автономное профессиональное образовательное учреждение

Вологодской области

«Вологодский колледж связи и информационных технологий»

КУРСОВОЙ ПРОЕКТ

РАЗРАБОТКА ИНТЕРНЕТ-СЕРВИСА ДЛЯ МЕДИЦИНСКИХ УЧРЕЖДЕНИЙ

Группа: ИСП-421ис, специальность 09.02.07 «Информационные системы и программирование»

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Курнышев «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Нормоконтролер: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.М. Ананьин «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.М. Ананьин «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Вологда,

2024 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc184180392)

[РАЗДЕЛ 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc184180393)

[1.1. Дизайн и верстка интернет-сервиса 4](#_Toc184180394)

[1.2. UI/UX (Пользовательский интерфейс и пользовательский опыт) 5](#_Toc184180395)

[1.3. Анализ аналогов 6](#_Toc184180396)

[1.4. Выбор инструментов и технологий 8](#_Toc184180397)

[1.4.1. Язык веб-разметки 8](#_Toc184180398)

[1.4.2. Таблица каскадных стилей 9](#_Toc184180399)

[1.4.3. JavaScript 10](#_Toc184180400)

[РАЗДЕЛ 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 11](#_Toc184180401)

[2.1. Цель и задачи проекта 11](#_Toc184180402)

[2.2. Создание дизайн-макета в Figma 11](#_Toc184180403)

[2.3. Вёрстка страниц 17](#_Toc184180404)

[2.3.1. HTML-разметка 17](#_Toc184180405)

[2.3.2. CSS-стилизация 24](#_Toc184180406)

[2.3.3. Адаптивность верстки 28](#_Toc184180407)

[2.3.4. Добавление интерактивности с JS 31](#_Toc184180408)

[2.3.5. Реализация графического интерфейса 33](#_Toc184180409)

[2.3.6. Тестирование проекта 33](#_Toc184180410)

[РАЗДЕЛ 3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ 34](#_Toc184180411)

[3.1. Затраты на электроэнергию 35](#_Toc184180412)

[3.2. Заработная плата разработчика 35](#_Toc184180413)

[3.3. Отчисления от заработной платы 35](#_Toc184180414)

[3.4. Амортизация основных средств 36](#_Toc184180415)

[3.5. Прочие расходы 36](#_Toc184180416)

[3.6. Общая сумма затрат 36](#_Toc184180417)

[3.7. Прибыль 36](#_Toc184180418)

[3.8. Цена продукта 36](#_Toc184180419)

[ЗАКЛЮЧЕНИЕ 37](#_Toc184180420)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 38](#_Toc184180421)

# **ВВЕДЕНИЕ**

Современная медицина активно использует информационные технологии для повышения эффективности работы медицинских учреждений, улучшения качества обслуживания пациентов и оптимизации внутренних процессов. Важнейшую роль в этом играет развитие интернет-сервисов, которые обеспечивают удобный доступ к медицинской информации, упрощают взаимодействие между врачами и пациентами, а также автоматизируют рутинные задачи. Создание таких сервисов становится неотъемлемой частью цифровизации здравоохранения, направленной на удовлетворение потребностей современного общества.

Актуальность разработки интернет-сервиса для медицинских учреждений обусловлена несколькими факторами. Во-первых, существует повышенная потребность в цифровых решениях, которые обеспечивают доступ к информации в режиме реального времени, особенно в условиях увеличения объемов данных и необходимости оперативного их анализа. Во-вторых, в последние годы наблюдается значительный рост числа пользователей, предпочитающих дистанционные формы взаимодействия с медицинскими учреждениями, включая запись на прием, получение рекомендаций и консультирование. Это требует создания надежных, удобных и безопасных интернет-сервисов, которые соответствуют ожиданиям пациентов и стандартам качества медицинских услуг.

Целью данной работы является разработка веб-сайта для медицинской информационной системы MIS, предназначенного для представления её возможностей, преимуществ, отзывов клиентов и примеров интеграций. Сайт должен включать главную страницу с информацией о системе, а также страницы с тарифами, входом и регистрацией, обеспечивая удобный доступ для пользователей, которые могут подключиться к сервису с бесплатным тестовым периодом.

Для достижения этой цели в рамках работы поставлены следующие задачи: • Провести анализ существующих веб-сайтов для медицинских учреждений, выявить их преимущества и недостатки.

• Определить требования к разрабатываемому сайту, включая функциональные и технические характеристики.

• Разработать архитектуру и интерфейс сайта, который будет удобным и интуитивно понятным для пользователей.

• Реализовать основные страницы сайта, включая главную страницу с информацией о системе, страницы с тарифами, а также страницы входа и регистрации, обеспечив их адаптивность и функциональность.

# **РАЗДЕЛ 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

Разработка интернет-сервисов для медицинских учреждений — это важная часть цифровизации здравоохранения, способствующая улучшению качества предоставляемых медицинских услуг, удобству пациентов и повышению эффективности работы медперсонала. Разработка таких сервисов требует учета множества факторов, включая требования к безопасности данных, соблюдение стандартов в области медицины и законов о защите персональной информации, а также оптимизацию пользовательского интерфейса (UI) и опыта (UX) для разных категорий пользователей.

В данной теоретической части будет рассмотрен ряд аспектов, связанных с проектированием, версткой и пользовательским интерфейсом/опытом (UI/UX) для интернет-сервисов медицинских учреждений, а также выбор инструментов и технологий для разработки.

## **1.1. Дизайн и верстка интернет-сервиса**

Дизайн интерфейса интернет-сервиса — это комплексная задача, включающая как визуальные элементы (графика, шрифты, цвета), так и архитектуру взаимодействия пользователя с системой. В медицинских приложениях и сервисах дизайн должен быть продуманным и интуитивно понятным для пользователей с разным уровнем технической подготовки. Важнейшими требованиями к дизайну медицинских интернет-сервисов являются:

* Простота и понятность: медучреждения обслуживают пользователей разного возраста, уровня технической грамотности, включая пожилых людей и людей с ограниченными возможностями. Поэтому интерфейс должен быть минималистичным, интуитивно понятным и четким.
* Доступность информации: все медицинские данные и рекомендации должны быть представлены доступным языком, без сложных медицинских терминов, а также с учетом особенностей восприятия.
* Мобильность: пользователи медицинских сервисов часто используют мобильные устройства. Поэтому важно, чтобы дизайн и верстка были адаптивными для различных экранов и устройств (мобильные телефоны, планшеты и десктопы).
* Цветовая палитра: в медицинских интерфейсах часто используется спокойная и нейтральная цветовая палитра для создания доверительной атмосферы. Зеленый, синий, белый цвета ассоциируются с безопасностью и спокойствием.

## **1.2. UI/UX (Пользовательский интерфейс и пользовательский опыт)**

UI (User Interface) — это непосредственно то, с чем взаимодействует пользователь (кнопки, меню, окна), а UX (User Experience) — это общее восприятие и ощущение от использования сервиса. Правильно разработанный интерфейс и опыт пользователя обеспечивают не только функциональность, но и удовлетворение от работы с сервисом.

Особенности UI/UX в медицинских сервисах

* Навигация: структура сервиса должна быть простой и логичной. Важно, чтобы пользователь мог быстро найти нужную информацию.
* Формы и кнопки: форма записи на прием, запрашиваемая информация должна быть компактной и понятной. Для этого используются большие, четкие кнопки и поля ввода с явными подсказками и ошибками.
* Инклюзивность: необходимо учитывать различные физические и когнитивные особенности пользователей. Это включает в себя увеличение шрифта, цветовые контрасты для людей с нарушением зрения, голосовые подсказки и так далее.
* Кнопка «Гамбургер» для мобильных устройств: необходимо для оптимизации пространства на экране и обеспечивания более удобной навигации.

Методология и подходы к UI/UX

Для разработки интерфейса медицинского сервиса можно использовать несколько подходов:

Agile — это набор принципов и методологий, направленных на гибкую и итерационную разработку программного обеспечения, которая позволяет быстро адаптироваться к изменениям и эффективно реагировать на запросы пользователей. В контексте разработки интернет-сервисов для медицинских учреждений Agile помогает ускорить процесс создания и улучшения сервисов, обеспечивая удобный пользовательский интерфейс и интерфейс пользователя, который может адаптироваться к изменениям в требованиях и отзывах. Основные принципы и элементы Agile:

