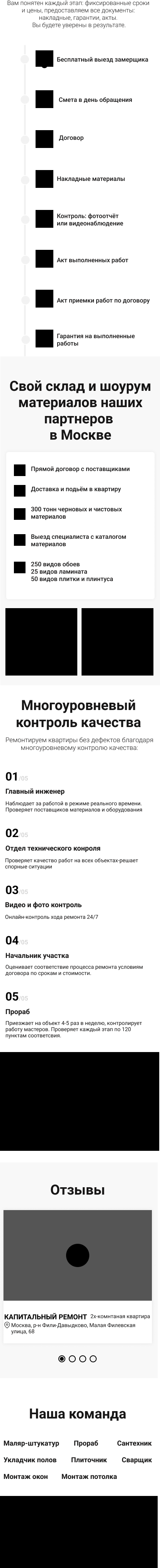


Рисунок А.1 – ПК-версия (Часть 3)



А.3 – Мобильная версия