МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ

**ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ ОКЭИ)**

**КУРСОВОЙ ПРОЕКТ**

ОКЭИ 09.02.07. 4 0 25. 14 О

(код документа)

|  |
| --- |
| *Стоматологическая клиника* |
|  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Количество листов | | | | | *N* | | |
| Дата готовности | | | | *10.11.2024* | | | |
| Руководитель | | | *Адамович Н.В.* | | | | |
| Разработал | | *Костюков М.В.* | | | | | |
| Защищен |  | | | | | с оценкой |  |

Оренбург 2025

**Содержание**

[Введение 2](#_Toc2594)

[1 Анализ предметной области 4](#_Toc11899)

[1.1. Определение проблемной сферы и потребности в автоматизации 4](#_Toc30135)

[1.2. Сущности предметной области 4](#_Toc19036)

[1.3. Процессы предметной области 6](#_Toc14293)

[2 Проектирование приложения 8](#_Toc11744)

[2.1. Введение 8](#_Toc20043)

[2.2. Требования к программному продукту 8](#_Toc27541)

[2.3. Основные модели приложения 9](#_Toc17794)

[3 Разработка программного обеспечения 10](#_Toc23052)

[3.1. Описание технологического стека разработки 10](#_Toc20073)

[3.2. Описание алгоритма работы 11](#_Toc20625)

[3.2.1. Принцип функционирования приложения 11](#_Toc3282)

[3.2.2. Ключевые моменты работы приложения 11](#_Toc11927)

[3.2.3. Структура кода 11](#_Toc10656)

[3.3. Описание интерфейса пользователя 12](#_Toc11182)

[4 Тестирование приложения 13](#_Toc20340)

[Заключение 14](#_Toc7166)

[Список используемых источников 15](#_Toc11713)

# Введение

Современные технологии веб-разработки играют ключевую роль в обеспечении доступности и удобства предоставления услуг, включая сферу здравоохранения и стоматологии. Разработка веб-сайтов для медицинских учреждений позволяет не только информировать клиентов об услугах и продукции, но и предоставлять функциональные возможности для взаимодействия, такие как покупка товаров, запись на приём и управление заказами. В рамках данной работы был разработан веб-сайт стоматологической клиники «SmileLand», который сочетает в себе информационные разделы, каталог товаров для ухода за полостью рта, систему авторизации, корзину покупок и адаптивный интерфейс.

Актуальность выбранной темы обусловлена растущей цифровизацией медицинских услуг и необходимостью создания удобных и функциональных онлайн-платформ для взаимодействия с клиентами. В условиях высокой конкуренции в стоматологической отрасли наличие веб-сайта с интуитивно понятным интерфейсом, возможностью покупки товаров и записи на приём становится важным конкурентным преимуществом. Кроме того, адаптивный дизайн, обеспечивающий корректное отображение на устройствах с разными разрешениями экрана, отвечает современным требованиям пользователей, значительная часть которых использует мобильные устройства для доступа к интернет-ресурсам. Разработка такого веб-сайта позволяет изучить и применить актуальные технологии веб-разработки, включая «HTML», «CSS», «JavaScript», а также принципы проектирования пользовательского интерфейса («UI/UX»).

Цель проекта заключается в создании автоматизированной функциональной системы для стоматологической клиники.

Задачи проекта:

* разработать структуру сайта;
* реализовать адаптивный дизайн;
* создать интерактивные элементы;
* настроить клиентскую логику корзины;
* добавить визуальные эффекты;
* обеспечить единый стиль и навигацию.

Раскрытие задач включает проектирование страниц (главная, о клинике, каталог, авторизация, регистрация), реализацию адаптивной вёрстки с «HTML5», «CSS3» и медиа-запросами, создание слайдера отзывов, модальных окон (корзина, ошибка, промо-акция) и анимаций с «JavaScript». Также настроена корзина с добавлением/удалением товаров и сохранением в «localStorage», добавлены эффекты (гифки, конфетти) и обеспечена удобная навигация с единым стилем.

Практическая значимость полученных результатов заключается в создании готового прототипа веб-сайта, который может быть использован стоматологической клиникой для продвижения услуг и продажи товаров. Веб-сайт предоставляет пользователям удобный доступ к каталогу продукции, возможность оформления заказов и записи на приём, что способствует повышению удовлетворённости клиентов и оптимизации бизнес-процессов клиники. Кроме того, проект демонстрирует применение современных подходов к веб-разработке, таких как адаптивная вёрстка, модульная структура «CSS» и клиентская логика на «JavaScript», что может служить основой для дальнейших учебных или коммерческих проектов.

Ресурсы, необходимые для реализации проекта, включают:

* технические: ПК, интернет, браузеры, редактор кода («VS Code»).
* программные: «HTML5», «CSS3», «JavaScript», «Google Fonts», «localStorage».
* графические: изображения (фоны, логотип, иконки), гифка ошибки.

Перспективы расширения функционала приложения включают:

* интеграция серверной части.
* добавление системы оплаты.
* расширение личного кабинета.
* внедрение чат-бота.
* оптимизация производительности.
* разработка мобильного приложения.

Данная работа посвящена созданию прототипа веб-сайта, который демонстрирует возможности современных технологий веб-разработки и может служить основой для дальнейшего развития функционального решения для стоматологической клиники.

# 1 Анализ предметной области

## 1.1. Определение проблемной сферы и потребности в автоматизации

Предметная область данного проекта связана с деятельностью стоматологической клиники «SmileLand», которая предоставляет услуги по уходу за полостью рта и продаже сопутствующих товаров (зубные пасты, щетки, ополаскиватели, зубные нити и дополнительные средства). Основная цель веб-сайта – автоматизация процессов взаимодействия с клиентами, включая предоставление информации о клинике, запись на прием, оформление заказов на товары и управление клиентскими данными. Проблемная сфера заключается в необходимости упрощения доступа клиентов к услугам и товарам, улучшения их пользовательского опыта и оптимизации внутренних процессов клиники, таких как обработка заявок и управление заказами.

В традиционной модели стоматологической клиники клиенты сталкиваются с неудобствами, связанными с необходимостью личного или телефонного контакта для записи на прием, отсутствием прозрачной информации о товарах и услугах, а также сложностями в управлении покупками. Веб-сайт «SmileLand» решает эти проблемы, предоставляя централизованную платформу для автоматизации следующих процессов:

* запись на прием через онлайн-форму;
* просмотр и покупка товаров для ухода за полостью рта;
* регистрация и авторизация клиентов для персонализации услуг;
* информирование клиентов об акциях и специальных предложениях;
* управление корзиной покупок с сохранением данных.

Автоматизация этих процессов позволяет сократить время на взаимодействие с клиентами, минимизировать человеческий фактор в обработке заявок и повысить удовлетворенность пользователей за счет удобного интерфейса и интерактивных функций.

## 1.2. Сущности предметной области

Предметная область стоматологической клиники как организации охватывает процессы оказания медицинских услуг, продажи сопутствующих товаров и управления клиентскими взаимодействиями. Независимо от функций веб-сайта «SmileLand», ниже описаны ключевые сущности, характерные для деятельности стоматологической клиники как организации:

* клиент;
* сотрудник;
* услуга;
* товар;
* запись на прием;
* заказ;
* медицинская карта;
* акция.