1. Итеративный процесс: в отличие от традиционного подхода (например, водопадной модели), где весь проект разрабатывается поэтапно от начала до конца, Agile делит процесс на небольшие итерации, называемые спринтами. Каждый спринт обычно длится от 1 до 4 недель и включает в себя полный цикл разработки — от планирования до тестирования и выпуска рабочей версии функционала.
2. Ранний и регулярный выпуск работающих версий: Agile ориентирован на регулярный выпуск рабочей версии продукта с каждым спринтом. Даже если проект еще не завершен, пользователи могут тестировать части функционала. Это особенно важно в контексте медицинских приложений, где ранняя версия может позволить собрать данные о потребностях пользователей и улучшить сервис.
3. Гибкость и адаптивность к изменениям: Одна из главных особенностей Agile — это способность оперативно вносить изменения в проект в ответ на обратную связь от пользователей, изменения в требованиях или новые идеи. В медицине это особенно актуально, потому что требования к функциональности могут изменяться в зависимости от нормативных актов, пожеланий пользователей или особенностей работы учреждения.
4. Интенсивное сотрудничество между командой разработки и заказчиком: В Agile особое внимание уделяется взаимодействию команды с заказчиком или конечным пользователем. Разработчики, дизайнеры и тестировщики работают в тесном сотрудничестве с представителями медицинских учреждений, чтобы лучше понять их потребности и быстро устранять возникающие проблемы.
5. Тестирование на каждом этапе: В Agile тестирование — это не отдельный этап, а постоянный процесс, который идет параллельно с разработкой. Каждая версия продукта тестируется на каждом спринте, что помогает выявлять ошибки и улучшать функциональность на ранних стадиях. В случае медицинских сервисов это особенно важно, так как безопасность данных, точность расчетов и правильность взаимодействия с пользователем должны быть проверены на всех этапах.
6. Пользовательский фокус и непрерывное улучшение: В Agile важное место отводится регулярным обзорам продукта с конечными пользователями и их отзывам. Этот процесс помогает определить, что именно необходимо улучшить в сервисе, чтобы сделать его более удобным и функциональным.
7. Минимизация документации: в отличие от традиционных подходов, где создается обширная документация на всех этапах, в Agile внимание сосредоточено на минимальной документации, которая необходима для выполнения работы.

## **1.3. Анализ аналогов**

Анализ аналогичных медицинских интернет-сервисов представленный в таблице 1, позволяет выявить лучшие практики и избежать распространенных ошибок.

Таблица 1

Общий сравнительный анализ

| Критерии | https://n3health.ru/ | https://archimed.pro/ | https://lechimbr.ru/ |
| --- | --- | --- | --- |
| Общий стиль дизайна | Чистый и современный, с акцентом на медицинские услуги | Простой, деловой, с фокусом на ветеринарию | Современный, медицинский стиль с акцентом на функциональность |
| Типографика | Стандартная медицинская типографика, легко читаемая | Простой шрифт, функциональный, читаемый | Современные шрифты, с хорошей читаемостью на всех устройствах |
| Навигация | Простое меню с главными разделами, быстрое переключение | Основное меню с выделением разделов и услуг | Простое и интуитивно понятное меню |
| Адаптивность (мобильная версия) | Хорошая адаптивность, элементы подстраиваются под экран мобильных устройств | Адаптивная верстка, но иногда перегрузка на маленьких экранах | Полностью адаптивный дизайн с фокусом на мобильные устройства, улучшенная навигация |
| Отзывчивость интерфейса | Блоки и элементы адаптируются под различные устройства, но иногда текст слишком мелкий | Хорошая отзывчивость на большинстве устройств, но иногда сложность в мобильной версии | Отличная отзывчивость и адаптация на всех устройствах, включая планшеты и мобильные устройства |
| Тестирование на различных устройствах | Прошло тестирование на мобильных устройствах, но есть некоторые баги | Хорошо работает на мобильных и планшетах | Тестирование на всех устройствах с учетом различных браузеров и экранов, улучшенная совместимость |
| Скорость загрузки | Средняя, иногда заметная задержка при загрузке элементов | Хорошая скорость загрузки на всех устройствах | Высокая скорость загрузки благодаря оптимизированной верстке и сжатию изображений |

Важные моменты, которые можно выделить из анализа:

* Минималистичный стиль: Дизайн необходимо делать в минималистичном стиле, так как он позволяет сосредоточить внимание пользователей на ключевой информации, не перегружая страницу лишними элементами. Это улучшает восприятие контента и повышает удобство навигации.
* Простое и логичное меню: Главное меню должно быть четким и логичным, чтобы пользователи могли быстро найти необходимые разделы, такие как услуги, запись на прием, контактные данные. Простота в навигации позволяет удерживать внимание и снижать вероятность того, что посетители уйдут, не найдя нужную информацию.
* Адаптивный дизайн: Сайт должен корректно отображаться на различных устройствах — мобильных телефонах, планшетах и десктопах. Это крайне важно, поскольку все больше людей используют мобильные устройства для доступа к медицинской информации.
* Оптимизация под мобильные устройства: Мобильная версия должна быть с легким доступом ко всем основным разделам, быстрой загрузкой и удобными кнопками. Убедитесь, что шрифты и элементы интерфейса не слишком мелкие и удобны для восприятия на маленьких экранах.

## **1.4. Выбор инструментов и технологий**

Для разработки медицинского интернет-сервиса важно выбрать инструменты и технологии, которые обеспечат удобство, безопасность и масштабируемость. Технологии, используемые при разработке интернет-сервиса для медицинских учреждений.

HTML5, CSS3 и JavaScript — это три ключевых технологии, которые используются для разработки современных веб-сайтов и интернет-сервисов, включая медицинские порталы и приложения. Рассмотрим их более подробно, чтобы понять, как они работают и как каждый из них влияет на создание веб-интерфейса.

### **1.4.1. Язык веб-разметки**

HTML5 (HyperText Markup Language 5) — это современный стандарт для разметки веб-страниц. Он определяет структуру и содержимое страницы, обеспечивая основу для всех элементов, которые отображаются в браузере. Основные возможности HTML5:

* Семантические теги: HTML5 привнес новые теги, которые помогают структуировать страницы и делают код более понятным. Например:
* <header>, <footer>, <article>, <section> — эти теги помогают разделять страницу на логические блоки, что улучшает как восприятие кода, так и SEO-оптимизацию.
* <nav> — для навигационных панелей.
* <figure> и <figcaption> — для удобной работы с изображениями и их подписями.
* Поддержка мультимедиа: HTML5 значительно улучшил работу с аудио и видео. Теги <audio> и <video> позволяют вставлять медиафайлы без использования внешних плагинов.
* Формы и ввод данных: HTML5 добавил новые типы форм (например, email, tel, date), которые позволяют упростить сбор данных, например, для записи на прием в медицинское учреждение.

HTML5 играет роль каркаса веб-страницы, на основе которого строится вся остальная функциональность и взаимодействие с пользователем.

### **1.4.2. Таблица каскадных стилей**

CSS3 (Cascading Style Sheets 3) — это стиль для оформления HTML-страниц. С помощью CSS можно управлять внешним видом и расположением элементов на странице. CSS3 добавляет новые возможности для создания красивых, адаптивных и динамичных интерфейсов. Основные возможности CSS3:

* Адаптивный дизайн (Responsive Design): CSS3 включает в себя медиа-запросы (@media), которые позволяют создавать сайты, адаптирующиеся под различные размеры экранов (мобильные устройства, планшеты, десктопы). В медицинских сервисах это особенно важно, так как многие пользователи будут заходить на портал с мобильных устройств.
* Например, для изменения размера и положения элементов на маленьких экранах (рисунок 1) можно использовать медиа-запросы:



Рисунок 1. Пример использования медиа-запросов

Flexbox и Grid Layout: Эти два новых метода в CSS3 позволяют более гибко и удобно управлять расположением элементов на странице. Flexbox (рисунок 2) помогает выравнивать элементы по горизонтали и вертикали, а Grid (рисунок 3) позволяет создавать сложные макеты с несколькими строками и колонками, что идеально подходит для создания адаптивных и многозадачных медицинских сервисов.

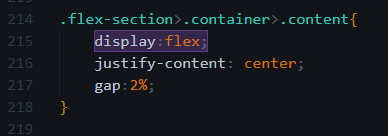


Рисунок 2. Пример использования Flexbox

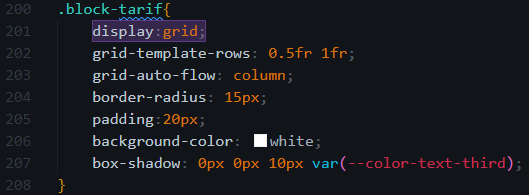


Рисунок 3. Пример использования Grid

CSS3 отвечает за визуальную составляющую веб-сайта, создавая привлекательный и удобный интерфейс.

### **1.4.3. JavaScript**

JavaScript — это язык программирования, который позволяет делать веб-страницы интерактивными и динамичными. С помощью JavaScript можно реализовать все основные функциональные элементы на сайте, такие как формы, динамическое обновление данных и взаимодействие с пользователем.

Основные возможности JavaScript:

Манипуляция DOM (Document Object Model): JavaScript позволяет изменять структуру HTML-страницы в реальном времени. Например, можно добавить или удалить элементы, изменить текст или атрибуты.

Обработка событий: JavaScript может отслеживать действия пользователя (клики, прокрутки, ввод данных) и выполнять действия в ответ на эти события.

JavaScript добавляет динамичность и интерактивность к сайту, что особенно важно для создания удобных и функциональных медицинских сервисов, таких как система записи на прием, персонализированные рекомендации или отображение статистики.

# **РАЗДЕЛ 2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

## **2.1. Цель и задачи проекта**

Целью данного проекта является создание интернет-сервиса для медицинских учреждений с использованием только HTML, CSS, JS и Figma для дизайна. Сервис должен включать в себя основные функциональные блоки, такие как: главная страница, тарифы, вход, регистрация. Все страницы должны быть адаптивными и удобными для пользователей, включая мобильные устройства.

