**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc201748490)

[1 АНАЛИЗ ПРОИЗВОДСТВЕННОГО ПРЕДПРИЯТИЯ 4](#_Toc201748491)

[1.1 Описание предметной области 4](#_Toc201748492)

[1.2 Сравнительный анализ систем-аналогов 6](#_Toc201748493)

[1.3 Выбор программных средств для разработки 9](#_Toc201748494)

[1.4 Выводы по первой главе 15](#_Toc201748495)

[2 СРЕДСТВА РЕАЛИЗАЦИИ ДИПЛОМНОЙ РАБОТЫ 16](#_Toc201748496)

[2.1 Обзор инструментальных средств создания продукта 16](#_Toc201748497)

[2.2 Обоснование выбора средств реализации дипломной работы 19](#_Toc201748498)

[3 СТРУКТУРА И РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 27](#_Toc201748499)

[3.1 Разработка интерфейса программного продукта 27](#_Toc201748500)

[3.2 Рекомендуемые системные требования 34](#_Toc201748501)

[3.3 Обеспечение безопасности информации 37](#_Toc201748502)

[3.4 Определение экономической эффективности внедрения программного продукта 39](#_Toc201748503)

[3.5 Выводы по третьей главе 42](#_Toc201748504)

[ЗАКЛЮЧЕНИЕ 44](#_Toc201748505)

[СПИСОК ЛИТЕРАТУРЫ 46](#_Toc201748506)

[ПРИЛОЖЕНИЕ А 49](#_Toc201748507)

[ПРИЛОЖЕНИЕ Б 51](#_Toc201748508)

# **ВВЕДЕНИЕ**

В современном мире развитие информационных технологий оказало значительное влияние на все сферы жизни, включая здравоохранение. Одной из ключевых задач, стоящих перед медицинскими учреждениями, является повышение доступности и качества медицинских услуг. В этой связи создание и функционализация сайтов поликлиник становятся особенно актуальными. Разработка удобного и информативного веб-ресурса для поликлиники может значительно упростить процесс получения медицинской помощи, сделав его более доступным как для пациентов, так и для медицинского персонала.

Актуальность темы нашей выпускной квалификационной работы заключается в необходимости повышения качества медицинского обслуживания через интеграцию информационных технологий в деятельность поликлиник. Сегодня пациенты все чаще сталкиваются с проблемами при записи на прием к врачу, получения информации о порядке предоставления услуг и получении доступной медицинской информации. Разработка сайта поликлиники, который будет учитывать потребности пользователей и отвечать современным требованиям, станет важным шагом в решении упомянутых проблем.

Новизна нашего проекта заключается не только в создании визуально привлекательного и функционального интерфейса, но и в внедрении интерактивных элементов, таких как онлайн-запись на прием, возможность получения информации о врачах и услугах, а также интеграция с актуальными системами здравоохранения. Это позволит пациентам сократить время ожидания и значительно упростить взаимодействие с поликлиникой.

Практическая ценность данной работы заключается в том, что созданный сайт может быть использован как модель для других медицинских учреждений, стремящихся улучшить свои сервисы. Кроме того, разработка веб-ресурса может повысить прозрачность работы поликлиники, улучшить коммуникацию между пациентами и врачами, а также обеспечить доступ к полезной информации о здоровье и лечении.

Выполнение данной работы обосновано запросами общества на улучшение медицинских услуг и доступ к ним через современные технологии. Реализация компетентного, удобного и безопасного сайта для поликлиники станет важным вкладом в развитие медицинского сервиса и повысит уровень удовлетворенности пациентов. Таким образом, разработка сайта для поликлиники является важной и необходимой задачей, которая будет способствовать улучшению системы здравоохранения в целом.

# **1 АНАЛИЗ ПРОИЗВОДСТВЕННОГО ПРЕДПРИЯТИЯ**

# **Описание предметной области**

Предметная область данной выпускной квалификационной работы охватывает сферу разработки веб-сайтов для поликлиник, что является важным аспектом современного здравоохранения. В последние десятилетия наблюдается значительный рост интереса к использованию информационных технологий в медицине, что связано с необходимостью оптимизации процессов обмена информации между пациентами и медиками, а также с повышением доступности медицинских услуг для населения.

Поликлиники, как важные элементы системы здравоохранения, обеспечивают первичную медицинскую помощь и услуги, включая диагностику, лечение и профилактику заболеваний. В современных условиях, когда пациенты стремятся к максимальному удобству и быстроте в получении медицинских услуг, создание сайта для поликлиник становится особенно актуальным. Такой ресурс позволяет не только обеспечить доступ к информации о предоставляемых услугах, но и реализовать возможности для удаленной записи на прием, упрощая процесс взаимодействия между пациентами и врачами.

На сайте поликлиники должно быть представлено множество разделов, таких как информация о врачах и медицинском персонале, расписание работы, описание услуг и процедур, а также возможность получения медицинских рекомендаций и справочной информации. Важным аспектом является также внедрение функций, позволяющих пользователям оставлять отзывы о качестве услуг, что повысит уровень прозрачности и доверия к медицинскому учреждению. В эпоху цифровизации критически важно учитывать и безопасность данных пациентов. Соответствие нормативным требованиям защиты информации является неотъемлемой частью разработки сайта.

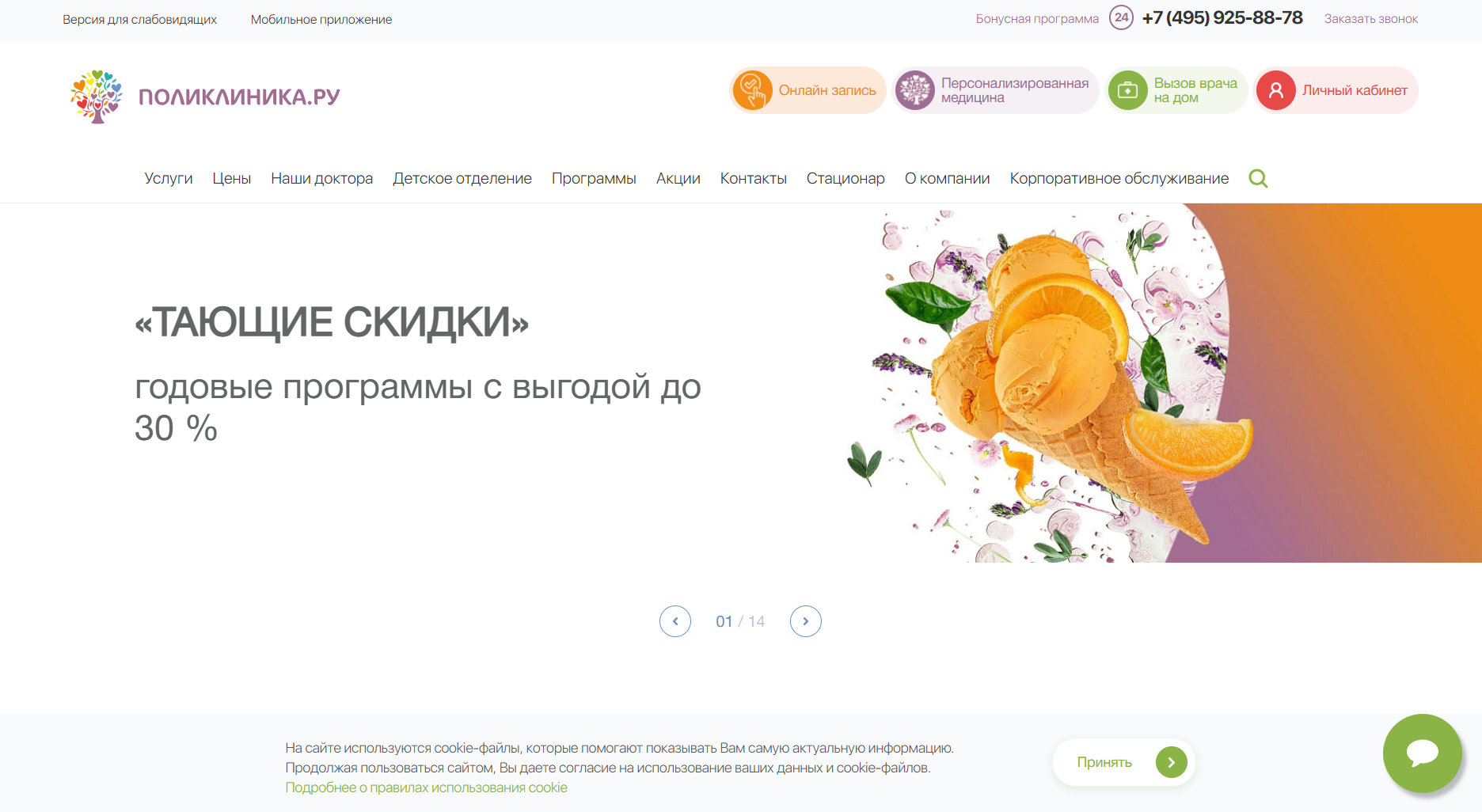
Современные тенденции в мире веб-разработки также оказывают влияние на создание сайтов поликлиник. Адаптивный дизайн, наличие мобильной версии, быстрая загрузка страниц и высокий уровень юзабилити становятся ключевыми факторами, определяющими успешность веб-ресурса. Кроме того, интеграция с системами электронной очереди и управления записями способствует улучшению организации работы поликлиники и снижению времени ожидания для пациентов.