Клиент представляет собой физическое лицо, которое обращается в стоматологическую клинику за медицинскими услугами или покупкой товаров для ухода за полостью рта. Клиент характеризуется персональными данными, такими как имя, фамилия, контактный телефон, адрес электронной почты, а в некоторых случаях — адрес проживания. Клиенты могут быть постоянными или разовыми, и их взаимодействие с клиникой включает запись на прием, получение консультаций, лечение, покупку товаров и участие в акциях. Эта сущность является центральной, так как все процессы клиники ориентированы на удовлетворение потребностей клиентов.

Сотрудник – это работник клиники, выполняющий профессиональные обязанности. К сотрудникам относятся стоматологи, ассистенты, администраторы, маркетологи и технический персонал. Каждый сотрудник имеет атрибуты, такие как имя, должность, квалификация, график работы и контактные данные. Сотрудники участвуют в оказании услуг, обработке записей на прием, продаже товаров и взаимодействии с клиентами, обеспечивая функционирование клиники.

Услуга представляет собой медицинскую или консультационную процедуру, предоставляемую клиникой. Услуги включают профилактические осмотры, лечение кариеса, ортодонтию, имплантацию, отбеливание зубов и другие процедуры. Каждая услуга характеризуется названием, описанием, стоимостью, продолжительностью и требованиями к квалификации исполнителя. Услуги являются основным источником дохода клиники и ключевым элементом взаимодействия с клиентами.

Товар – это продукция, продаваемая клиникой для ухода за полостью рта, такая как зубные пасты, щетки, ополаскиватели, зубные нити и дополнительные средства (например, ирригаторы). Товары характеризуются названием, категорией, ценой, производителем и характеристиками (например, объем или тип щетины). Они дополняют медицинские услуги, предоставляя клиентам возможность поддерживать гигиену дома, и являются частью коммерческой деятельности клиники.

Запись на прием отражает процесс бронирования времени для посещения клиники клиентом. Эта сущность включает данные о клиенте (имя, телефон, email), времени и дате приема, типе услуги и назначенном сотруднике. Запись на прием обеспечивает планирование работы клиники, распределение ресурсов и удобство для клиентов, минимизируя ожидание и конфликты в расписании.

Заказ представляет собой покупку товаров клиентом, оформленную в клинике или через онлайн-платформу. Заказ включает список товаров, их количество, общую сумму и статус (например, оформлен, оплачен, доставлен). Эта сущность связывает клиентов с коммерческой деятельностью клиники, обеспечивая учет продаж и управление запасами товаров.

Медицинская карта хранит информацию о здоровье клиента и истории его посещений клиники. Она включает данные о диагнозах, проведенных процедурах, назначениях, аллергиях и рекомендациях. Медицинская карта используется сотрудниками для планирования лечения, отслеживания прогресса и обеспечения качества медицинских услуг, а также для соблюдения нормативных требований.

Акция – это маркетинговое предложение, направленное на привлечение клиентов. Акции могут включать скидки на услуги или товары, специальные предложения для новых клиентов или программы лояльности. Каждая акция характеризуется описанием, сроком действия, условиями участия и целевой аудиторией. Эта сущность помогает клинике увеличивать поток клиентов и стимулировать повторные посещения.

## 1.3. Процессы предметной области

Ниже описаны ключевые процессы, характерные для деятельности стоматологической клиники как организации:

* оказание медицинских услуг;
* запись на прием;
* продажа товаров;
* ведение медицинских карт;
* проведение акций;
* управление персоналом;
* учет и логистика товаров.

Оказание медицинских услуг является основным процессом деятельности стоматологической клиники. Этот процесс включает консультации, диагностику, лечение и профилактические процедуры, выполняемые квалифицированными стоматологами. Он начинается с обращения клиента, продолжается через диагностику и выбор подходящей услуги, и завершается выполнением процедуры с последующим предоставлением рекомендаций. Процесс требует координации между сотрудниками, клиентом и медицинской картой для обеспечения качества и безопасности.

Запись на прием организует взаимодействие клиента с клиникой для получения услуг. Клиент выбирает удобное время и услугу, связывается с клиникой (лично, по телефону или через онлайн-систему), а администратор или система фиксирует запись, согласовывая ее с расписанием сотрудников. Этот процесс минимизирует ожидание клиентов и оптимизирует загрузку персонала.

Продажа товаров охватывает процесс реализации продукции для ухода за полостью рта. Клиент выбирает товары в клинике или через каталог, оформляет заказ, оплачивает его и получает продукцию. Процесс включает учет товаров, управление запасами и обработку платежей, обеспечивая дополнительный доход клиники и удобство для клиентов.

Ведение медицинских карт обеспечивает документирование истории посещений и лечения клиентов. Сотрудники вносят данные о диагнозах, процедурах и рекомендациях в медицинскую карту после каждого визита. Этот процесс необходим для планирования дальнейшего лечения, контроля качества услуг и соблюдения медицинских стандартов.

Проведение акций направлено на привлечение и удержание клиентов. Маркетинговый отдел разрабатывает акционные предложения, информирует клиентов через различные каналы (рассылки, объявления в клинике, социальные сети) и отслеживает их участие. Процесс включает анализ эффективности акций и корректировку маркетинговых стратегий.

Управление персоналом включает найм, обучение, составление графиков работы и оценку эффективности сотрудников. Этот процесс обеспечивает наличие квалифицированного персонала, способного выполнять медицинские и административные задачи, и поддерживает бесперебойную работу клиники.

Учет и логистика товаров охватывает управление запасами продукции, закупку, хранение и распределение товаров. Клиника отслеживает остатки, прогнозирует спрос и организует поставки, чтобы товары были доступны для продажи. Этот процесс поддерживает коммерческую деятельность и предотвращает дефицит продукции.

# 2 Проектирование приложения

## 2.1. Введение

Проектирование веб-сайта стоматологической клиники «SmileLand» направлено на создание удобной и функциональной платформы для автоматизации взаимодействия с клиентами, включая запись на прием, покупку товаров и информирование об акциях. На основе анализа предметной области сформулированы функциональные и нефункциональные требования к программному продукту. В данном разделе описываются основные модели приложения, включая функциональные возможности, архитектуру и пользовательские сценарии. Для наглядности представлена диаграмма вариантов использования (приложение А), иллюстрирующая взаимодействие пользователей с системой. Проектирование ведется с точки зрения еще не существующего приложения, чтобы отразить процесс разработки на этапе планирования.

## 2.2. Требования к программному продукту

2.2.1. Функциональные требования

Функциональные требования определяют ключевые задачи, которые должен выполнять веб-сайт «SmileLand» для обеспечения взаимодействия с клиентами стоматологической клиники.

Ниже перечислены функции разрабатываемой системы:

* просмотр информации о клинике (предоставляет доступ к статической странице с данными о миссии, истории и контактах клиники);
* запись на прием через форму (позволяет отправить заявку на визит с указанием имени, телефона, email и комментария);
* просмотр каталога товаров (отображает продукцию, разделенную по категориям, с основными характеристиками);
* добавление товаров в корзину (сохраняет выбранные товары для последующей покупки);
* управление содержимым корзины (обеспечивает изменение количества товаров, их удаление и инициирование заказа);
* регистрация пользователя (создает учетную запись с личными данными);
* авторизация пользователя (предоставляет доступ к учетной записи через email и пароль);
* подписка на акционные рассылки (позволяет подписаться на маркетинговые предложения через форму);
* просмотр отзывов клиентов (демонстрирует отзывы в формате слайдера с навигацией).