## **2.2. Создание дизайн-макета в Figma**

Процесс разработки начинается с проектирования интерфейса в Figma. Это позволяет создать визуальное представление сайта, которое затем будет преобразовано в реальную верстку. В Figma создаются макеты всех страниц (рисунок 4), учитывая требования к дизайну и удобству пользователя.

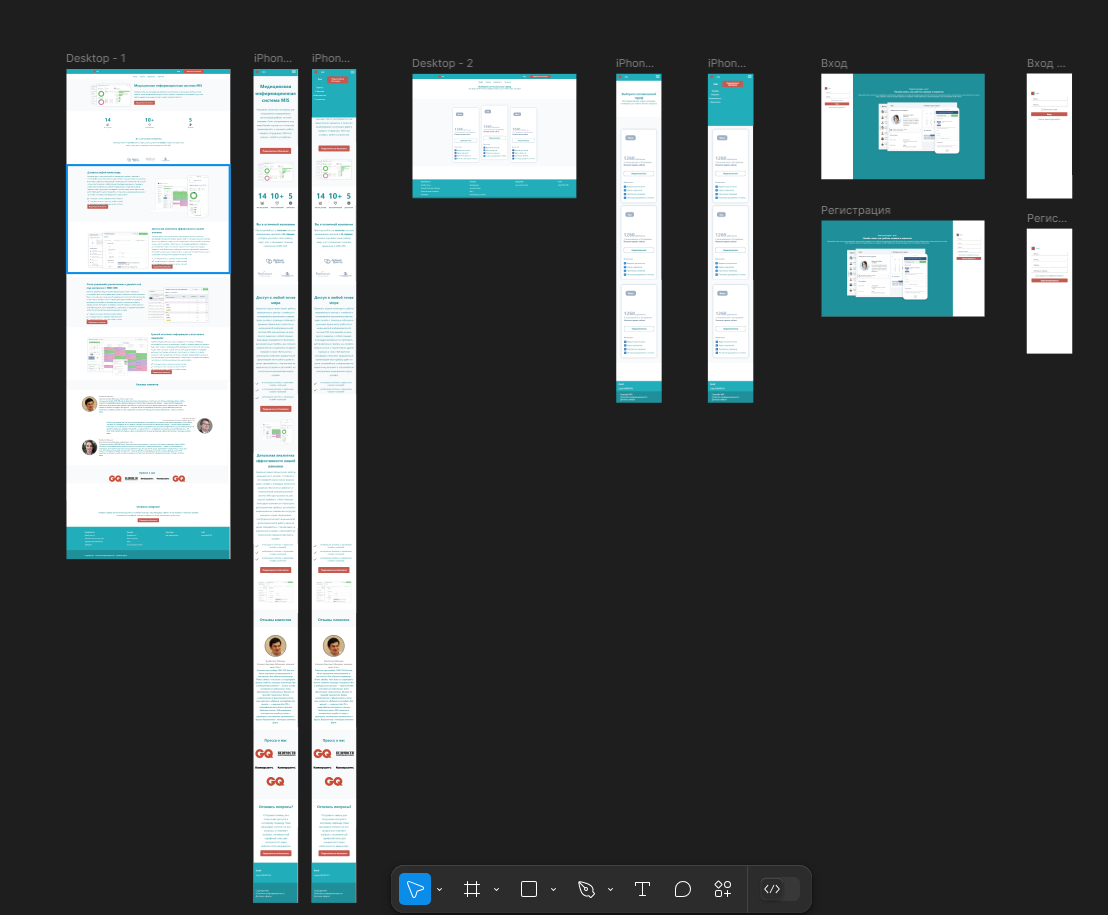


Рисунок 4. Полный вид дизайн-макета Figma

На главной странице (рисунок 5) размещаются ключевые элементы: логотип, меню навигации, кнопки для входа и регистрации. Важно, чтобы эта страница была простой и интуитивно понятной.

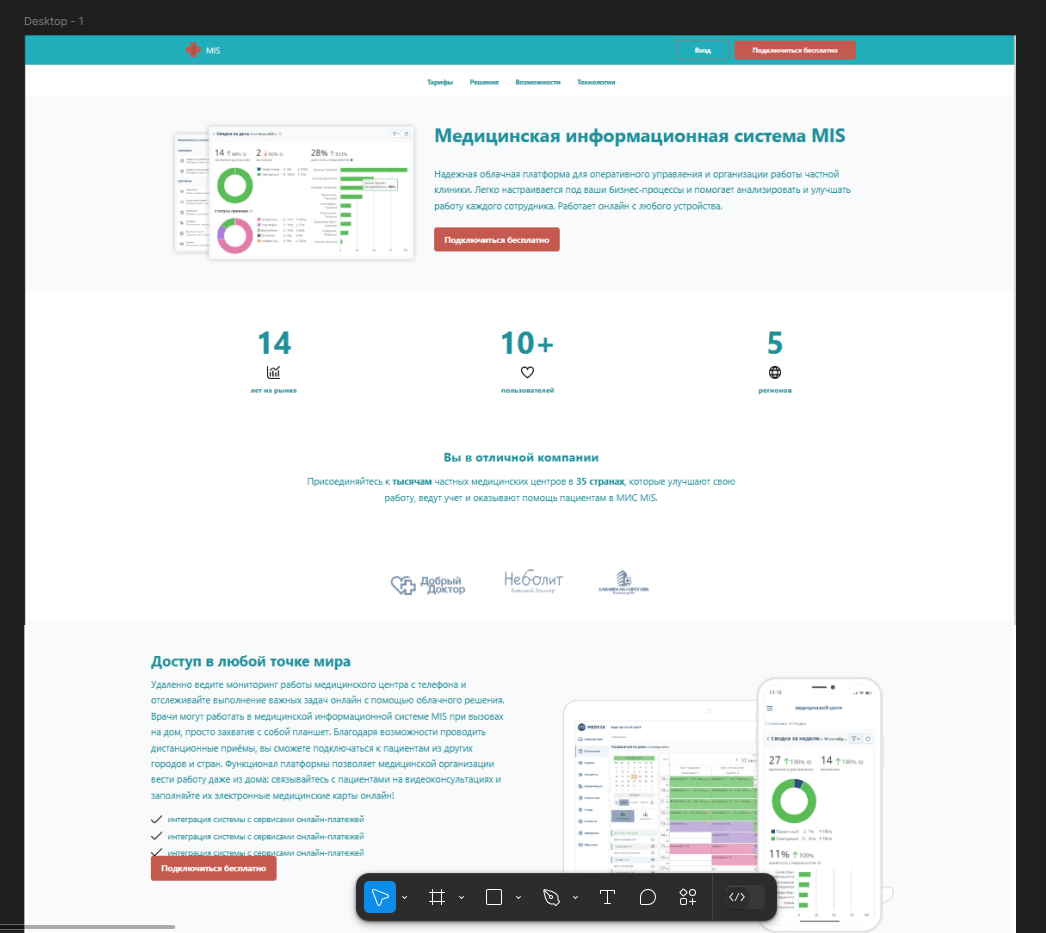


Рисунок 5. Фрагмент дизайн-макета главной страницы

Страница тарифы (рисунок 6) включает блоки, где показаны актуальные тарифы на услуги медицинской информационной системы «MIS». Эта страница должна быть минималистичной и не перегруженной.