В заключение, предметная область разработки сайта для поликлиники представляет собой пересечение медицины и информационных технологий, фокусируясь на улучшении взаимодействия между пациентами и медицинскими учреждениями. В условиях постоянного роста потребностей населения в высококачественном медицинском обслуживании создание эффективного веб-ресурса становится не просто желанием, а необходимостью для каждого медицинского учреждения, стремящегося повысить свой уровень сервиса и удовлетворенность пациентов.

# **Сравнительный анализ систем-аналогов**

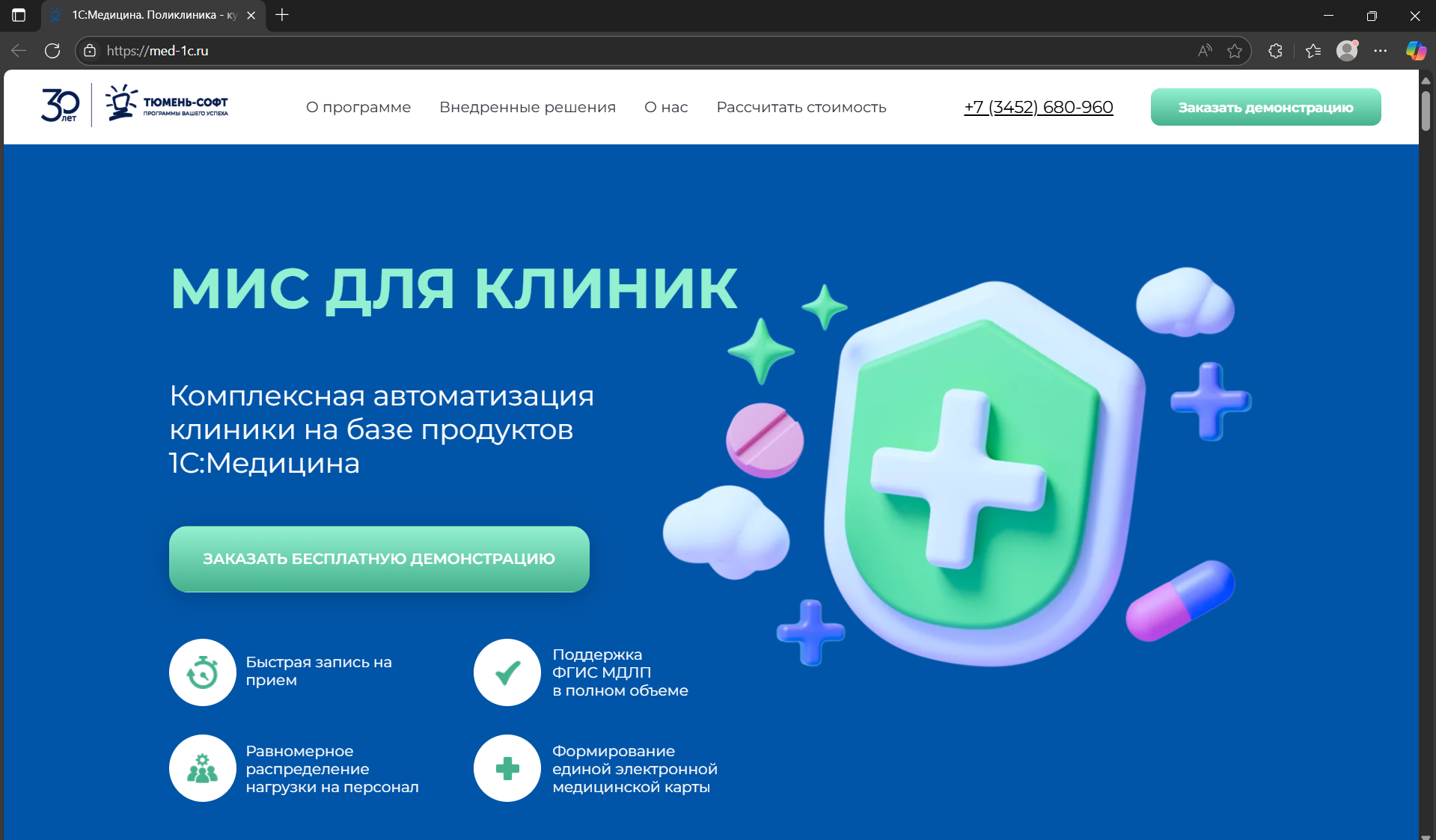
В процессе разработки сайта для поликлиники важно провести сравнительный анализ существующих систем-аналогов, что позволит выявить как достоинства, так и недостатки различных платформ и решений, используемых в данной области. На сегодняшний день существует множество веб-ресурсов, предназначенных для автоматизации работы поликлиник, однако их функциональные возможности и пользовательский опыт могут значительно отличаться.

Одним из наиболее известных решений является система «Поликлиника.ру» (рисунок 1). Этот ресурс позволяет пользователям записываться на прием к врачам, получать результат анализов и управлять медицинской документацией. Преимуществами данной платформы являются интуитивно понятный интерфейс и возможность интеграции с различными системами лабораторных исследований. Однако стоит отметить и недостатки, такие как ограниченные возможности персонализации для конкретной поликлиники и необходимость оплаты за использование некоторых функций.