2.2.2. Нефункциональные требования

Нефункциональные требования определяют качественные характеристики приложения и представлены ниже:

* адаптивность (корректное отображение на устройствах с разными экранами);
* производительность (загрузка страниц за 3 секунды, плавная работа интерактивных элементов);
* удобство использования (интуитивный интерфейс с минимальным количеством кликов);
* визуальная привлекательность (современный дизайн с фирменными цветами, шрифтом «Segoe UI» и анимациями для улучшения опыта);
* Доступность (поддержка клавиатурной навигации и экранных читалок);
* Безопасность (валидация данных на клиенте, с перспективой «HTTPS» для передачи данных);
* кроссбраузерность (работа в браузерах «Chrome», «Firefox», «Safari», «Edge»);
* масштабируемость (возможность добавления новых функций без значительных изменений кода).

## 2.3. Основные модели приложения

2.3.1. Архитектура приложения

Веб-сайт «SmileLand» проектируется как клиентское веб-приложение с использованием «HTML5», «CSS3» и «JavaScript». Клиентская часть включает HTML для создания структуры страниц («index.html», «about.html», «catalog.html», «login.html», «register.html»), содержащих общие элементы («хедер», «футер», «топ-слайдер») и уникальные секции. «CSS» отвечает за стилизацию и адаптивный дизайн: общие стили («common.css») определяют оформление «хедера», «футера», модальных окон и анимаций, а специфические стили («index.css», «about.css», «catalog.css», «login.css», «register.css») задают внешний вид отдельных страниц. «JavaScript» обеспечивает интерактивность, включая управление корзиной, валидацию форм, работу слайдеров и модальных окон, с сохранением данных корзины в «localStorage». Серверная часть на этапе проектирования не реализуется, но предусматривается интеграция с сервером для обработки заказов, заявок и пользовательских данных с использованием технологий, таких как «Node.js» или «PHP», и базы данных, например «MySQL». Хранилище данных использует «localStorage» для корзины, с перспективой внедрения базы данных. Интерфейс включает адаптивный дизайн с «хедером», «футером», модальными окнами (корзина, акции, ошибки) и интерактивными элементами (слайдеры, формы, кнопки), ориентированный на простоту и визуальную привлекательность.

# 3 Разработка программного обеспечения

## 3.1. Описание технологического стека разработки

HTML (HyperText Markup Language, версия 5). Универсальность, поддержка всеми браузерами, простота интеграции с CSS и JavaScript. Формирование структуры веб-страниц, включая разметку для навигации, форм, каталога товаров, модальных окон и других элементов. Используется семантические теги (<header>, <section>, <footer>) для улучшения читаемости кода и доступности.

HTML5 обеспечивает поддержку мультимедиа и мета-тегов для адаптивности (например, <meta name="viewport">)

(см. рисунок 1 – структура файла index.html).



Рисунок 1 – Структура файла index.html

CSS (Cascading Style Sheets). Гибкость, поддержка адаптивного дизайна, высокая производительность за счет клиентской обработки. Стилизация интерфейса, включая оформление элементов, адаптивный дизайн, анимации и эффекты (например, масштабирование изображений при наведении).

Используется модульный подход с разделением стилей на файлы:

* common.css (общие стили (хедер, футер, модальные окна));
* index.css, about.css, catalog.css, login.css, register.css (стили для конкретных страниц);
* responsive.css (медиазапросы для адаптивности);
* hamburger.css (стили мобильного меню).

Используются CSS-анимации (@keyframes) для эффектов появления изображений и модальных окон (см. рисунок 2 – пример стилей в catalog.css).

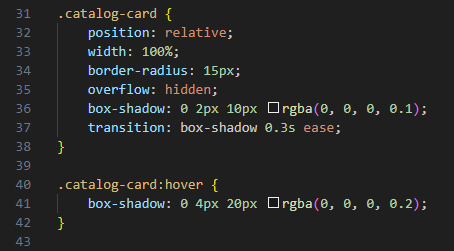


Рисунок 2 – Пример стилей в catalog.css (сетка товаров и эффект ховера)

JavaScript (ES6+). Быстрая интеграция, высокая производительность, отсутствие необходимости в сборщиках (например, Webpack). Обеспечение интерактивности, включая управление корзиной, валидацию форм, модальные окна и слайдер отзывов. Применяется ванильный JavaScript без фреймворков для минимизации зависимостей.

Логика реализована в двух файлах:

* scripts.js (управление корзиной, формами, модальными окнами и слайдером).
* mobile-menu.js (логика мобильного меню).

Используются события (click, submit, touchstart, touchend) для обработки пользовательских действий (см. рисунок 3 – код обработчиков в scripts.js).



Рисунок 3 – Код обработчиков событий в scripts.js

localStorage. Хранение данных корзины и состояния модального окна на стороне клиента. Простота реализации, отсутствие серверных запросов, быстрая работа с данными. Используется Сохранение массива cart (список товаров) и флага showPromoModal (для отображения промо-окна после регистрации) в формате JSON. Проект не использует сторонние фреймворки (например, React, Vue.js) или API, так как:

* ванильный JavaScript обеспечивает достаточную функциональность для реализации корзины, форм и слайдеров;
* отсутствие фреймворков сокращает объем кода и время загрузки страницы;
* отсутствие API упрощает развертывание, так как сайт полностью статический.

Выбранный стек (HTML, CSS, JavaScript, localStorage) оптимален для создания легковесного, быстрого и интерактивного веб-сайта с минимальными затратами на разработку и развертывание. Он обеспечивает баланс между функциональностью, производительностью и простотой поддержки.

## 3.2. Описание алгоритма работы

3.2.1 Принцип функционирования приложения

Веб-сайт "SmileLand" – это статическое веб-приложение, предоставляющее пользователю следующие возможности:

* просмотр страниц: главная (index.html), о клинике (about.html), каталог товаров (catalog.html), вход (login.html), регистрация (register.html);
* добавление товаров в корзину и управление ею (изменение количества, удаление);
* регистрация/авторизация пользователей с валидацией форм;
* запись на прием через форму с анимацией подтверждения;
* отображение модальных окон (корзина, ошибка, промо) и слайдера отзывов с поддержкой сенсорного управления.

Процесс добавления товара в корзину и отображения корзины представлен на диаграмме последовательности (см. Приложение Б). Алгоритм включает следующие шаги:

* пользователь нажимает кнопку "Добавить в корзину" на странице каталога.
* javascript-обработчик извлекает данные о товаре (название, цена, изображение) из DOM;
* данные добавляются в массив cart, который сохраняется в localStorage через функцию saveCart();
* функция updateCart() обновляет содержимое модального окна корзины и отображает его.

3.2.2. Ключевые моменты работы приложения

Хранение данных. Данные корзины (cart) и флаг отображения промо-окна (showPromoModal) хранятся в localStorage в формате JSON (см. рисунок 4 — код функции saveCart() в scripts.js).

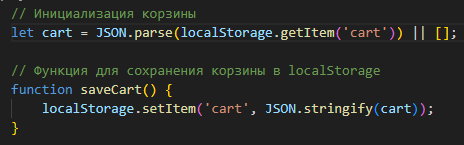


Рисунок 4 – Код функции saveCart() в scripts.js

Массив cart содержит объекты с полями name (название), price (цена), image (URL изображения), quantity (количество).

Добавление товара в корзину. При клике на кнопку .add-to-cart JavaScript извлекает данные из элемента .catalog-item (см. рисунок 5 – код обработчика .add-to-cart в scripts.js).