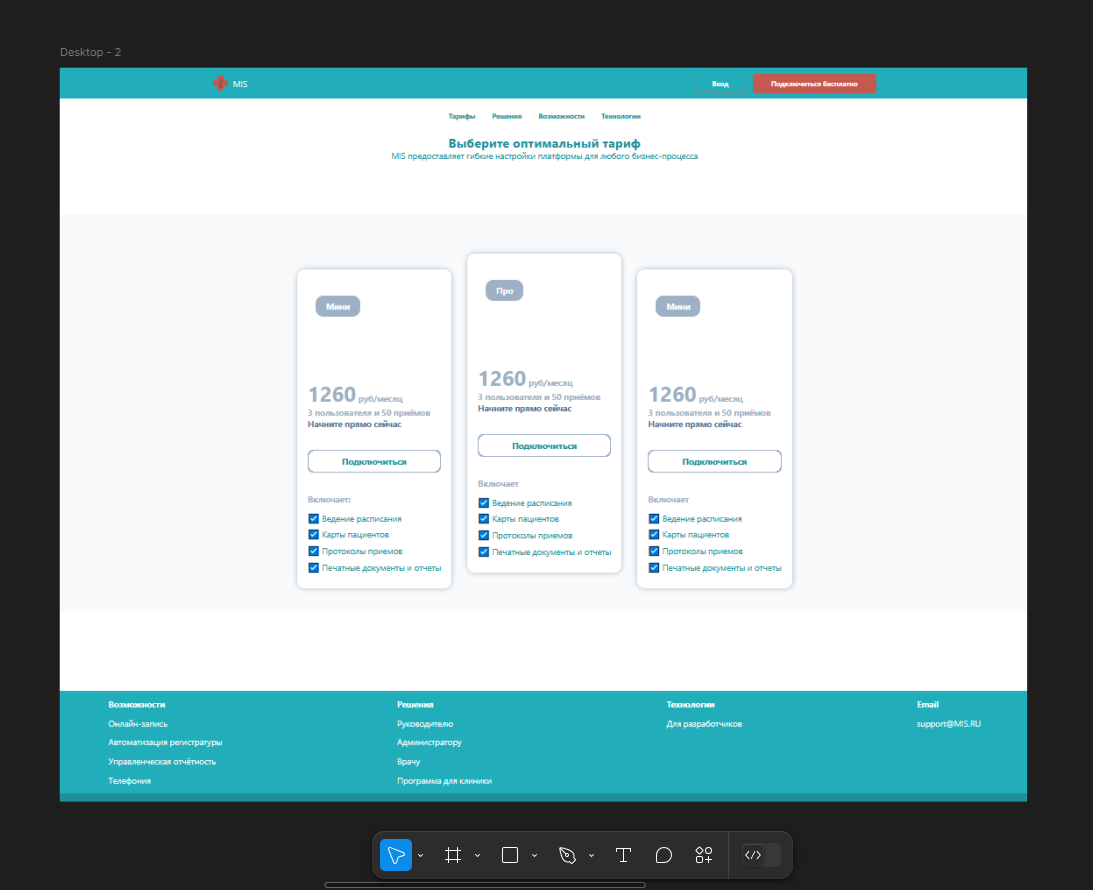


Рисунок 6. Дизайн-макет страницы тарифы

На странице регистрации (рисунок 7) должны быть поля для ввода логина, почты, пароля и его повторение, также должно присутствовать согласие на обработку данных, и кнопка зарегистрироваться.

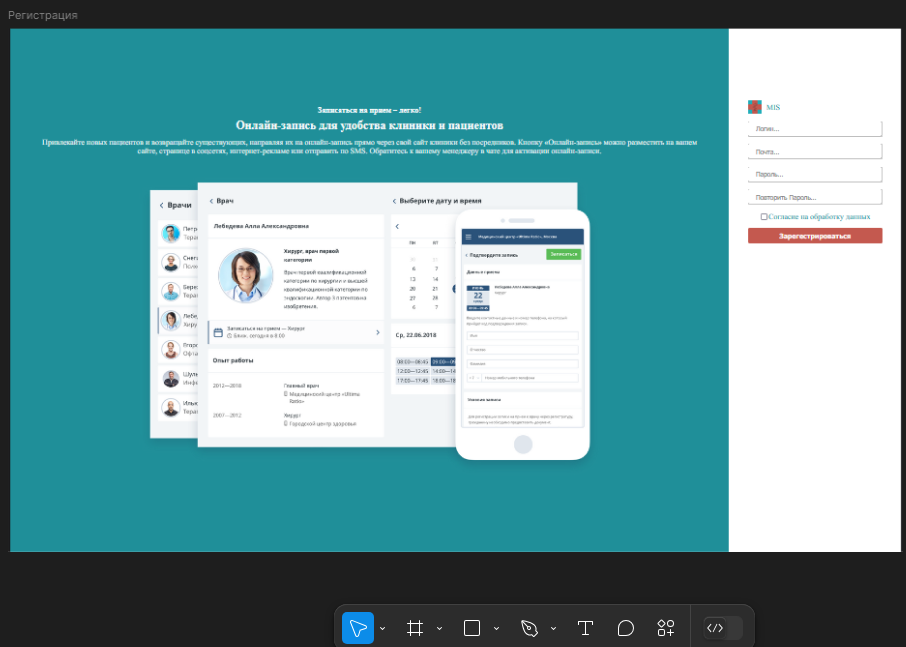


Рисунок 7. Дизайн-макет страницы регистрации

На странице входа (рисунок 8) должны быть поля для ввода логина, пароля, кнопка для входа.

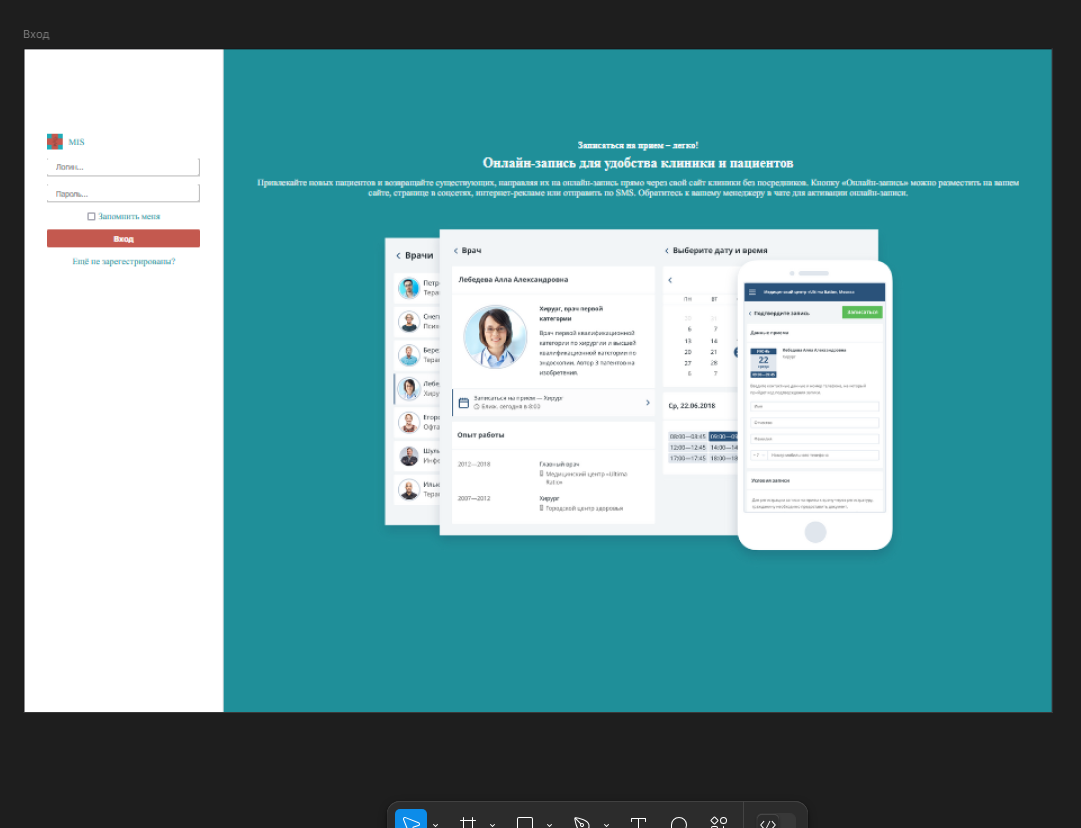


Рисунок 8. Дизайн-макет страницы вход

Адаптация сайта под мобильные устройства — важная часть разработки, особенно для медицинского интернет-сервиса. Большинство пользователей сегодня ищут медицинские услуги и записываются на прием именно с мобильных устройств, поэтому мобильная версия сайта должна быть удобной, функциональной и быстрой.

Для главной страницы (рисунок 9) важно адаптировать логотип и навигационное меню, которое в последствии заменится на «Гамбургер» (рисунок 10), уменьшить размеры крупных изображений, а также обеспечить удобный доступ к ключевым кнопкам и читаемость текста.

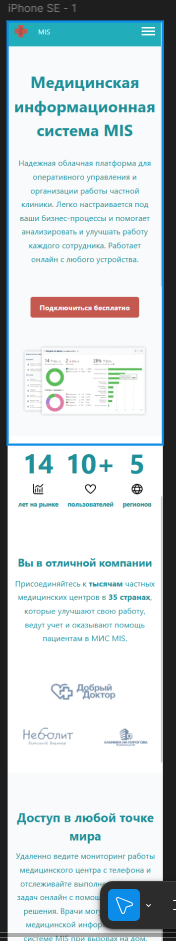


Рисунок 9. Дизайн-макет мобильной версии главной страницы

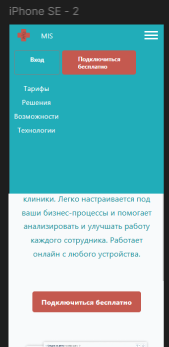


Рисунок 10. Дизайн-макет работы «Гамбургер» на главной странице

На странице входа (рисунок 11) необходимо централизовать форму ввода, увеличить кнопки для удобства касания.

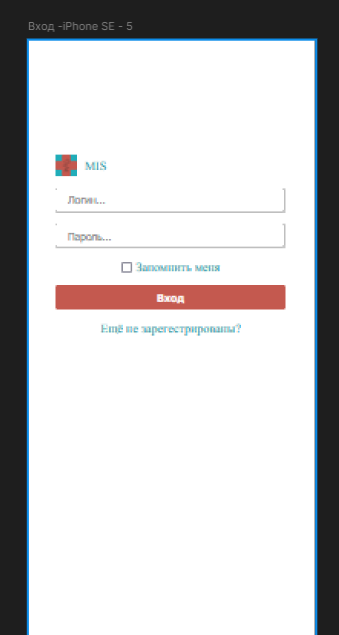


Рисунок 11. Дизайн-макет мобильной версии страницы входа

Страницу регистрации (рисунок 11) следует сделать максимально простой, адаптировав поля ввода для мобильного экрана и добавив возможность автоматической проверки данных.