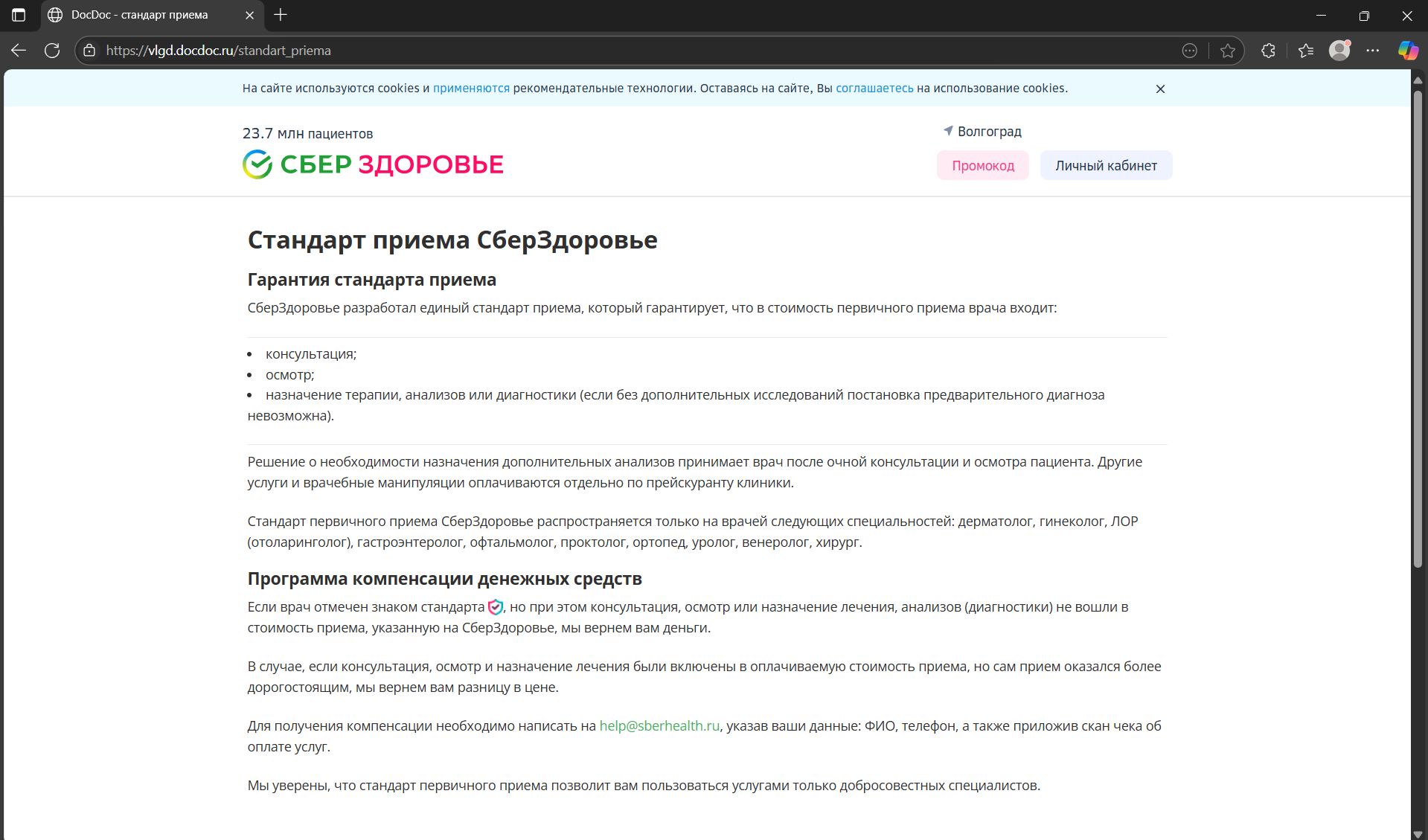
1. Сайт Поликлиника.ру

Система «1С:Медицинская Практика» (рисунок 2) представляет собой комплексное решение для управления медицинскими учреждениями. Она включает в себя функции планирования, учета пациентского потока и автоматизации процессов документооборота. Основным преимуществом этой системы является ее масштабируемость и возможность интеграции с другими программными продуктами 1С, что делает ее подходящей для крупных учреждений. Однако для малых и средних поликлиник использование данной платформы может быть неподъемным из-за высокой стоимости внедрения и поддержки.



1. 1С:Медицинская Практика

Еще одним интересным вариантом является система «DocDoc» (рисунок 3), которая акцентируется на поиске и выборе врачей по отзывам и рейтингам пользователей. Эта платформа позволяет не только записываться на прием, но и сравнивать врачей по различным критериям. Однако, несмотря на удобство, она не всегда предоставляет полное представление о качестве медицинских услуг, так как полагается на субъективные оценки пациентов.



1. Сервис DocDoc

В заключение, можно сказать, что выбор системы для разработки веб-сайта поликлиники зависит от конкретных потребностей и ресурсов учреждения. Каждое решение имеет свои сильные и слабые стороны, и важным шагом в процессе выбора является тщательное изучение функциональных возможностей, интерфейса и экономической целесообразности. Успешная реализация проекта требует глубокого понимания системы и способных адаптировать ее под индивидуальные нужды, что в конечном счете скажется на качестве оказываемых медицинских услуг и удовлетворенности пациентов.

# **Выбор программных средств для разработки**

Разработка сайта для поликлиники представляет собой многоэтапный процесс, в котором важную роль играют выбор и использование соответствующих программных средств. Правильный выбор инструментов позволяет создать эффективный, удобный и безопасный ресурс, который будет соответствовать современным требованиям пользователей и актуальным стандартам в области медицины. В данном разделе мы рассмотрим ключевые категории программных средств, используемых в процессе создания веб-сайта для поликлиники.

На первом этапе разработки необходимо выбрать язык программирования и фреймворк, на котором будет построен сайт. Одним из самых популярных языков для веб-разработки является JavaScript[9], в сочетании с фреймворками, такими как React или Vue.js. Эти инструменты предоставляют широкие возможности для создания динамичных и отзывчивых интерфейсов. Использование таких библиотек позволяет не только ускорить процесс разработки, но и улучшить взаимодействие с пользователем благодаря высокой производительности.

Для бэкенд-разработки одним из наиболее распространенных выборов является использование языка PHP[7] вместе с фреймворком Laravel. Это решение обеспечивает создание мощных и безопасных серверных приложений, а также упрощает работу с базами данных. Laravel предлагает множество встроенных функций, таких как маршрутизация, управление сессиями и отладка, что значительно облегчает процесс разработки. Альтернативным вариантом может быть использование Python и фреймворка Django, который обеспечивает разработку безопасных и масштабируемых приложений с высоким уровнем производительности.

Следующим шагом является выбор системы управления базами данных (СУБД). В контексте разработки сайта для поликлиники оптимальным решением может быть использование реляционных баз данных, таких как MySQL[10] или PostgreSQL[10]. Эти СУБД обеспечивают надежное хранение данных о пациентах, медицинских записях и записях на прием. Также стоит рассмотреть NoSQL[10] решения, такие как MongoDB[10], которые могут быть полезны для обработки больших объемов различных типов данных.

Не менее важным аспектом является выбор инструментов для проектирования пользовательского интерфейса. Для этого можно использовать утилиты, такие как Figma или Adobe XD, которые позволяют создать прототипы и макеты интерфейса. Эти инструменты предоставляют возможность визуально представить структуру сайта, проработать пользовательский опыт и повысить его удобство, что особенно важно для сайтов медицинских учреждений.

Для обеспечения безопасности данных, а также хранения учетных записей пользователей необходимо обратить внимание на инструменты аутентификации и авторизации. В этом плане подходящими решениями могут стать OAuth 2.0 или JWT (JSON Web Tokens). Эти протоколы позволяют безопасно управлять доступом к данным, что особенно актуально в контексте работы с личной информацией пациентов.

Наконец, нельзя забывать о тестировании и оптимизации сайта. Инструменты, такие как Selenium или Jest, помогут автоматизировать процесс тестирования приложения, обеспечивая его стабильную работу при различных сценариях использования. Использование Google Lighthouse позволит анализировать производительность, доступность и SEO-оптимизацию сайта, что в свою очередь поможет в достижении высоких позиций в результатах поисковых систем.

После проведения анализа можно сделать вывод, что Visual Studio Code является оптимальным выбором в качестве среды разработки для сайта поликлиники по нескольким ключевым причинам.

Во-первых, Visual Studio Code — это легковесный и многофункциональный редактор кода, который поддерживает множество языков программирования, включая HTML[8], CSS[8], JavaScript, PHP и Python. Это делает его идеальным инструментом для разработки как фронтенд, так и бэкенд части сайта, позволяя разработчикам работать с разными технологиями в одном интерфейсе.

Во-вторых, Visual Studio Code предлагает обширный набор плагинов и расширений, что позволяет кастомизировать среду под конкретные задачи. Можно установить расширения для работы с системами контроля версий (например, Git), интеграции с базами данных, поддержки фреймворков и библиотек, а также инструменты для работы с API. Это значительно ускоряет процесс разработки и упрощает workflow.

Третьим аспектом является интуитивно понятный интерфейс и наличие мощных инструментов для отладки. Встроенная поддержка отладки позволяет разработчикам эффективно находить и устранять ошибки в коде, что критически важно для создания надежного программного обеспечения, особенно в контексте работы с медицинскими данными.

Четвертым важным преимуществом является возможность работы с удаленными проектами через SSH или другие протоколы. Это позволяет команде разработчиков работать совместно над проектом, независимо от их физического местоположения, что особенно актуально в современных условиях удаленной работы.