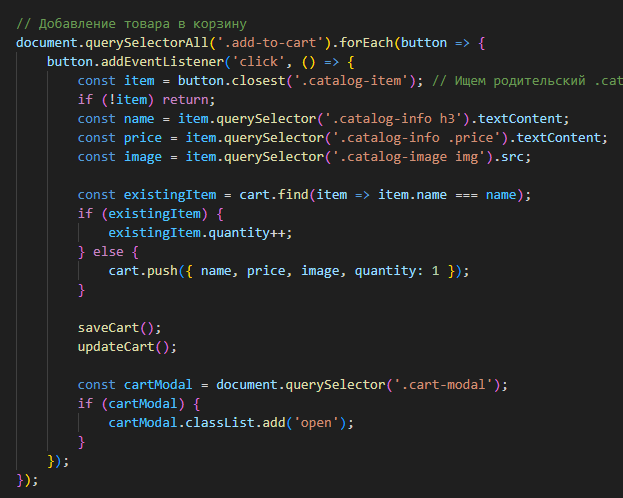


Рисунок 5 – Код обработчика .add-to-cart в scripts.js

Если товар уже есть в корзине, увеличивается поле quantity; иначе создается новый объект. После добавления вызываются функции saveCart() и updateCart(), а модальное окно корзины открывается автоматически.

Модальные окна. События click на кнопках .cart-button, .cart-modal-close, .promo-link, .cancel-button управляют отображением/скрытием корзины, окна ошибки и промо-окна (см. рисунок 6 – код обработчиков модальных окон в scripts.js).

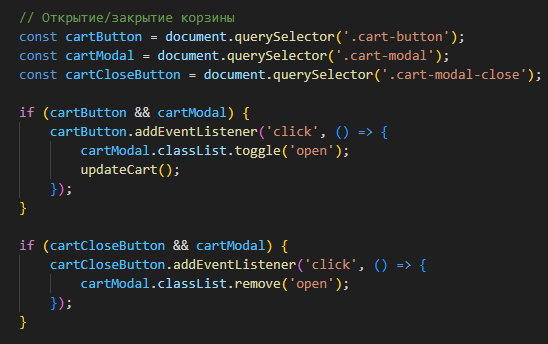


Рисунок 6 – Код обработчиков модальных окон в scripts.js

Промо-окно открывается автоматически после регистрации/авторизации, если установлен флаг showPromoModal.

Валидация форм. Формы (loginForm, registerForm, appointmentForm, promoForm) обрабатывают событие submit с валидацией полей (email, пароль, телефон) с использованием регулярных выражений (см. рисунок 7 – код валидации формы в scripts.js).



Рисунок 7 – Код валидации формы в scripts.js

Слайдер отзывов. События click на стрелках (prev-arrow, next-arrow) и сенсорные события (touchstart, touchend) обеспечивают переключение карточек отзывов (см. Приложение А, рисунок 8 – код слайдера в scripts.js).

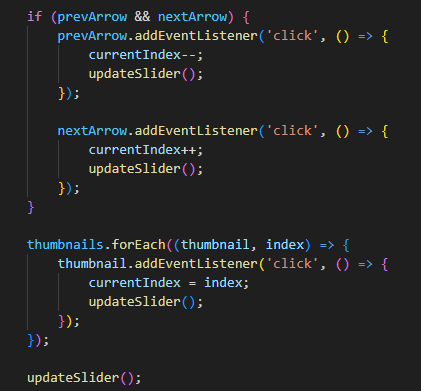


Рисунок 8 – Код слайдера отзывов в scripts.js

3.2.3. Структура кода

Код организован по модульному принципу для упрощения поддержки и масштабируемости:

* index.html (главная страница с баннером, отзывами и формой записи).
* about.html (информация о клинике).
* catalog.html (каталог товаров с возможностью добавления в корзину).
* login.html и register.html (формы авторизации и регистрации).
* common.css (общие стили (контейнер, хедер, футер, модальные окна)).
* index.css, about.css, catalog.css, login.css, register.css (стили для конкретных страниц).
* responsive.css (медиазапросы для адаптивности).
* hamburger.css (стили мобильного меню).
* scripts.js (основная логика (корзина, формы, модальные окна, слайдер)).
* mobile-menu.js (логика мобильного меню-гамбургера).

Так как проект использует ванильный JavaScript без объектно-ориентированного подхода, диаграмма классов заменена схемой структуры данных и функций (см. Приложение В).

## 3.3. Описание интерфейса пользователя

3.3.1. Принципы создания интерфейса

Интерфейс веб-сайта "SmileLand" разработан с учетом следующих принципов:

* простота и интуитивность: Четкая структура навигации, понятные CTA-кнопки и минималистичный дизайн упрощают взаимодействие.
* адаптивность: Поддержка устройств с разными разрешениями (от 480px до 1200px) через медиазапросы в responsive.css.
* визуальная привлекательность: Использование фирменных цветов, анимаций и эффектов (например, масштабирование изображений).
* доступность: Семантическая разметка и атрибуты alt для изображений улучшают поддержку экранных читалок.
* интерактивность: Быстрая реакция на действия пользователя (открытие корзины, валидация форм, анимации).

3.3.2. Основные приемы реализации интерфейса

Хедер присутствует на всех страницах и включает логотип, меню ("Главная", "О клинике", "Уход за собой / аксессуары") и кнопки "Мои заказы" и "Корзина" (см. рисунок 9 – скриншот хедера).

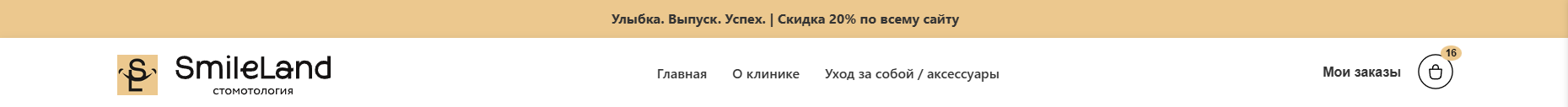


Рисунок 9 – Скриншот хедера

На экранах менее 768px активируется мобильное меню-гамбургер, реализованное через mobile-menu.js и стили в hamburger.css (см. рисунок 10 – скриншот мобильного меню).

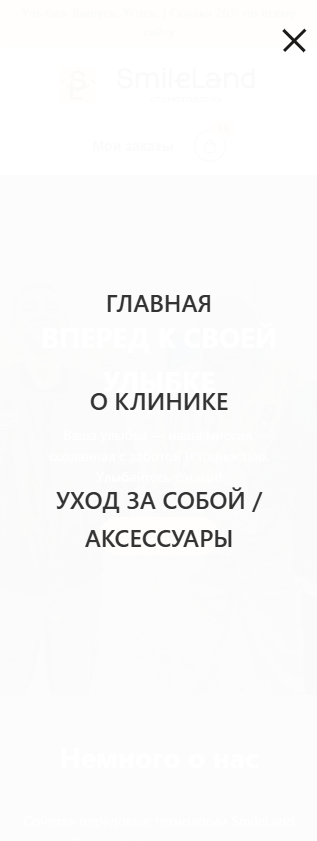


Рисунок 10 – Скриншот мобильного меню-гамбургера

Корзина. Отображается при клике на кнопку с иконкой корзины, содержит список товаров, их количество и общую сумму (см. рисунок 11 – скриншот корзины). Управляется через CSS-переходы (right: 0) и JavaScript (updateCart()).

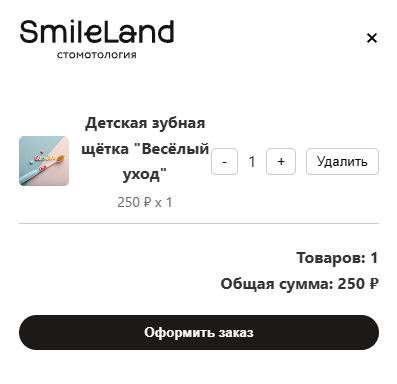


Рисунок 11 – Скриншот модального окна корзины

Промо-окно. Появляется после регистрации/авторизации или при клике на промо-ссылку, содержит форму подписки и анимацию появления (modalFadeIn) (см. рисунок 12 – скриншот промо-окна).