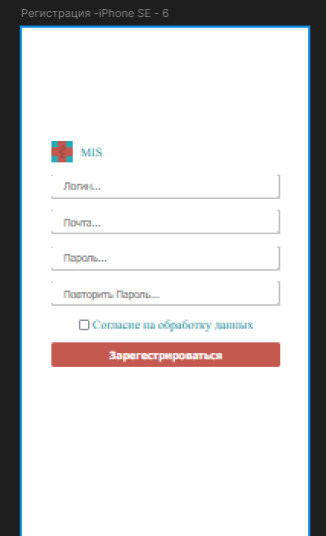


Рисунок 11. Дизайн-макет мобильной версии страницы регистрации

Для страницы тарифов (рисунок 12) нужно перестроить карточки тарифов в вертикальный вид.

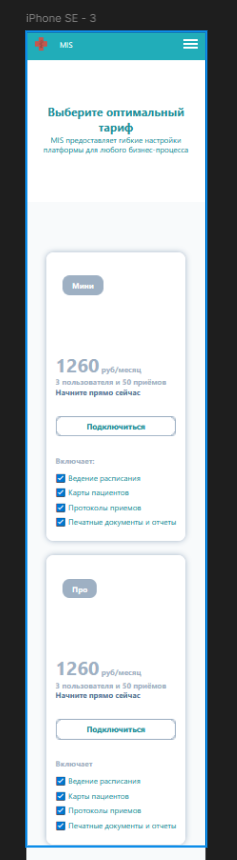


Рисунок 12. Дизайн-макет мобильной версии страницы тарифы

## **2.3. Вёрстка страниц**

После создания макетов начинается процесс верстки. Этот этап включает в себя создание HTML-структуры, оформление с помощью CSS и использование JavaScript для интерактивных элементов, таких как «Гамбургер»-меню.

### **2.3.1. HTML-разметка**

Главная страница представляет собой промо-сайт медицинской информационной системы MIS, содержащий описание её возможностей, преимущества, отзывы клиентов, примеры интеграций и призывы к подключению с бесплатным тестовым периодом.

Шапка главной страницы (рисунок 13) содержит: логотип, кнопки для входа и регистрации, навигационное меню с разделами и боковое меню для мобильной версии.

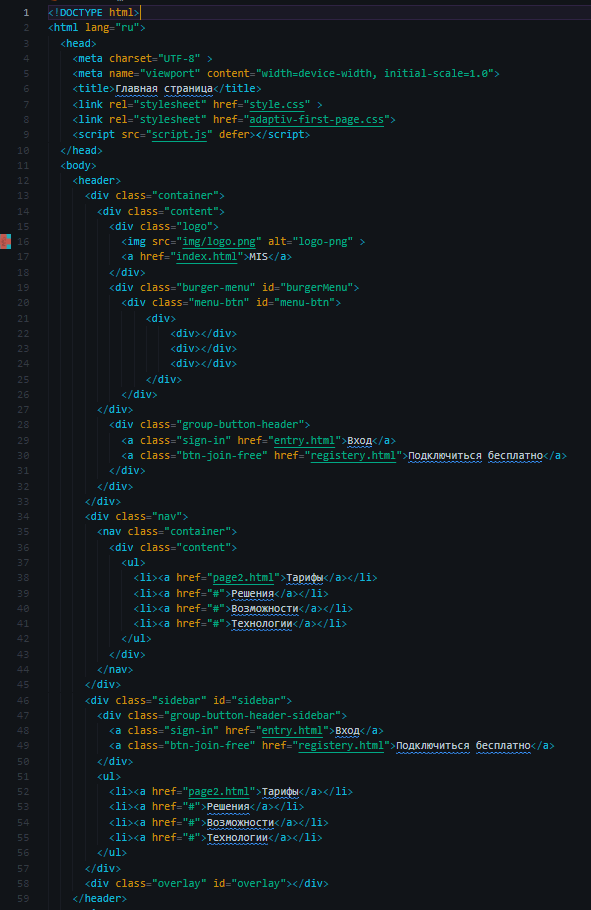


Рисунок 13. HTML разметка шапки главной страницы

Основной контент главной страницы (рисунок 14) содержит секции с описанием продукта, информации о достижениях компании, отзывы клиентов, аналитика, интеграция и другое.

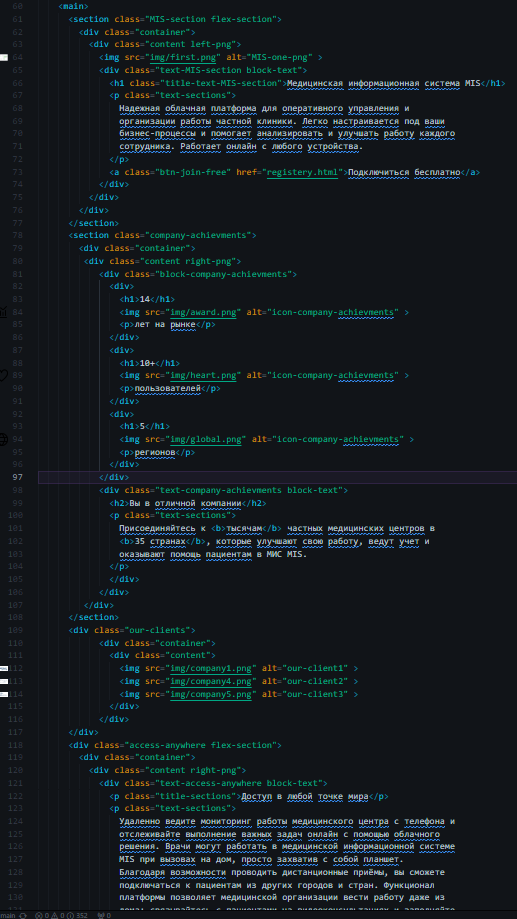


Рисунок 14. HTML разметка фрагмента основного контента главной страницы

Подвал главной страницы (рисунок 15) содержит ссылки на основные разделы сайта, информацию о компании и контактные данные.

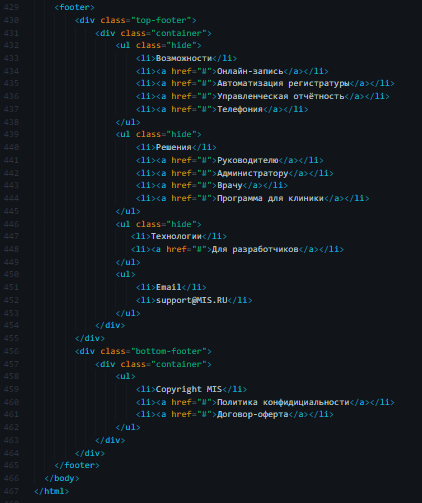


Рисунок 15. HTML разметка подвала главной страницы

Страница тарифы необходима для предоставления пользователям информацию о доступных тарифных планах, их стоимости и функциональных возможностях, а также побудить их подключиться к системе.

Шапка тарифов (рисунок 16) включает: логотип, навигационное меню

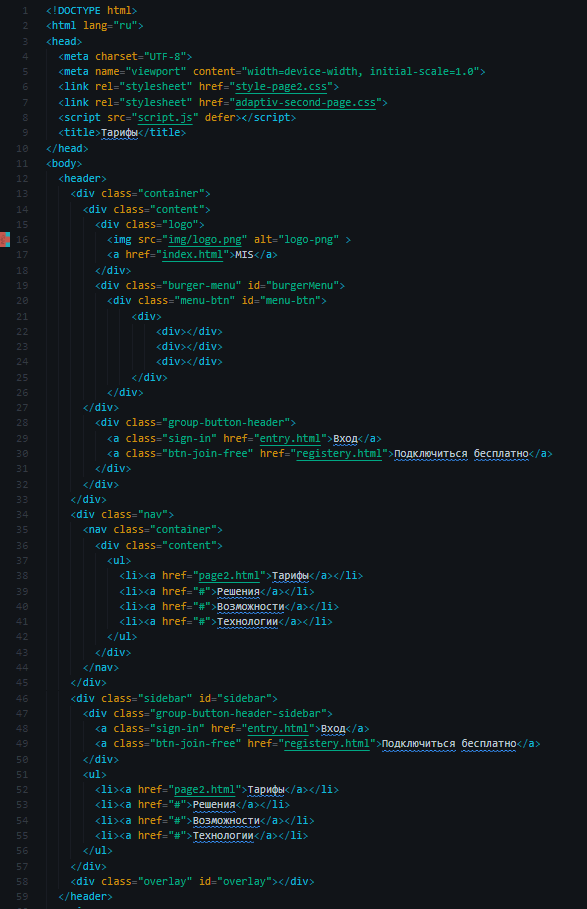


Рисунок 16. HTML разметка шапки страницы тарифы

Основной контент тарифов (рисунок 17) включает в себя: блоки с тарифами, в которых перечислены название тарифа, стоимость, что входит в него.