Наконец, Visual Studio Code активно поддерживается сообществом и регулярно обновляется, что гарантирует наличие современных инструментов и технологий, необходимых для разработки актуальных веб-решений. Широкая база знаний и множество обучающих материалов помогут разработчикам быстро освоить среду и эффективно использовать её возможности.

Таким образом, благодаря своей гибкости, функциональности и поддержке современных технологий, Visual Studio Code является лучшим решением для разработки сайта поликлиники, обеспечивая высокую продуктивность, удобство работы и надежность конечного продукта.

После анализа можно выделить несколько причин, почему HTML и CSS являются лучшими инструментами для разработки фронтенда сайта поликлиники.

Во-первых, HTML (HyperText Markup Language)[8] — это основополагающий язык разметки для веб-страниц. Он обеспечивает структуру контента, позволяя разработчикам создавать семантически корректные страницы, которые легко воспринимаются пользователями и поисковыми системами. Четкая структура важна для сайтов медицинских учреждений, где понятность информации и доступность услуг имеют первостепенное значение.

Во-вторых, CSS (Cascading Style Sheets)[9] — это язык стилевого оформления, который позволяет отделить содержание от представления. С помощью CSS разработчики могут стилизовать элементы HTML, обеспечивая удобочитаемость и привлекательный визуальный дизайн. Это особенно важно для сайта поликлиники, так как эстетика и удобство интерфейса могут повысить уровень комфорта пользователей.

Третьим важным аспектом является кросс-браузерная совместимость. HTML и CSS поддерживаются всеми современными браузерами, что гарантирует, что сайт будет корректно отображаться на различных устройствах и платформах, включая мобильные телефоны и планшеты. Это критично для пользователей, которые могут записываться на прием или искать информацию в дороге.

Также стоит отметить, что HTML и CSS обеспечивают высокую производительность и быстрые скорости загрузки сайтов. Разработка на этих языках позволяет создавать легкие и быстрые страницы, что особенно важно для поддержания интереса пользователей и снижения показателя отказов.

Еще одним преимуществом является легкость освоения. Для начинающих разработчиков HTML и CSS являются отличной отправной точкой, поскольку они имеют простой синтаксис и множество ресурсов для изучения. Это позволяет команде разработчиков быстро наладить работу и в дальнейшем интегрировать более сложные технологии, если это потребуется.

Наконец, HTML и CSS идеально подходят для создания отзывчивых интерфейсов благодаря использованию медиа-запросов и фреймворков, таких как Bootstrap. Это обеспечивает оптимальное отображение сайта на экранах различных размеров, позволяя пользователям удобно взаимодействовать с ресурсом с любого устройства.

В итоге, HTML и CSS — это надежные, эффективные и простые в использовании инструменты для разработки фронтенда сайта поликлиники, что делает их лучшим выбором для создания качественного и доступного веб-ресурса.

После анализа можно выделить несколько причин, почему PHP и MAMP являются оптимальным выбором для разработки серверной части сайта и администрирования базы данных.

Во-первых, PHP (Hypertext Preprocessor)[9] — это один из самых популярных языков программирования для веб-разработки. Он идеально подходит для создания динамических веб-страниц и обработки данных на сервере. PHP легко интегрируется с HTML, что позволяет разработчикам создавать интерактивные сайты с высокой функциональностью, что особенно важно для сайта поликлиники, где требуется обработка форм и взаимодействие с пользователями.

Во-вторых, PHP обладает широкой поддержкой и сообществом. Существуют обширные библиотеки и фреймворки, такие как Laravel и Symfony, которые ускоряют процесс разработки и позволяют использовать проверенные решения. Это значит, что команда разработчиков может быстрее решать задачи и внедрять новые функции.

Третьим важным аспектом является возможность простого подключения к различным базам данных, таким как MySQL, PostgreSQL и другими. PHP отлично работает с MySQL, что делает его идеальным для создания и управления базами данных, необходимых для хранения информации о пациентах, расписаниях врачей и других данных.

MAMP (Macintosh, Apache, MySQL, PHP)[9] предоставляет разработчикам удобный локальный сервер для разработки. MAMP позволяет легко устанавливать и настраивать веб-сервер с поддержкой PHP и MySQL на компьютерах Mac, что значительно упрощает процесс разработки. Он предоставляет простые инструменты для управления базами данных через интерфейс phpMyAdmin, позволяя быстро выполнять операции с базами данных без необходимости написания SQL-запросов через командную строку.

Также стоит отметить, что использование MAMP позволяет разработчикам легко тестировать и отлаживать код на локальном сервере перед развертыванием на удаленном. Это снижает количество ошибок и повышает качество конечного продукта.

Наконец, PHP и MAMP прекрасно подходят для создания отзывчивых и масштабируемых приложений. Это важно для поликлиники, где может потребоваться добавление новых функций или расширение системы для обслуживания большего количества пользователей с течением времени.

В итоге, PHP и MAMP составляют мощную связку для написания серверной части сайта и администрирования базы данных, обеспечивая разработчиков необходимыми инструментами, функциональностью и удобством работы.

# **Выводы по первой главе**

В первой главе проведен детальный анализ производственного предприятия, что является основой для дальнейшего понимания его функционирования и разработки программного обеспечения. Начнем с описания предметной области, где рассматриваются ключевые аспекты производственного процесса, специфика бизнеса, а также текущие тенденции и проблемы, с которыми сталкивается предприятие. Это позволяет выявить основные требования к системе и определить ее задачи.

Далее в главе представлен сравнительный анализ систем-аналогов. В этом разделе проанализированы существующие решения, их преимущества и недостатки, что позволяет понять, какие функции критически важны для новой системы, а какие можно улучшить. Сравнительный анализ помогает определить лучшие практики и вдохновиться успешными примерами, а также избежать потенциальных ошибок.

Завершающим элементом первой главы является выбор программных средств для разработки. На основании предшествующего анализа сделан выбор технологий и инструментов, которые наилучшим образом соответствуют требованиям производственного предприятия. Учтены как функциональные возможности, так и аспекты удобства использования, совместимости и поддержки.

В итоге, первая глава является фундаментом для последующих разделов работы, предоставляя всесторонний анализ предметной области, что позволит создать качественное и эффективное программное обеспечение, соответствующее потребностям предприятия.

# **2 СРЕДСТВА РЕАЛИЗАЦИИ ДИПЛОМНОЙ РАБОТЫ**

# **Обзор инструментальных средств создания продукта**

Обзор инструментальных средств создания продукта содержит анализ концепций, которые можно использовать для разработки сайта поликлиники. В процессе создания веб-приложений разработчики могут выбрать различные подходы, каждый из которых имеет свои достоинства и недостатки.

Первой концепцией является монолитная архитектура. В этом подходе все компоненты приложения разрабатываются и разворачиваются как единое целое. Достоинства этой концепции заключаются в простоте разработки и тестирования, а также в том, что все функциональные элементы находятся в одном месте, что упрощает поддержку и обновление. Однако у монолитной архитектуры есть и недостатки: сложности с масштабированием, особенно при увеличении нагрузки, а также трудности в усовершенствовании системы без модификации всего приложения.

Следующей концепцией является микросервисная архитектура. В этом подходе приложение разбивается на множество небольших, независимо развиваемых и разворачиваемых сервисов. Это позволяет командам работать параллельно над различными частями системы, что ускоряет разработку и внедрение новых функций. Однако внедрение микросервисов требует более сложной инфраструктуры и может привести к увеличению затрат на поддержку и управление взаимодействием между сервисами.

Третьей концепцией является использование фреймворков и CMS, таких как Laravel, Symfony или WordPress. Эти платформы предоставляют множество готовых инструментов и библиотек, которые значительно ускоряют процесс разработки. Преимущества включают в себя скорость разработки, наличие встроенных функций безопасности и поддержка сообщества. Однако использование фреймворков может ограничивать гибкость и зависеть от обновлений и поддержки третьих сторон.

Также стоит рассмотреть концепцию разработки на основе API. Создание RESTful или GraphQL API позволяет легко интегрировать различные клиентские приложения, такие как мобильные приложения или сторонние сервисы. Это делает систему более гибкой и расширяемой. Однако разработка API требует дополнительных усилий по проектированию и документированию интерфейсов.