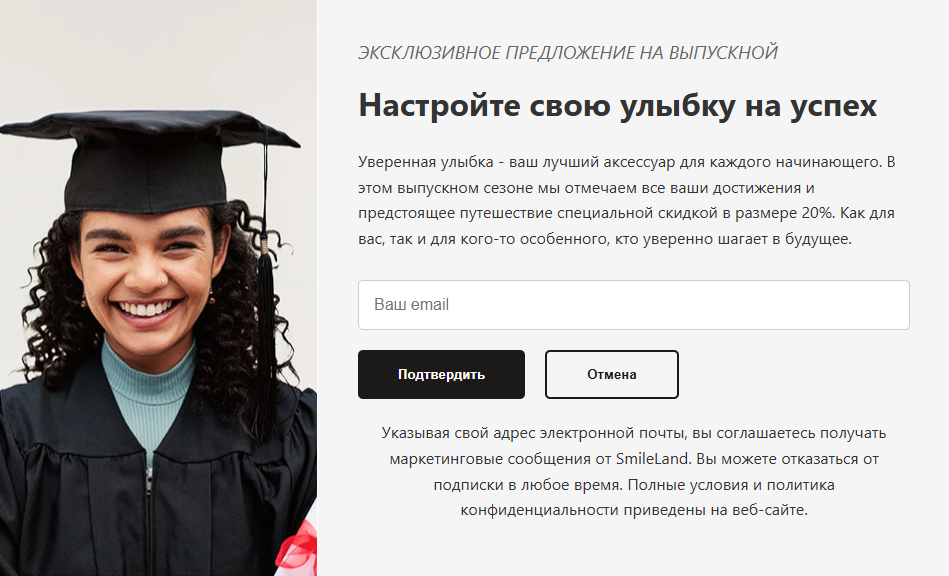


Рисунок 12 – Скриншот промо-окна

Окно ошибки. Отображается при попытке оформить заказ, имитируя серверную ошибку (см. рисунок 13 – скриншот окна ошибки).

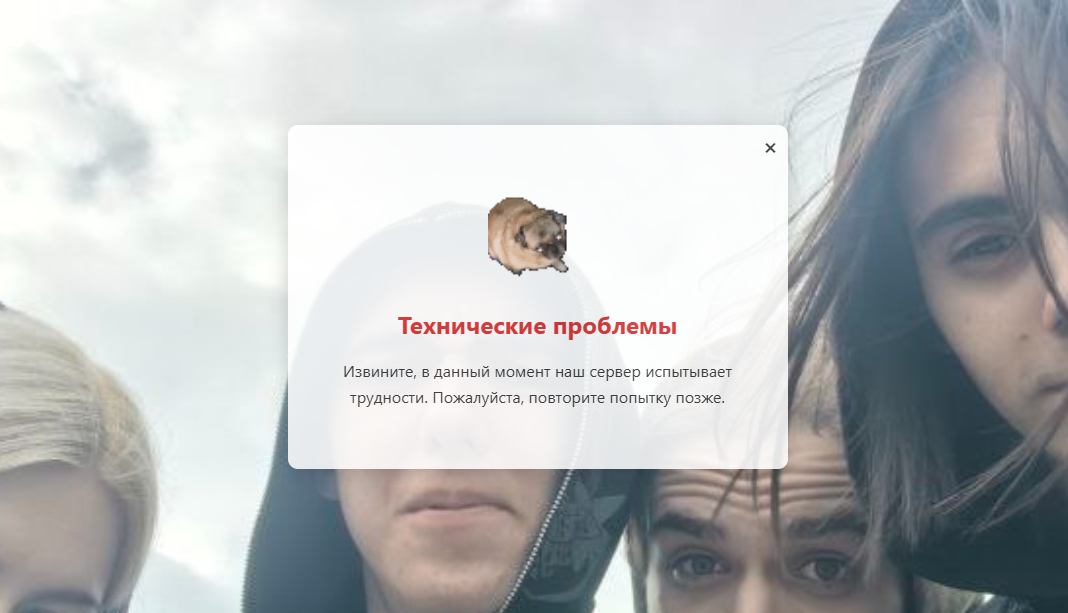


Рисунок 13 – Скриншот окна ошибки

Каталог товаров. Товары представлены в сетке (grid-template-columns) с изображениями, названием, рейтингом и ценой. Эффект масштабирования изображения при наведении (transform: scale(1.1)) и появление кнопки "Добавить в корзину" реализованы через CSS (catalog.css) (см. рисунок 14 – скриншот каталога).

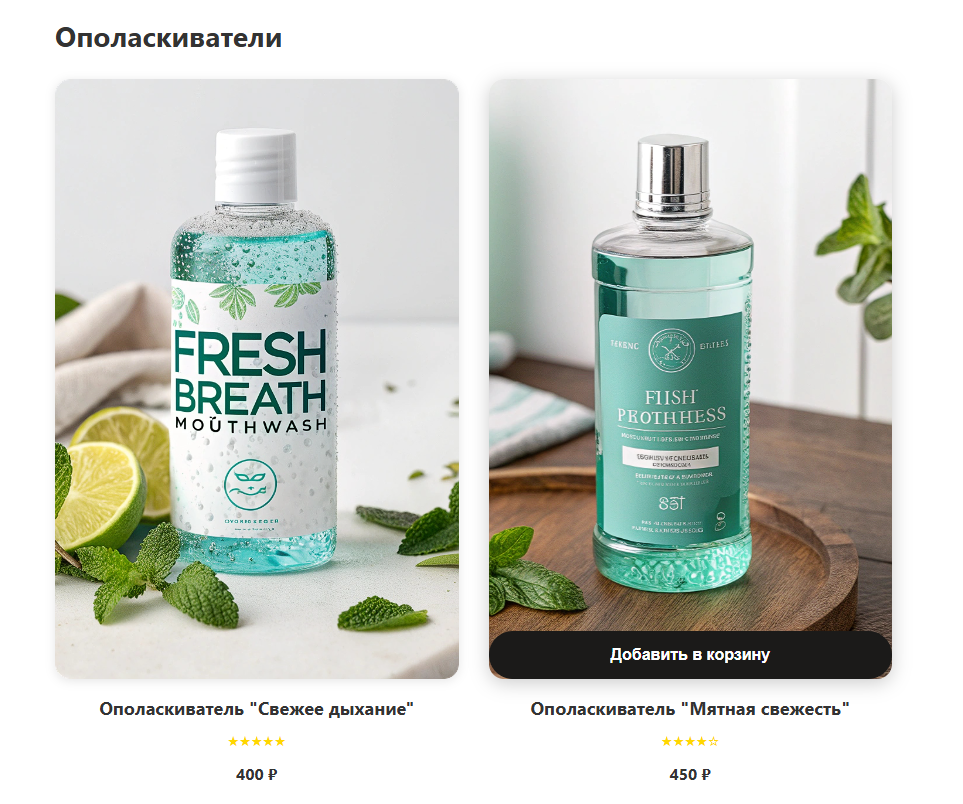


Рисунок 14 – Скриншот каталога товаров

Слайдер отзывов. Поддерживает сенсорное управление и навигацию через стрелки и миниатюры, реализованные через JavaScript и CSS-переходы (transform: translateX). Слайдер отображает три карточки отзывов одновременно на десктопе и одну на мобильных устройствах (см. рисунок 15 – скриншот слайдера).