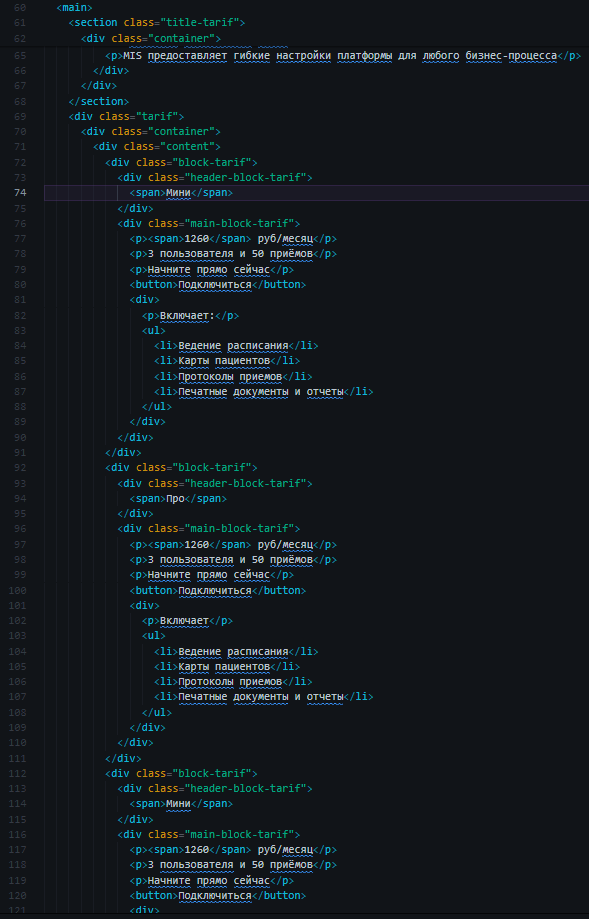


Рисунок 17. HTML разметка основного контента страницы тарифы

Подвал тарифов (рисунок 18) включает в себя ссылки на основные разделы сайта, информацию о компании и контактные данные.



Рисунок 18. HTML разметка подвала страницы тарифы

Вход и регистрация на сайте служат для предоставления пользователю персонализированного доступа к сервису, связанного с медицинской информационной системой «MIS». Эти функции помогают разграничить права доступа и предоставить возможности.

Страница входа и страница регистрации имеют один файл веб-разметки (рисунок 19), но отличаются расположением. Вход представляет собой форму, которая принимает логин, пароль и кнопку. Регистрация также представляет собой форму, но отличается тем, что она регистрирует нового пользователя в системе. Форма принимает логин, пароль, повторение пароля и подтверждение согласия на обработку данных.

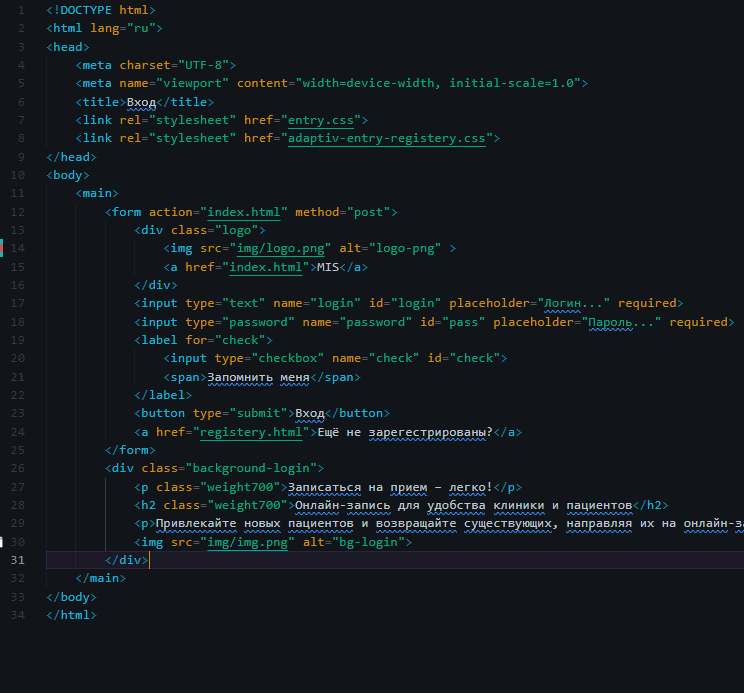


Рисунок 19. HTML разметка для страницы вход

### **2.3.2. CSS-стилизация**

Для каждой страницы сайта был применён набор CSS-стилей, обеспечивающих удобное и эстетичное отображение различных элементов интерфейса. Все стили были разделены на специфические для каждой из них.

Для главной страницы были использованы стили (рисунок 20), направленные на оформление навигационного меню, заголовков, кнопок и других элементов. Основное внимание уделено адаптивности и взаимодействию с пользователем.

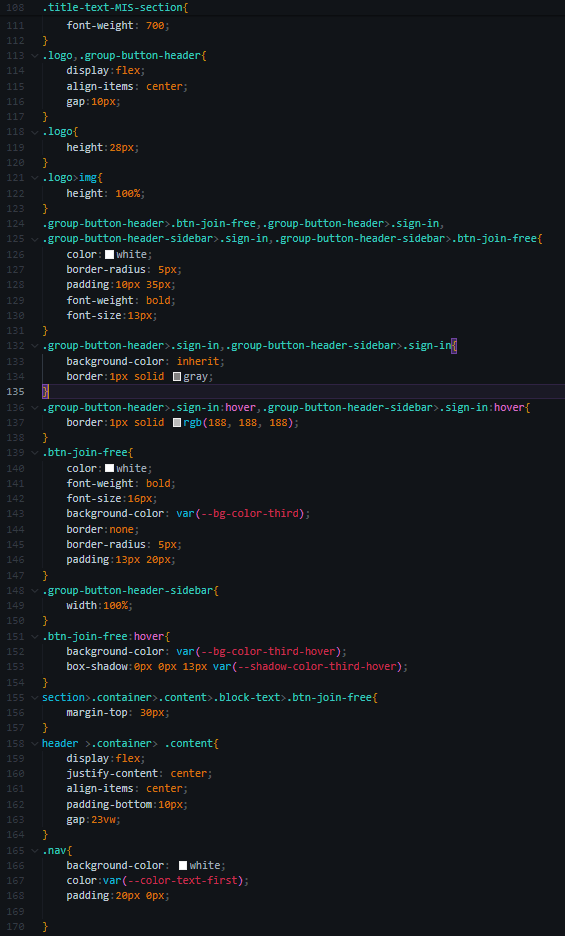


Рисунок 20. Фрагмент стилей для главной страницы

Страница входа имеет форму для авторизации пользователей (рисунок 21) CSS стили направлены на оформление формы, кнопок и фона страницы.

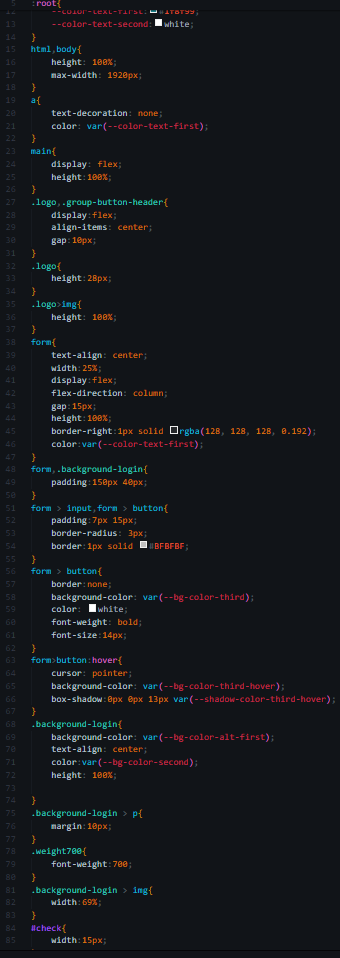


Рисунок 21. Стили для страницы входа

Для страницы регистрации были использованы аналогичные стили (рисунок 22) как для страницы входа, но с добавлением дополнительных полей формы, ссылки на вход и изменена позиция элементов.

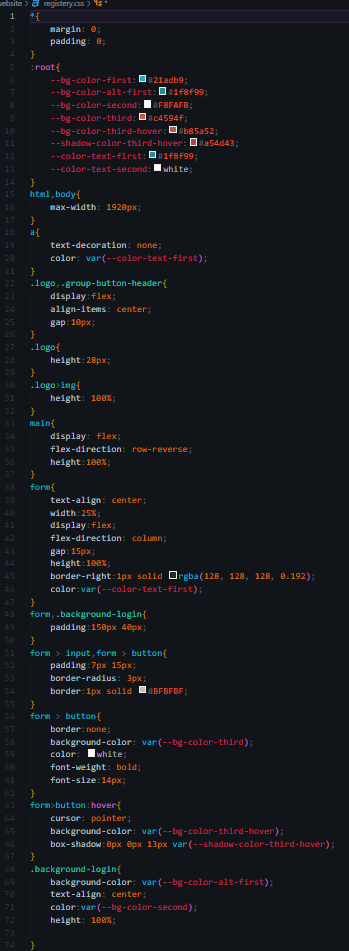


Рисунок 22. Стили для страницы регистрации

Для страницы тарифов используются стили для блоков с тарифами (рисунок 23), заголовков и кнопок. Важной задачей было оформление блоков с тарифами, чтобы пользователи могли легко их различать и выбирать.