Еще одной важной концепцией является агильная методология разработки, которая предполагает итеративный подход к созданию продукта. Эта методология включает в себя регулярные проверки и изменения в процессе разработки с учетом обратной связи от пользователей. Достоинства включают в себя повышенную гибкость и возможность быстрого реагирования на изменения требований. Однако агильная разработка может привести к недостаточной документации, если команда не будет внимательно следить за процессом.

В заключение, каждая из представленных концепций создания программного продукта имеет свои преимущества и недостатки. Выбор подхода зависит от конкретных требований проекта, бюджета, сроков и технической квалификации команды. Для разработки сайта поликлиники целесообразно рассмотреть использование микросервисной архитектуры или фреймворков, чтобы обеспечить масштабируемость и быструю возможность внедрения новых функций, что крайне важно в быстро меняющемся медицинском секторе.

Однако создание сайта с клиентской и серверной частями представляет собой оптимальный подход для разработки современных веб-приложений, поскольку он обеспечивает гармоничную работу и взаимодействие пользователей с платформой. Во-первых, такая архитектура позволяет разделить обязанности между клиентом и сервером, что способствует более эффективному управлению ресурсами. Клиентская часть отвечает за пользовательский интерфейс и взаимодействие с пользователем, что позволяет создавать интерактивные и отзывчивые приложения. Сервер, в свою очередь, обрабатывает данные, управляет бизнес-логикой и обеспечивает безопасность.

Во-вторых, использование клиент-серверной архитектуры значительно повышает масштабируемость веб-приложения. Серверная часть может быть легко обновлена или заменена, не затрагивая пользовательский интерфейс, а это значит, что разработчики могут внедрять новые функции и технологии без необходимости переписывать всё приложение.

Кроме того, данная архитектура позволяет эффективно использовать API, что способствует интеграции с внешними сервисами и системами. Это открывает новые возможности для расширения функционала сайта, улучшения пользовательского опыта и упрощения доступа к необходимым данным.

Еще одним важным аспектом является возможность обработки больших объемов данных. Сервер может выполнять сложные вычисления и хранить информацию, что минимизирует нагрузку на клиентские устройства и обеспечивает более быструю и стабильную работу приложения.

Наконец, использование клиент-серверной модели способствует более высокому уровню безопасности. Сервер может реализовывать различные механизмы аутентификации и авторизации, защищая пользовательскую информацию и предотвращая несанкционированный доступ.

В результате, сочетание гибкости, возможности масштабирования, внедрения новых технологий и повышения безопасности делает архитектуру с разделением клиентской и серверной частей наилучшим выбором для создания современных, надежных и эффективных веб-приложений.

# **Обоснование выбора средств реализации дипломной работы**

Обоснование выбора средств реализации дипломной работы связано с необходимостью создания эффективного и доступного сайта для поликлиники, который будет отвечать потребностям пользователей и обеспечивать высокую производительность. В процессе выбора технологий была проведена тщательная оценка различных вариантов.

В качестве фронтенд-технологий были выбраны HTML и CSS. Эти языки обеспечивают создание структурированного и стилизованного контента, что позволяет формировать интуитивно понятный интерфейс для пользователей. HTML позволяет создавать семантически правильную разметку, что способствует улучшению доступности сайта, а CSS предоставляет гибкие возможности для оформления и адаптивного дизайна. Простота и надежность этих технологий также делают их идеальными для реализации веб-приложения, которое должно быть доступно на различных устройствах.

Серверная часть сайта была реализована на PHP. Этот язык программирования широко используется для разработки динамических веб-приложений и имеет большую поддержку сообщества. PHP позволяет эффективно обрабатывать запросы от пользователей и взаимодействовать с базами данных, что необходимо для управления информацией о пациентах и записях на прием. Его простота и широкие возможности интеграции с различными системами делают его идеальным выбором для серверной части проекта.

Для работы с базой данных была выбрана система MAMP, которая включает в себя Apache, MySQL и PHP. MAMP обеспечивает простую настройку локального веб-сервера, что делает его удобным инструментом для разработки и тестирования. Использование MySQL позволяет эффективно хранить и обрабатывать данные, обеспечивая при этом высокую производительность при работе с большими объемами информации. Эта система управления базами данных считается надежной и хорошо документированной, что упрощает процесс разработки и администрирования.

Таким образом, выбор средств реализации дипломной работы обоснован необходимостью создания доступного, безопасного и эффективного сайта для поликлиники. Использование HTML и CSS для фронтенда, PHP для серверной части и MAMP для работы с базой данных позволит обеспечить удобный пользовательский интерфейс и надежную обработку данных, что является ключевым для успешной работы медицинского учреждения. Эти технологии обеспечивают стабильность, простоту и масштабируемость проекта, что позволит в будущем легко адаптировать сайт под изменяющиеся потребности пользователей.

1. **Описание логики функционирования программного продукта**

Основные требования к функционированию сайта включают возможность записи на прием к врачу, просмотр расписания, получение информации о медперсонале и услугах, а также возможность обратной связи с поликлиникой. Внешний вид сайта должен быть интуитивно понятным и адаптивным, что обеспечит комфортное использование на различных устройствах – от настольных компьютеров до мобильных телефонов. На главной странице будет представлена информация о поликлинике, а также быстрые ссылки на основные сервисы: запись на прием, расписание врачей и контактные данные [6].

Входная информация для системы включает данные пользователей, такие как имя, контактная информация, информация о медицинских услугах и записи на прием, а также данные о врачах и расписании. Выходная информация будет представлять собой подтверждения записей, информацию о доступности услуг, расписаниях, рекомендации врача и сообщения, отправленные пользователям.

Схема функциональной структуры программного продукта включает следующие основные компоненты:

1. Пользовательский интерфейс – главная страница, страницы регистрации и авторизации, страницы управления записями на прием [1].
2. Серверная логика – обработка запросов от пользователей, взаимодействие с базой данных [2].
3. База данных – хранение информации о пациентах, врачах, услугах и записях на прием [10].

Логическое моделирование базы данных предполагает создание нескольких ключевых таблиц:

* Таблица пользователей: содержит поля для ID пользователя, имени, контактных данных и роли (пациент, администратор).
* Таблица врачей: включает ID врача, имя, специальность, доступные часы работы.
* Таблица услуг: информация о медицинских услугах, их стоимости и времени, необходимом для их предоставления.
* Таблица записей на прием: связи между пользователями и врачами, время записи, статус (подтверждено/отменено).

Физическое моделирование базы данных включает реализацию вышеуказанных таблиц в выбранной системе управления базами данных (MySQL). Каждая таблица будет иметь соответствующие индексы для оптимизации поиска данных, что обеспечит быструю обработку запросов от пользователей. Важное внимание будет уделено соблюдению норм и правил хранения персональных данных в соответствии с законодательством.

Таким образом, логика функционирования программного продукта тщательно продумана с акцентом на удобство и безопасность пользователей, что позволяет создать надежный и эффективный сайт для поликлиники.

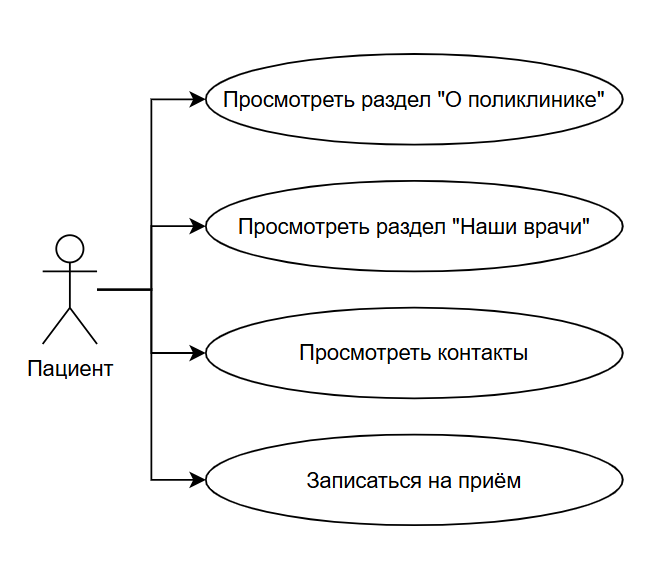
UML диаграмма деятельности для сайта — это визуальная схема, которая описывает последовательность действий, происходящих при взаимодействии пользователя с сайтом, и отображает процессы внутри системы (см. ПРИЛОЖЕНИЕ А). Она нужна для того, чтобы наглядно представить, какие шаги выполняет пользователь (например, регистрация, поиск информации, оформление заказа), какие решения принимает система на каждом этапе, а также как переходят данные между различными частями сайта. Такая диаграмма помогает понять и структурировать логику работы сайта, выявить возможные проблемы и согласовать требования между разработчиками, дизайнерами и заказчиком. В итоге ее использование упрощает процесс проектирования, способствует более эффективному взаимодействию участников проекта и снижает риск ошибок при разработке сайта.