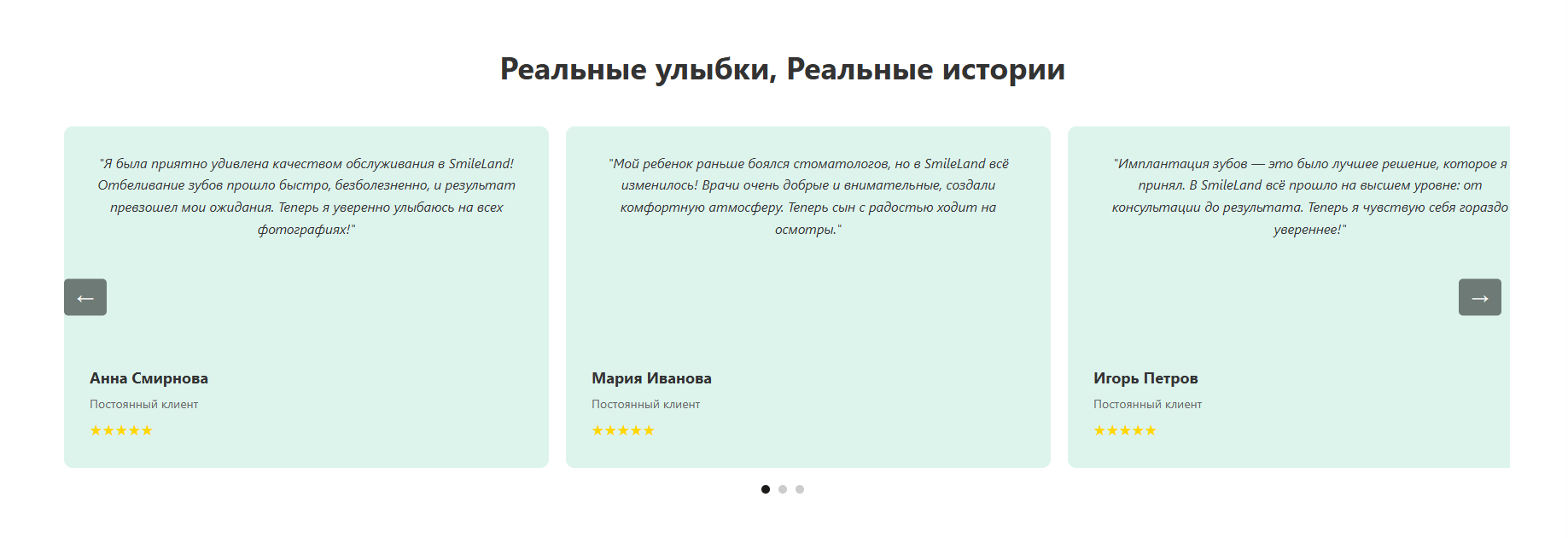


Рисунок 15 – Скриншот слайдера отзывов

Формы регистрации, авторизации и записи на прием имеют единообразный стиль с полями ввода, уведомлениями об успехе/ошибке и кнопкой отправки (см. рисунок 16 – скриншот формы регистрации).

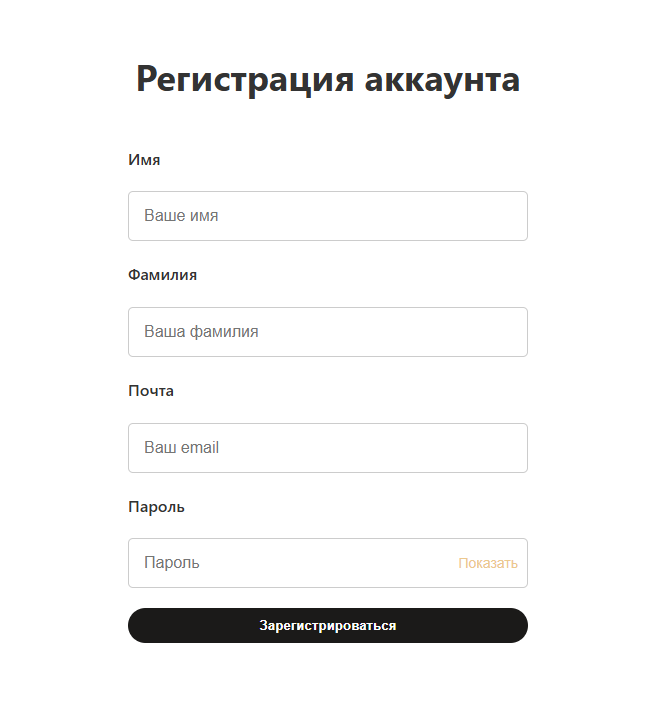


Рисунок 16 – Скриншот формы регистрации

Форма записи на прием сопровождается анимацией звездочек и конфетти при успешной отправке, что улучшает UX (см. рисунок 17 – скриншот анимации формы).

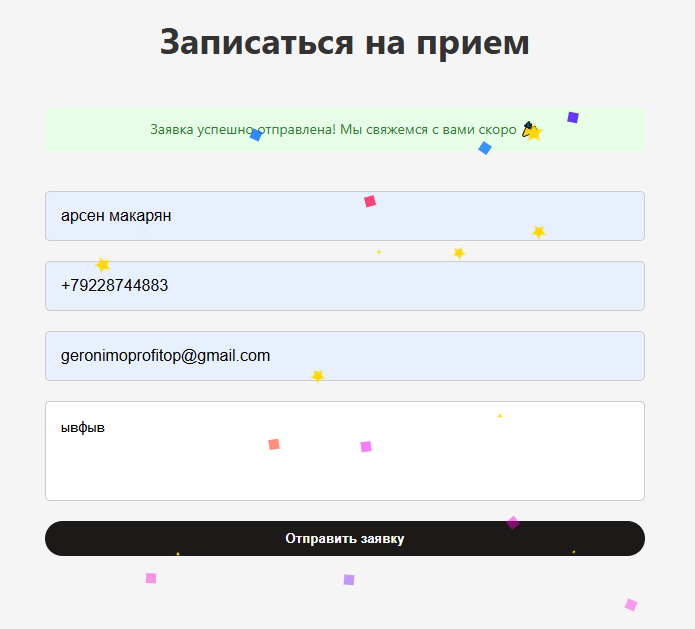


Рисунок 17 – Скриншот анимации формы записи на прием

Адаптивность. Медиазапросы в responsive.css обеспечивают корректное отображение на устройствах от 480px до 1200px (см. рисунок 18 – скриншот мобильной версии каталога). Мобильное меню и сенсорный слайдер оптимизированы для малых экранов, а элементы интерфейса масштабируются для удобства использования.