Рисунок 23. Стили для страницы тарифы

### **2.3.3. Адаптивность верстки**

Адаптивная верстка — это ключевой аспект разработки веб-страниц, позволяющий корректно отображать сайт на устройствах с различными размерами экранов (от десктопов до мобильных телефонов). Для реализации адаптивности в этом проекте использовалась комбинация CSS-стилей и медиа-запросов, что позволило обеспечить удобство просмотра и взаимодействия с сайтом на всех типах устройств.

На главной странице использованы медиа-запросы (рисунок 24) для адаптации элементов к мобильным устройствам. Эти запросы необходимы для динамического изменения расположения элементов в зависимости от ширины экрана. Например, на мобильных устройствах меню и другие блоки перестраиваются для удобства навигации, что значительно улучшает восприятие интерфейса. Особенно важным элементом является добавление «Гамбургер», которое заменяет традиционное горизонтальное меню на компактный и доступный элемент, не занимающий лишнего пространства.

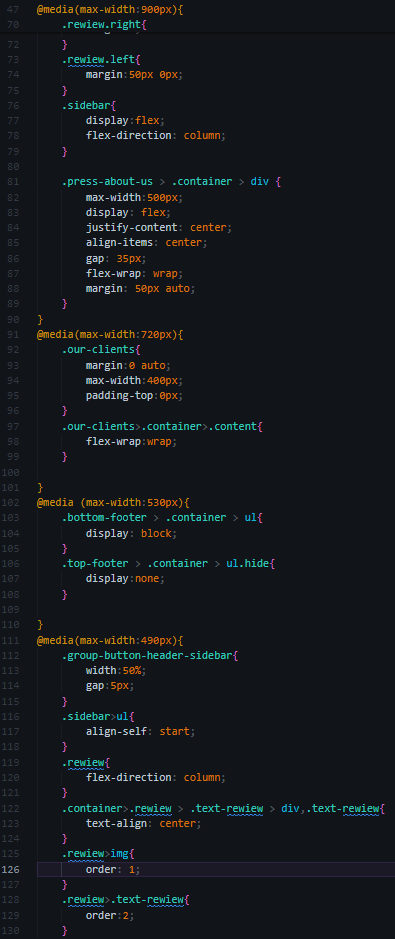


Рисунок 24. Медиа-запросы для главной страницы

Для страницы входа и регистрации на мобильных (рисунок 25) устройствах форма входа сжата до максимально удобного для пользователя размера. Также используется более компактное расположение кнопок и полей ввода.

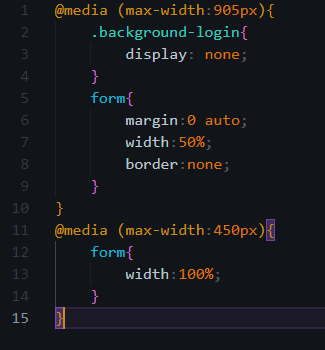


Рисунок 25. Медиа-запросы для страниц вход и регистрация

На странице тарифов была применена адаптивная верстка для правильного отображения блоков с тарифами на мобильных устройствах (рисунок 26). Вместо трёх колонок с тарифами, на мобильных устройствах тарифы показываются в одну колонку, что облегчает восприятие.

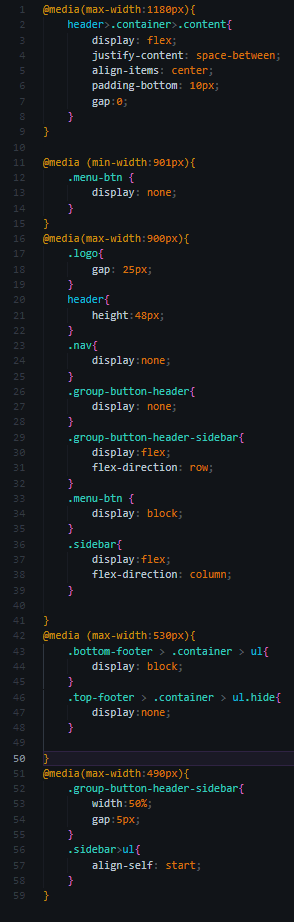


Рисунок 26. Медиа-запросы для страницы тарифы

### **2.3.4. Добавление интерактивности с JS**

Для добавления «Гамбургер» меню (рисунок 27) заместо навигационной панели, а также сделать отображение шторки с отображением навигационных элементов потребуется JS.



Рисунок 27. Код для «Гамбургера»

Этот код реализует функционал гамбургер-меню, которое появляется при клике на кнопку, а затем скрывается при клике на оверлей (полупрозрачную область) или на кнопку.

Описание элементов:

1. menuBtn: это кнопка, которая при клике открывает или закрывает боковую панель (меню).
2. overlay: это полупрозрачный фон, который появляется, когда меню открыто. Он помогает закрыть меню, когда пользователь кликает на него.
3. sidebar: это боковая панель, которая содержит элементы навигации (например, ссылки).

Как работает код:

1. Клик по кнопке меню:

* При нажатии на кнопку с id menu-btn вызывается функция toggleSidebar().
* Эта функция проверяет текущее положение боковой панели:
* Если панель уже открыта (ее стиль left равен '0px'), то она закрывается, и оверлей скрывается.
* Если панель закрыта, она открывается, а оверлей становится видимым.

1. Клик по оверлею:

* Если пользователь кликает на полупрозрачный оверлей, то меню снова закрывается, а оверлей скрывается.

Пошаговое руководство:

1. Открытие меню:

* Когда пользователь нажимает на кнопку меню, боковая панель выезжает с левой стороны экрана, а оверлей становится видимым. Оверлей помогает сделать интерфейс более понятным, ограничив взаимодействие с остальной частью страницы.

1. Закрытие меню:

* Клик на оверлей или повторное нажатие на кнопку меню закрывает боковую панель, а оверлей исчезает.

### **2.3.5. Реализация графического интерфейса**

Готовый проект находиться по адресу: https://deepjohny828.github.io/website/

### **2.3.6. Тестирование проекта**

Проверка валидации кода проводилась при помощи W3C валидатора (рисунок 28).

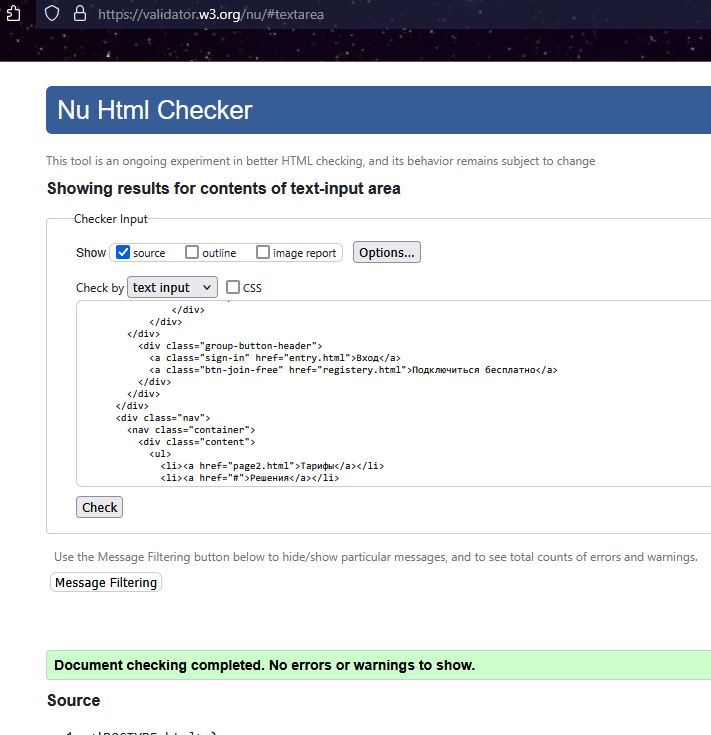
****

Рисунок 28. Проверка HTML на корректность

# **РАЗДЕЛ 3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ**

Важный фактор, влияющий на величину выручки от реализации продукта – процесс

ценообразования.

В конечном итоге цена продукта определяется рынком, она представляет собой компромисс между желанием продавца и возможностями покупателя.

Исходным пунктом формирования цены продукта является калькуляция. В настоящее время калькуляционный метод формирования цен остаётся основным, так как цена должна обеспечивать покрытие всех затрат и сверх того – получение прибыли.

Если рассчитанная калькуляционным методом цена окажется выше сложившейся на рынке, следует изыскать пути снижения затрат или отказаться от разработки данного продукта.

Наряду с калькуляционным методом применяются и другие методы формирования цен на вновь разработанные продукты: параметрические, зависящие от определённых параметров; статистический, основанный на анализе динамического ряда цен на аналогичные продукты.