UML диаграмма деятельсности для одностраничного сайта поликлиники описывает последовательность взаимодействия пользователя с основными разделами сайта и переходы между ними. Пользователь начинает работу с сайта с открытия главной страницы, где ему доступны ключевые разделы: информация о поликлинике, список врачей, запись на прием, отзывы и контакты[3]. При выборе любого из разделов пользователь переходит к соответствующему состоянию: знакомится с информацией о поликлинике, просматривает карточки врачей, либо заполняет форму для онлайн-записи к врачу, последовательно вводя имя, телефон, выбирая специалиста и удобное время. После отправки формы пользователь получает уведомление об успешной записи или ошибке. В разделе отзывов представлены мнения и оценки других посетителей, а в контактах — необходимая информация для связи с учреждением. С любого состояния пользователь может вернуться на главную страницу или открыть другой раздел либо завершить работу с сайтом. Диаграмма помогает наглядно показать, как организована навигация и взаимодействие на сайте, что облегчает планирование, разработку и обсуждение интерфейса между всеми участниками проекта.

UML-диаграмма состояний для сайта — это визуальное средство, позволяющее описать, как пользователь или система последовательно перемещается между различными состояниями или разделами сайта в ответ на действия или события (см. ПРИЛОЖЕНИЕ В) [4]. Такая диаграмма показывает, с чего начинается взаимодействие, какие разделы или функции доступны в разные моменты, что происходит при выборе определенного действия, а также как можно вернуться к предыдущему состоянию или завершить работу. Использование UML-диаграммы состояний помогает разработчикам, дизайнерам и заказчикам видеть логику переходов и сценариев на сайте целиком: это упрощает проектирование интерфейса, выявление ошибок или «узких мест», согласование задач между членами команды и обеспечивает понимание того, каким должен быть «путь» пользователя. Диаграмма особенно полезна для сложных проектов, где важно не упустить ни одного шага пользовательского взаимодействия и сделать сайт интуитивно понятным.

Данная диаграмма показывает, в каких состояниях может находиться пользователь сайта поликлиники и как он может переходить между этими состояниями. Пользователь начинает работу с сайта на главной странице, откуда он может попасть в любой другой раздел: О поликлинике, где содержится основная информация об учреждении; Наши врачи, с фотографиями и данными о специалистах; Форма записи на приём, где нужно заполнить необходимые поля для оформления визита; Отзывы, где представлены мнения пациентов; а также Контакты с актуальными контактными данными. С любого из этих разделов пользователь может перейти в любой другой через меню. Особенность состояния "Записаться" заключается в том, что пользователь может оставаться в нем до завершения или отмены заполнения формы, после чего происходит возврат на главную или переход в любой другой раздел. Структура сайта построена так, чтобы все разделы были взаимосвязаны, обеспечивая быстрый и удобный доступ к любой информации или функции из любого места, что характерно для одностраничных приложений. В итоге, диаграмма наглядно демонстрирует, как пользователь может полноценно взаимодействовать с сайтом поликлиники: от первого входа до онлайн-записи или изучения подробной информации о врачах и самом учреждении.

UML-диаграмма прецедентов — это визуальное представление основных сценариев взаимодействия пользователя с системой. На такой диаграмме показываются внешние участники (например, пациенты и администраторы) и основные действия, которые они могут выполнять на сайте поликлиники, например, просмотр информации о врачах, запись на приём, чтение отзывов или отправка сообщения через форму обратной связи. Такая диаграмма помогает лучше понять, какие функции необходимы на сайте и какие задачи он должен решать для разных пользователей. Кроме того, это упрощает взаимодействие между заказчиком, аналитиками и разработчиками, ведь диаграмма помогает согласовать требования и ожидания от продукта. Она также служит основой для дальнейшего проектирования интерфейса и архитектуры системы. В итоге, UML-диаграмма прецедентов позволяет заранее определить, что должен уметь сайт поликлиники и кто будет им пользоваться, что очень важно на начальных этапах работы, чтобы избежать недопонимания и определить основные сценарии использования сайта.



1. Диаграмма прецедентов сайта

Пациент, являясь основным пользователем одностраничного сайта поликлиники, может взаимодействовать с несколькими функциями сайта. Он имеет возможность просматривать раздел «О поликлинике», где представлена основная информация о медицинском учреждении, а также знакомиться с разделом «Наши врачи», чтобы узнать о специалистах, работающих в поликлинике. Также пациент может читать отзывы других пациентов и просматривать контактные данные поликлиники. Помимо информационных возможностей, предусмотрен сценарий записи на приём: для этого пациент должен заполнить специальную электронную форму, поочерёдно вводя своё имя, номер телефона, выбирая нужного врача, подходящие дату и время посещения, а затем отправить заявку на приём. Все перечисленные действия инициируются самим пациентом и выполняются последовательно через интерфейс сайта.

1. **Выводы по второй главе**

Во второй главе рассматриваются средства реализации дипломной работы, что является ключевым элементом для успешного завершения проекта. Начнем с обзора инструментальных средств создания программного продукта. В этом разделе проанализированы различные технологии и инструменты, использующиеся для разработки, такие как языки программирования, среды разработки, базы данных и фреймворки. Оценены их функциональные возможности, совместимость и удобство использования, а также то, как они соответствуют требованиям, выявленным на этапе анализа производственного предприятия.

Следующим пунктом является обоснование выбора средств реализации дипломной работы. Здесь представлено детальное объяснение, почему были выбраны именно те инструменты и технологии, которые описаны в предыдущем разделе. Рассматриваются ключевые факторы, повлиявшие на решение: производительность, легкость в освоении, поддержка сообщества, а также наличие документации. Это обоснование поможет читателю понять, как выбранные средства будут способствовать построению эффективного и надежного программного продукта.

Завершает главу описание логики функционирования программного продукта. Этот раздел содержит информацию о том, как будет организовано взаимодействие всех компонентов системы, включая пользовательский интерфейс, бизнес-логику и взаимодействие с базами данных. Описываются основные алгоритмы работы, сценарии использования, а также архитектура системы, что позволит понять, каким образом реализуются все функции и обеспечивается работоспособность продукта в целом.

В итоге, вторая глава подводит итоги выбора инструментов и технологий, обосновывая их полезность и эффективность, а также дает полное представление о логике функционирования программного продукта, что станет важным этапом для его реализации и успешного завершения дипломной работы.