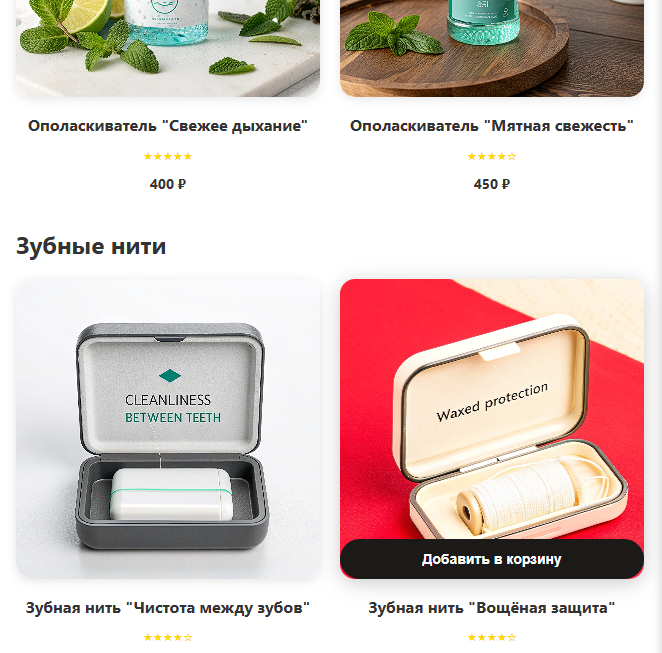


Рисунок 18 – Скриншот мобильной версии каталога.

# 4 Тестирование приложения

## 4.1 План тестирования

4.1.1. Цель и задачи тестирования

Тестирование веб-приложения "SmileLand" проводится с целью проверки его функциональности, надежности и соответствия требованиям. Основные задачи включают выявление ошибок в клиентской логике (управление корзиной, валидация форм, работа слайдера), проверку корректности хранения данных в localStorage и обеспечение адаптивности интерфейса. Тестирование охватывает unit-тесты для отдельных функций, функциональные тест-кейсы, интеграционное тестирование для проверки взаимодействия компонентов и приемочное тестирование для оценки пользовательского опыта.

4.1.2. Объекты тестирования

Объектами тестирования являются ключевые компоненты приложения. К клиентской логике относятся функции управления корзиной (добавление, удаление товаров, расчет суммы), валидация форм (регистрация, авторизация, запись на прием), отображение модальных окон (корзина, промо, ошибка) и работа слайдера отзывов с сенсорным управлением. Интерфейс тестируется на адаптивность (поддержка разрешений от 480px до 1200px) и корректность стилей. Хранилище данных проверяется на правильность сохранения и извлечения массива cart и флага showPromoModal в localStorage.

4.1.3. Виды тестирования

Для достижения целей применяются следующие виды тестирования. Unit-тестирование проверяет отдельные функции JavaScript, такие как saveCart(), updateCart() и validateForm(), с использованием фреймворка Jest. Интеграционное тестирование оценивает взаимодействие компонентов, например, добавление товара в корзину с обновлением DOM и localStorage. Приемочное тестирование, проводимое вручную, подтверждает соответствие приложения пользовательским требованиям, включая удобство навигации и интуитивность интерфейса. Каждый вид тестирования детализирован в соответствующих тест-кейсах и сценариях.

4.1.4. Unit-тестирование

Unit-тестирование проводится для проверки корректности отдельных функций в файле scripts.js с использованием Jest, настроенного с jsdom для симуляции браузерной среды. Тестируются функции saveCart() (сохранение корзины в localStorage), updateCart() (обновление DOM корзины), validateForm() (валидация форм) и togglePassword() (переключение типа поля пароля). Пример теста для saveCart(): входной массив cart = [{name: "Зубная паста", price: "150", image: "paste.jpg", quantity: 1}] должен сохраняться в localStorage как строка JSON. Тесты размещаются в директории \_\_tests\_\_ (см. рисунок 21 – пример unit-теста).

4.1.5. Интеграционное тестирование

Интеграционное тестирование проверяет взаимодействие компонентов приложения, таких как JavaScript, DOM и localStorage, с использованием Jest и jsdom. Тестируются два сценария: добавление товара с обновлением корзины и валидация формы с отображением промо-окна. Например, в первом сценарии проверяется, что вызов функции addToCart() обновляет localStorage и DOM-элементы .cart-items и .cart-total. Тесты включают симуляцию пользовательских действий и проверку состояния интерфейса (см. рисунок 25 – скриншот интеграционного теста).

## 4.2 Оценка результатов проведения тестирования

Тест-кейсы – это документы, описывающие условия и конкретные действия, необходимые для проверки функциональности программного обеспечения. Помогают убедиться, что приложение или система работает правильно и соответствует требованиям.

Основные элементы тест-кейса:

* идентификатор тест-кейса (уникальный номер или название тест-кейса для идентификации);
* краткое описание того, что тестируется и какие функции проверяются;
* предусловия (условия, которые должны быть выполнены до запуска теста);
* пошаговые инструкции, как выполнить тест, включая вводимые данные и ожидаемые результаты;
* ожидаемый результат (результат, который должен быть получен, если программное обеспечение работает корректно);
* фактический результат (результат, полученный в ходе тестирования, который может быть записан для анализа);
* статус (показывает, прошел тест успешно или нет).

Для сайта дизайна интерьеров были разработаны следующие тест-кейсы, представленные в таблице 1.

Таблица 1 – Тест-кейсы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Наименование | Входные данные | Ожид. Результат | Получ. Результат | Итог |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | Открытие сайта | Открыть сайт через поиск в браузере | Система открывает сайт и подгружает все содержимое корректно | Система открывает сайт и подгружает все содержимое корректно | Тест выполнен успешно |
| 2 | Проверка отображения главной страницы | Открыть сайт и прокрутить главную страницу вниз | Все элементы (блоки) отображаются корректно | Все элементы (блоки) отображаются корректно | Тест выполнен успешно |
| 3 | Переход на страницу «О компании» | Нажать на раздел «О компании» в меню навигации | Сайт переходит на страницу и корректно загружает содержимое | Сайт переходит на страницу и корректно загружает содержимое | Тест выполнен успешно |
| 4 | Переход на страницу «Галерея» | Нажать на раздел «Галерея» в меню навигации | Сайт переходит на страницу и корректно загружает все изображения | Сайт переходит на страницу и корректно загружает все изображения | Тест выполнен успешно |
| 5 | Проверка отправки формы обратной связи | Заполнить все поля формы и нажать отправить | Система подтверждает успешную отправку формы | Система подтверждает успешную отправку формы | Тест выполнен успешно |

# Заключение

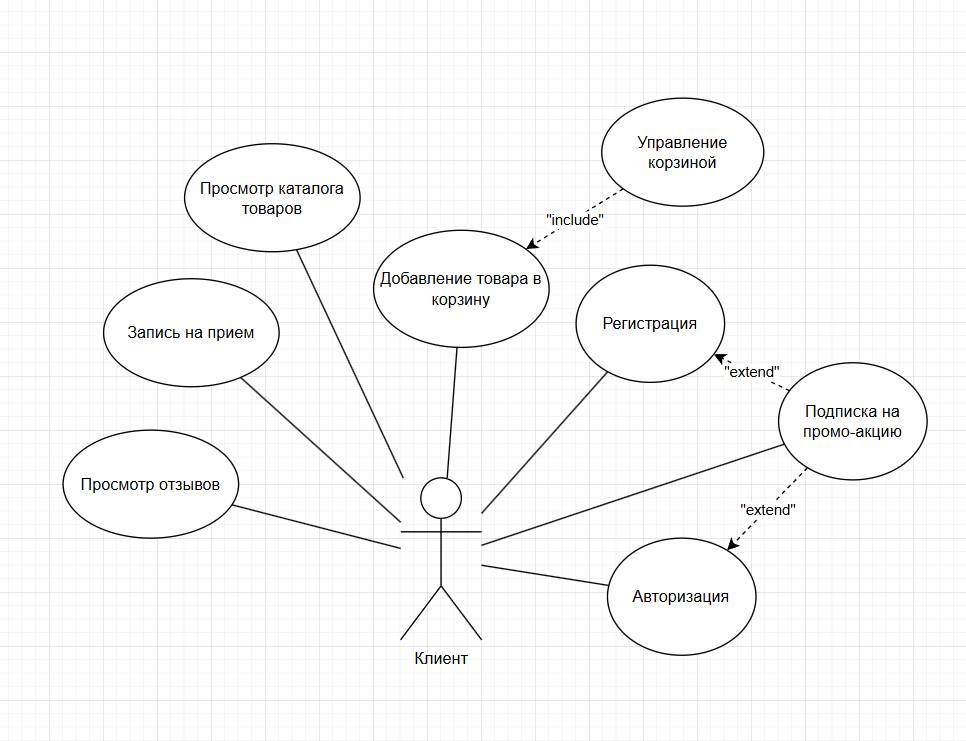
В рамках курсовой работы было разработано статическое веб-приложение "SmileLand" для стоматологической клиники, включающее функционал каталога товаров, корзины, форм регистрации, авторизации, записи на прием и слайдера отзывов. Основной целью проекта являлось создание интерактивного, адаптивного и легковесного сайта с использованием HTML, CSS, JavaScript и localStorage. Поставленные задачи, включая реализацию клиентской логики, адаптивного интерфейса и тестирования, были выполнены в полном объеме. Разработанное приложение соответствует заявленным требованиям, обеспечивая удобство использования и быструю загрузку страниц благодаря отсутствию серверных зависимостей.