Но при всех условиях цена должна возмещать затраты изготовителя (разработчика) продукта, поэтому использование калькуляций необходимо.

В Российских условиях проектируемая цена продукта при применении калькуляционного способа может рассчитываться (см. таблицу 2) следующим образом:

Таблица 2

Расчет стоимости продукта

| Статья затрат и элементы цены | Сумма, руб |
| --- | --- |
| Затраты на электроэнергию | 478,56 |
| Заработная плата разработчика | 7324,72 |
| Отчисления с заработной платы | 2203,42 |
| Амортизация основных средств | 3422,14 |
| Прочие расходы | 712,85 |
| Общая сумма затрат | 15163,69 |
| Прибыль (15%) | 2274,55 |
| Цена продукта: | 17438,24 руб |

## **3.1. Затраты на электроэнергию**

Затраты на электроэнергию складываются из расходов на освещение помещения и расходов на производственное потребление. Исходные данные для расчётов отражены в таблице 3.

Таблица 3

Затраты на электроэнергию

|  |  |
| --- | --- |
| Характеристика | Значение |
| Усредненный расход энергии, необходимой для освещения одного квадратного метра площади помещения в год, кВт | 28 |
| Площадь помещения, м2 | 1350 |
| Расход энергии за год, кВт | 10 |
| Количество дней для разработки продукта | 34 (=1350/365\*10) |
| Тариф за 1 кВт/час, руб | 5,75 |
| Стоимость электроэнергии для освещения помещения, руб | 183,5 |
| Количество компьютеров, шт. | 4 |
| Потребление энергии одним компьютером, кВт | 0,25 |
| Коэффициент, учитывающий потери в сети | 1,1 |
| Количество часов работы компьютера за период разработки, час | 85 |
| Стоимость электроэнергии для производственного потребления, руб | 297,4 (40,2585\*1,1) |
| Итого затраты на электроэнергию: | 480,9 |

## **3.2. Заработная плата разработчика**

Поскольку среднемесячная заработная плата составляет 54 317 рублей, то затраты на оплату труда разработчика могут составить: 54 317 рублей разделить на количество рабочих дней в месяце (26) и умножить на количество дней разработки.

## **3.3. Отчисления от заработной платы**

В настоящее время размер отчислений составляет 32,1 % от фонда заработной платы.

## **3.4. Амортизация основных средств**

Средняя стоимость компьютера (ноутбука) составляет 74 500 рублей. Норма

амортизации 4,5% в мес. Срок эксплуатации составляет 22 месяца, значит, норма амортизации рассчитывается следующим образом: 100% разделить на 22 мес. (= 4,5%) Значит, амортизация за период разработки продукта составляет: 74 500 рублей умножить на 4,5%, разделить на 32 дней и умножить на количество дней разработки.

## **3.5. Прочие расходы**

Прочие расходы в размере 5,2% от суммы предыдущих расходов.

## **3.6. Общая сумма затрат**

Общая сумма затрат складывается из суммы всех вышеперечисленных расходов.

## **3.7. Прибыль**

Экономический интерес любого предприятия заключается в получении прибыли, которая необходима для дальнейшего развития. В цену продукта заложим получение прибыли в размере 15% от общей суммы затрат.

## **3.8. Цена продукта**

В результате расчётов формируется цена продукта, которая складывается из общих затрат и прибыли.

Таким образом цена продукта составляет: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ руб.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения работы был разработан веб-сайт для медицинской информационной системы MIS, состоящий из главной страницы, страницы тарифов, а также страниц входа и регистрации. Основной целью проекта было создание удобного и информативного ресурса, который демонстрирует возможности системы, её преимущества, отзывы клиентов и примеры интеграций, предоставляя пользователям доступ к ключевым функциям, таким как подключение к сервису с бесплатным тестовым периодом.

Все поставленные задачи были выполнены:

* + Проведен анализ существующих решений для медицинских сайтов, выявлены сильные стороны и недостатки, которые учтены при разработке.
  + Определены требования к функционалу и дизайну сайта, что обеспечило интуитивно понятный и удобный интерфейс для целевой аудитории.
  + Реализована структура сайта, включающая адаптивный дизайн для корректного отображения на различных устройствах, что повысило удобство использования.
  + Внедрены современные технологии HTML, CSS и JavaScript для обеспечения визуальной привлекательности, функциональности и интерактивности сайта.

Сайт предоставляет пользователям четкую и доступную информацию о системе MIS, её функциях и возможностях. Главная страница стала ключевым элементом, которая знакомит пользователей с MIS, содержит призывы к регистрации и интеграцию с другими страницами, такими как тарифы, вход и регистрация.

Таким образом, цели и задачи, поставленные в начале работы, успешно достигнуты. Разработанный сайт соответствует требованиям современного веб-дизайна и обеспечивает эффективное представление медицинской информационной системы MIS, создавая комфортные условия для взаимодействия пользователей с системой.

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

I. Нормативно-правовые акты

Нет нормативно-правовых актов.

II. Основные источники

1. Метанит. HTML5. – Текст : электронный // Метанит : [сайт]. – URL: https://metanit.com/web/html5/ (дата обращения: 10.11.2024).

2. Код.ру. Книга по разметке. – Текст : электронный // Код.ру : [сайт]. – URL: https://code.mu/ru/markup/book/prime/ (дата обращения: 11.11.2024).

3. Mozilla Developer Network. Введение в HTML. – Текст : электронный // Mozilla Developer Network : [сайт]. – URL: https://developer.mozilla.org/ru/docs/Learn/HTML (дата обращения: 12.11.2024).

4. Mozilla Developer Network. Основы CSS. – Текст : электронный // Mozilla Developer Network : [сайт]. – URL: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/CSS\_basics (дата обращения: 13.11.2024).

5. HTML Academy. Flexbox для начинающих. – Текст : электронный // HTML Academy : [сайт]. – URL: https://htmlacademy.ru/blog/css/flex-for-beginner (дата обращения: 04.12.2024).

6. Mozilla Developer Network. Основы Flexbox. – Текст : электронный // Mozilla Developer Network : [сайт]. – URL: https://developer.mozilla.org/ru/docs/Web/CSS/CSS\_flexible\_box\_layout/Basic\_concepts\_of\_flexbox (дата обращения: 14.11.2024).

7. ТПрогер. Как работает CSS Flexbox. – Текст : электронный // ТПрогер : [сайт]. – URL: https://tproger.ru/translations/how-css-flexbox-works (дата обращения: 04.12.2024).

КиберЛенинка. Грид-технологии: статус и перспективы. – Текст : электронный // КиберЛенинка : [сайт]. – URL: https://cyberleninka.ru/article/n/grid-tehnologii-status-i-perspektivy (дата обращения: 15.11.2024).

8. Mozilla Developer Network. Основы CSS Grid Layout. – Текст : электронный // Mozilla Developer Network : [сайт]. – URL: https://developer.mozilla.org/ru/docs/Web/CSS/CSS\_grid\_layout/Basic\_concepts\_of\_grid\_layout (дата обращения: 16.11.2024).

9. Ultimeta. Грид-технологии. – Текст : электронный // Ultimeta : [сайт]. – URL: http://www.ultimeta.ru/technologies/grid-technologies/ (дата обращения: 17.11.2024).

10. Метанит. Django. – Текст : электронный // Метанит : [сайт]. – URL: https://metanit.com/python/django/ (дата обращения: 18.11.2024).

11. Mozilla Developer Network. Введение в Django. – Текст : электронный // Mozilla Developer Network : [сайт]. – URL: https://developer.mozilla.org/ru/docs/Learn/Server-side/Django (дата обращения: 19.11.2024).

12. Практикум Яндекса. Что такое Use Case и как их написать. – Текст : электронный // Практикум Яндекса : [сайт]. – URL: https://practicum.yandex.ru/blog/chto-takoe-use-case-kak-ih-napisat/ (дата обращения: 20.11.2024).

13. Практикум Яндекса. Диаграмма последовательности. – Текст : электронный // Практикум Яндекса : [сайт]. – URL: https://practicum.yandex.ru/blog/sequence-diagram/ (дата обращения: 21.11.2024).

14. Национальная команда WorldSkills. Проектирование диаграмм деятельности UML (Activity Diagram). – Текст : электронный // WorldSkills : [сайт]. – URL: https://nationalteam.worldskills.ru/skills/proektirovanie-diagrammy-deyatelnosti-uml-activity-diagram/ (дата обращения: 22.11.2024).

15. SkyPro. Обзор Yougile. – Текст : электронный // SkyPro : [сайт]. – URL: https://sky.pro/wiki/profession/obzor-yougile/ (дата обращения: 23.11.2024).

16. Robotrends. Лексикон в области летящих беспилотников. – Текст : электронный // Robotrends : [сайт]. – URL: https://robotrends.ru/robopedia/leksikon-v-oblasti-letayushih-bespilotnikov (дата обращения: 24.11.2024).

17. SkyPro. Работа с фреймами в Figma: советы и трюки. – Текст : электронный // SkyPro : [сайт]. – URL: https://sky.pro/wiki/digital-art/rabota-s-frejmami-v-figma-sovety-i-tryuki/ (дата обращения: 25.11.2024).