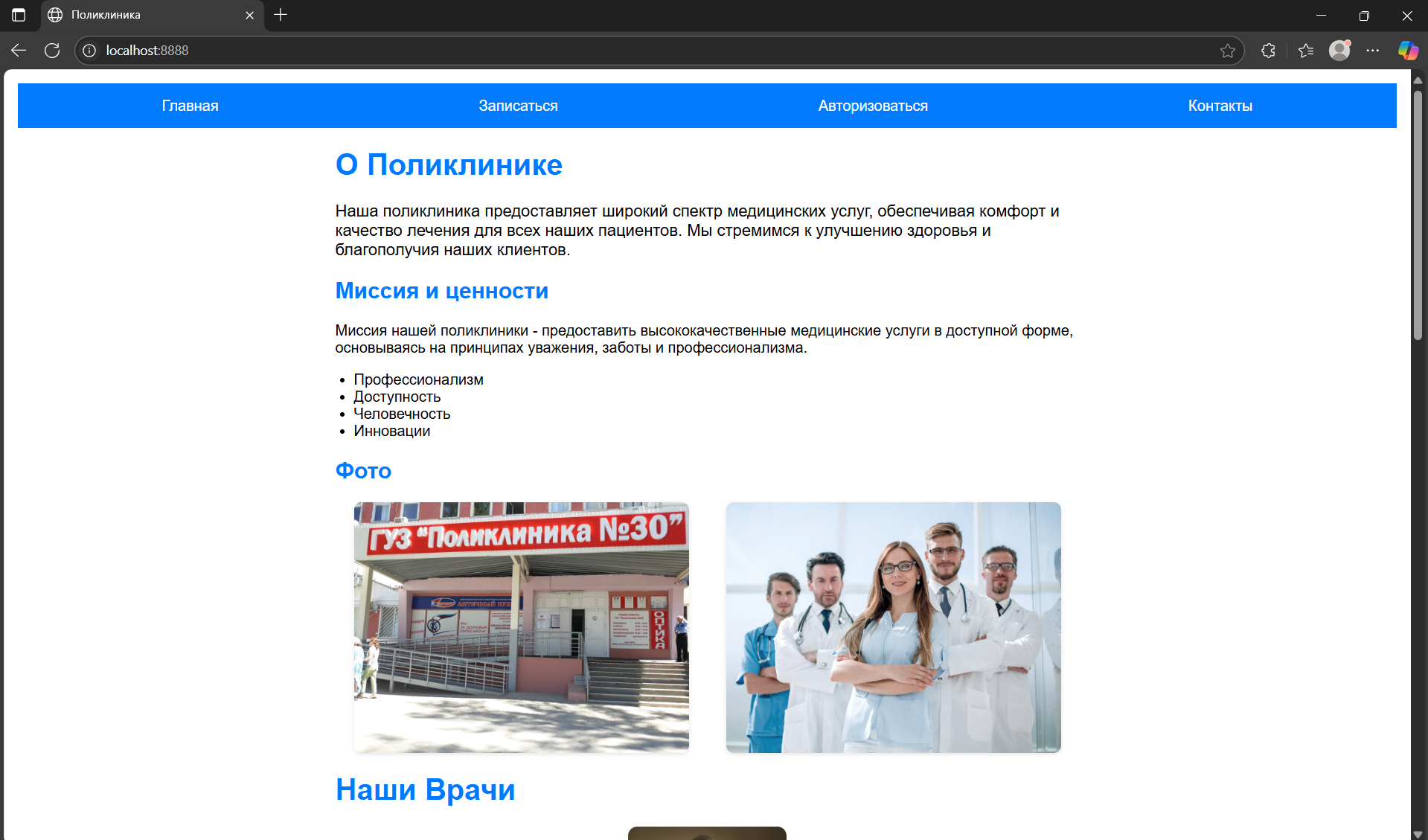
# **3 СТРУКТУРА И РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**

# **Разработка интерфейса программного продукта**

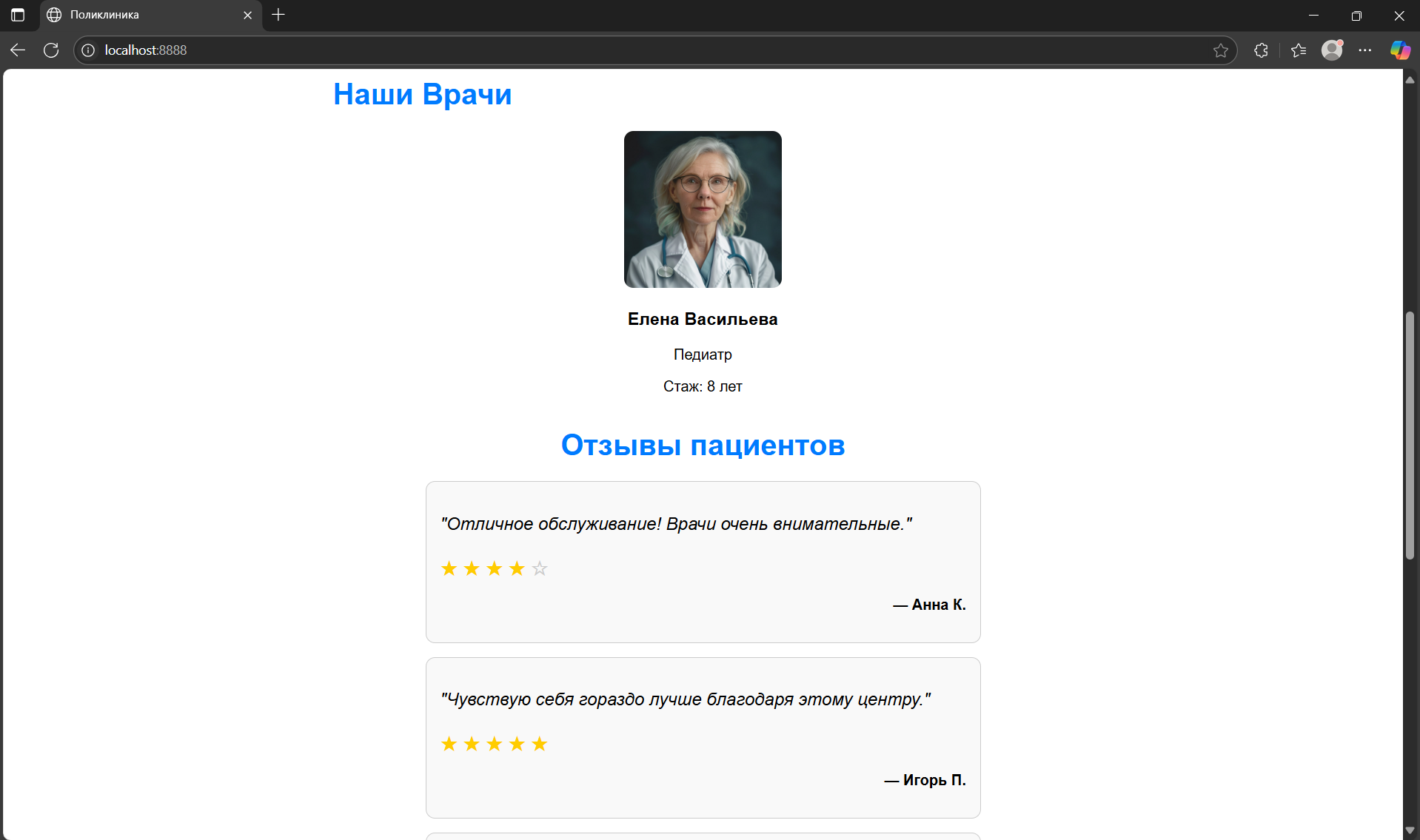
Пользовательский интерфейс – это все, что помогает человеку управлять программой, системой или устройством. Это своеобразный мост между пользователем и "мозгами" машины. Примеры интерфейсов в реальной жизни - кнопки в лифте, меню банкомата, экран электронной книги. В цифровом мире интерфейсами оснащены компьютеры, смартфоны и планшеты. Графический интерфейс Windows, меню и иконки приложений на телефоне, интернет-магазин на сайте - все это примеры UI. Основная задача интерфейса – упростить работу пользователя, дать возможность решать нужные задачи быстро и комфортно. Хороший UI позволяет управлять гаджетом или программой интуитивно, без лишних раздумий. Он незаметен и дает возможность фокусироваться на цели, а не на инструментах. Плохой интерфейс заставляет пользователя ломать голову, злиться и тратить время впустую.

Разработка программного интерфейса является ключевым этапом в создании современного программного обеспечения. Интерфейсы служат связующим звеном между пользователями и приложением, поэтому их грамотная реализация играет важную роль в удобстве и эффективности использования продукта. Для создания полноценного интерфейса необходимо учитывать основные разделы, такие как главная, записаться, авторизоваться и контакты.