Функционал приложения реализован в соответствии с техническим заданием. Каталог товаров позволяет пользователям просматривать продукты и добавлять их в корзину, которая сохраняется в localStorage и обновляет интерфейс в реальном времени. Формы регистрации, авторизации и записи на прием включают валидацию с использованием регулярных выражений и анимации для улучшения пользовательского опыта. Слайдер отзывов поддерживает сенсорное управление, а интерфейс адаптирован для устройств с разрешениями от 480px до 1200px. Тестирование (unit-тесты, функциональные тест-кейсы, интеграционное и приемочное тестирование) подтвердило корректность работы приложения, а выявленные ошибки, такие как некорректный расчет суммы в корзине и проблемы с анимацией на мобильных устройствах, были устранены.

В процессе разработки возникли определенные сложности. Наиболее значимыми оказались настройка Jest для unit- и интеграционного тестирования в среде jsdom, так как проект использует ванильный JavaScript без фреймворков, что потребовало ручной симуляции DOM. Оптимизация анимации конфетти для мобильных устройств также вызвала затруднения из-за различий в производительности браузеров. Кроме того, ограничения localStorage (объем до 5–10 МБ и отсутствие серверной обработки) затрудняли реализацию сложных функций, таких как обработка заказов. Эти проблемы были решены путем упрощения анимаций, добавления обработки исключений для localStorage и тщательной отладки тестов.

Дальнейшее развитие приложения может включать несколько направлений. Во-первых, интеграция серверной части (например, с использованием Node.js и базы данных MongoDB) позволит реализовать полноценную обработку заказов, авторизацию с хранением пользовательских данных и отправку уведомлений. Во-вторых, добавление CMS (например, WordPress) или админ-панели упростит управление каталогом товаров и отзывами. В-третьих, расширение функционала, такого как фильтрация товаров по категориям, интеграция с платежными системами (Stripe, PayPal) или чат-бот для консультаций, повысит привлекательность сайта. Наконец, оптимизация производительности (например, ленивая загрузка изображений) и улучшение доступности (поддержка экранных читалок) сделают приложение более конкурентоспособным.

В заключение, разработка и тестирование веб-приложения "SmileLand" продемонстрировали возможность создания функционального и адаптивного сайта с использованием минимального технологического стека. Несмотря на сложности с тестированием и ограничениями localStorage, поставленные цели достигнуты, а приложение готово к использованию в текущем виде. Опыт, полученный в ходе работы, включая написание тестов, оптимизацию анимаций и проектирование интерфейса, будет полезен для дальнейших проектов. Перспективы развития приложения открывают возможности для его масштабирования и интеграции с современными веб-технологиями.



# Список используемых источников

* 1. ГОСТ Р ИСО/МЭК 12207-2017 «Жизненный цикл программного обеспечения. Процессы, мероприятия и задачи».
  2. ГОСТ Р 50597-2018 «Системы управления информационной безопасностью. Общие требования».
  3. ГОСТ Р 51522-2020 «Информационные технологии. Порядок проектирования интерфейсов для программных средств».
  4. ГОСТ Р 51901.1-2019 «Информационные технологии. Безопасность информации. Общие требования к безопасности информации в автоматизированных системах».
  5. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных».
  6. Федеральный закон от 29.07.2004 № 98-ФЗ «Об электронной подписи».
  7. Бочаров А.А. «Тестирование информационных систем: от теории к практике». - М.: Альпина Паблишер, 2015. - 256 с.
  8. Иванов И.И. «Введение в разработку информационных систем». - М.: Научное издательство, 2015.
  9. Кравченко С.И. «Моделирование и разработка информационных систем». - Ростов-на-Дону: Феникс, 2019. - 280 с.
  10. Николаев С.А. «Управление IT-проектами: учебное пособие». - М.: Инфра-М, 2019. - 368 с.
  11. Петрова Н.Н. «Технологии разработки веб-приложений». - СПб.: Питер, 2019. - 312 с.
  12. Рогов Е.Н. «Основы автоматизированного проектирования информационных систем». - М.: Юрайт, 2020. - 352 с.
  13. Сидоров А.А. «Программирование для бизнеса». - М.: Эксмо, 2017.
  14. Федоренко М.Л., Иванов А.Н. «Резервное копирование и восстановление данных: теория и практика». - М.: Наука и Техника, 2019. - 216 с.
  15. Шилов А.В. «Системный анализ и проектирование информационных систем». - М.: Академический проект, 2020. 400 с.
  16. D. J. Perry. Web Development with JavaScript. New York: Wiley, 2018.
  17. R. W. Roberts. Software Engineering for Web Applications. London: Springer, 2020.
  18. Электронный ресурс: Stack Overflow, https://stackoverflow.com [Электронный ресурс], – Режим доступа: https://stackoverflow.com/questions/1234567/what-is-the-best-way-to-structure-a-web-application – дата использования: 09.12.2024.
  19. Электронный ресурс: W3C, https://www.w3.org [Электронный ресурс], – Режим доступа: https://www.w3.org/standards/webdesign – дата использования: 14.12.2024.
  20. Электронный ресурс: GitHub, https://github.com [Электронный ресурс], – Режим доступа: https://github.com/topics/web-development – дата использования: 11.12.2024.
  21. Электронный ресурс: Mozilla Developer Network (MDN), https://developer.mozilla.org [Электронный ресурс], – Режим доступа: https://developer.mozilla.org/en-US/docs/Web/HTML – дата использования: 19.12.2024.
  22. Электронный ресурс: WebDev, https://webdev.ru [Электронный ресурс], – Режим доступа: https://webdev.ru/blog/ – дата использования: 17.12.2024.
  23. Электронный ресурс: Техномагия, https://techno-magia.ru [Электронный ресурс], – Режим доступа: https://techno-magia.ru/articles/web-development – дата использования: 18.12.2024.
  24. Электронный ресурс: Хабр, https://habr.com [Электронный ресурс], – Режим доступа: https://habr.com/ru/post/123456/ – дата использования: 16.12.2024.
  25. Электронный ресурс: Dev.to, https://dev.to [Электронный ресурс], – Режим доступа: https://dev.to/ru/ – дата использования: 13.12.2024.