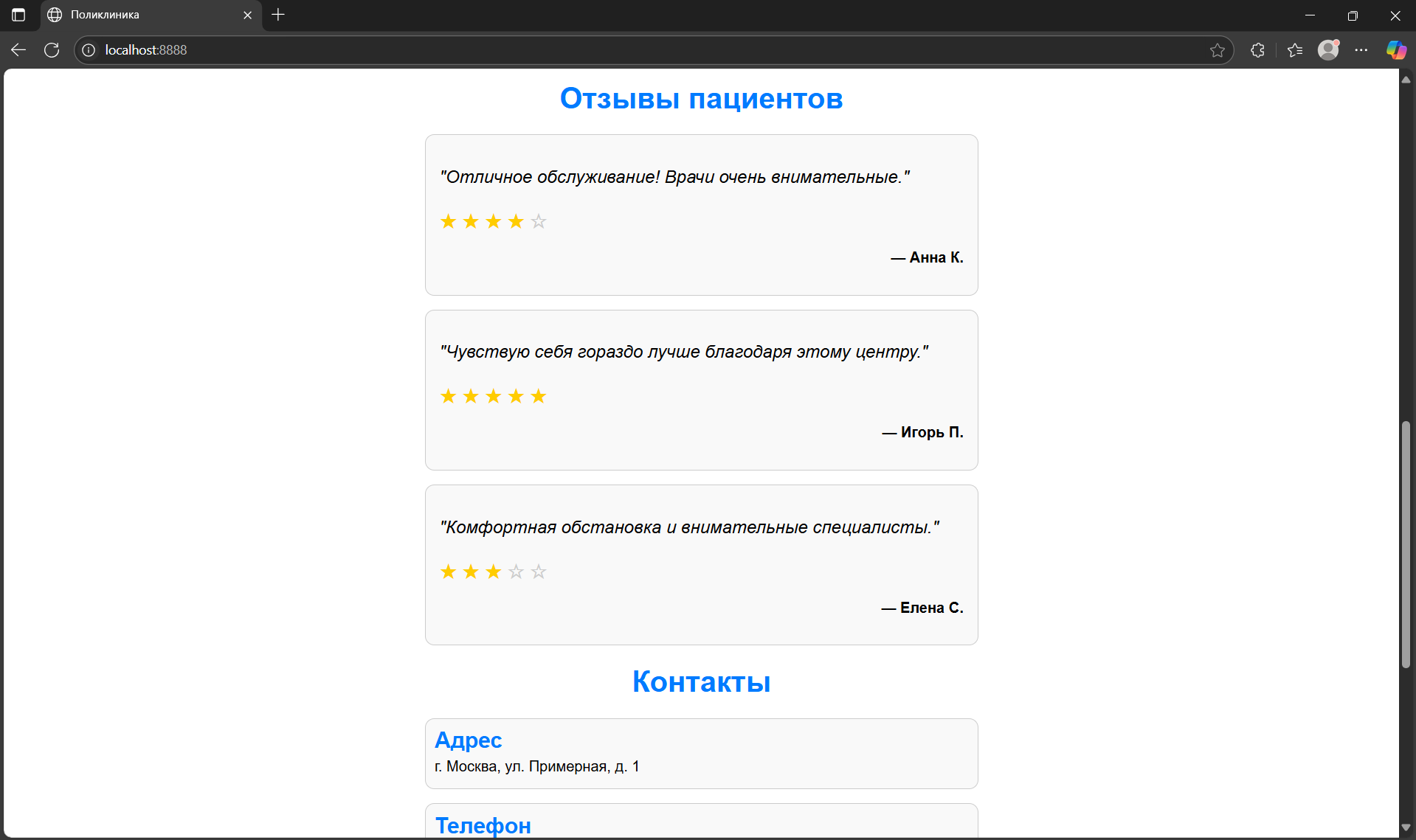
Начнём с главной страницы (рисунки 5-8), которая является визитной карточкой приложения. Она должна привлекать внимание пользователя и кратко знакомить его с функционалом. На этой странице размещается логотип приложения, который способствует узнаваемости, а также меню навигации, позволяющее без труда перейти ко всем основным разделам. Важно добавить краткое описание приложения, где будут изложены его преимущества и ключевые функции. Визуальные элементы, такие как слайды и баннеры, помогут ярко продемонстрировать функционал. Соблюдение единой цветовой палитры и шрифтов важно для визуальной гармонии и интуитивной понятности интерфейса.



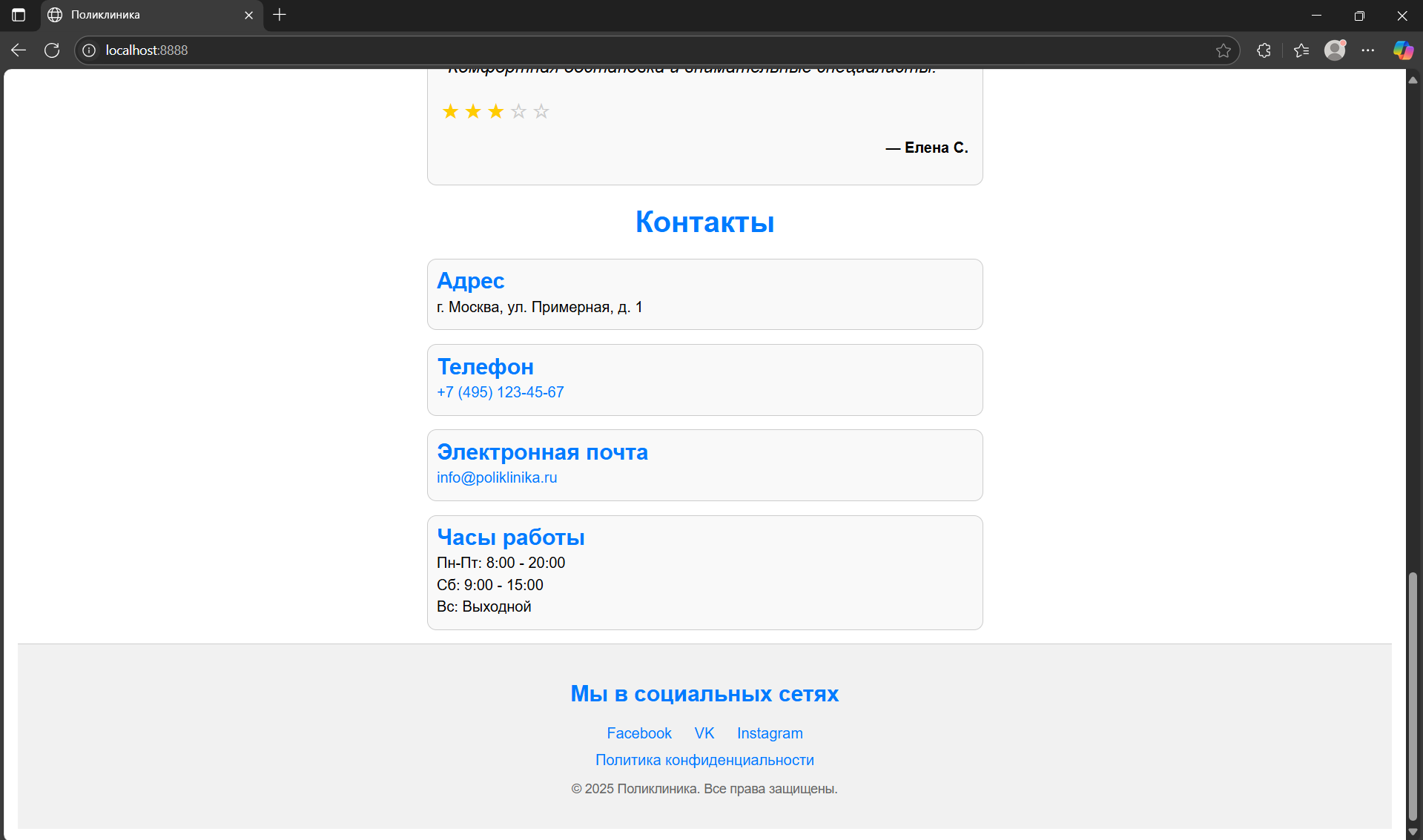
1. Главная страница разделы «О поликлинике», миссия и ценности, фото



1. Главная страница: наши врачи и отзывы пациентов

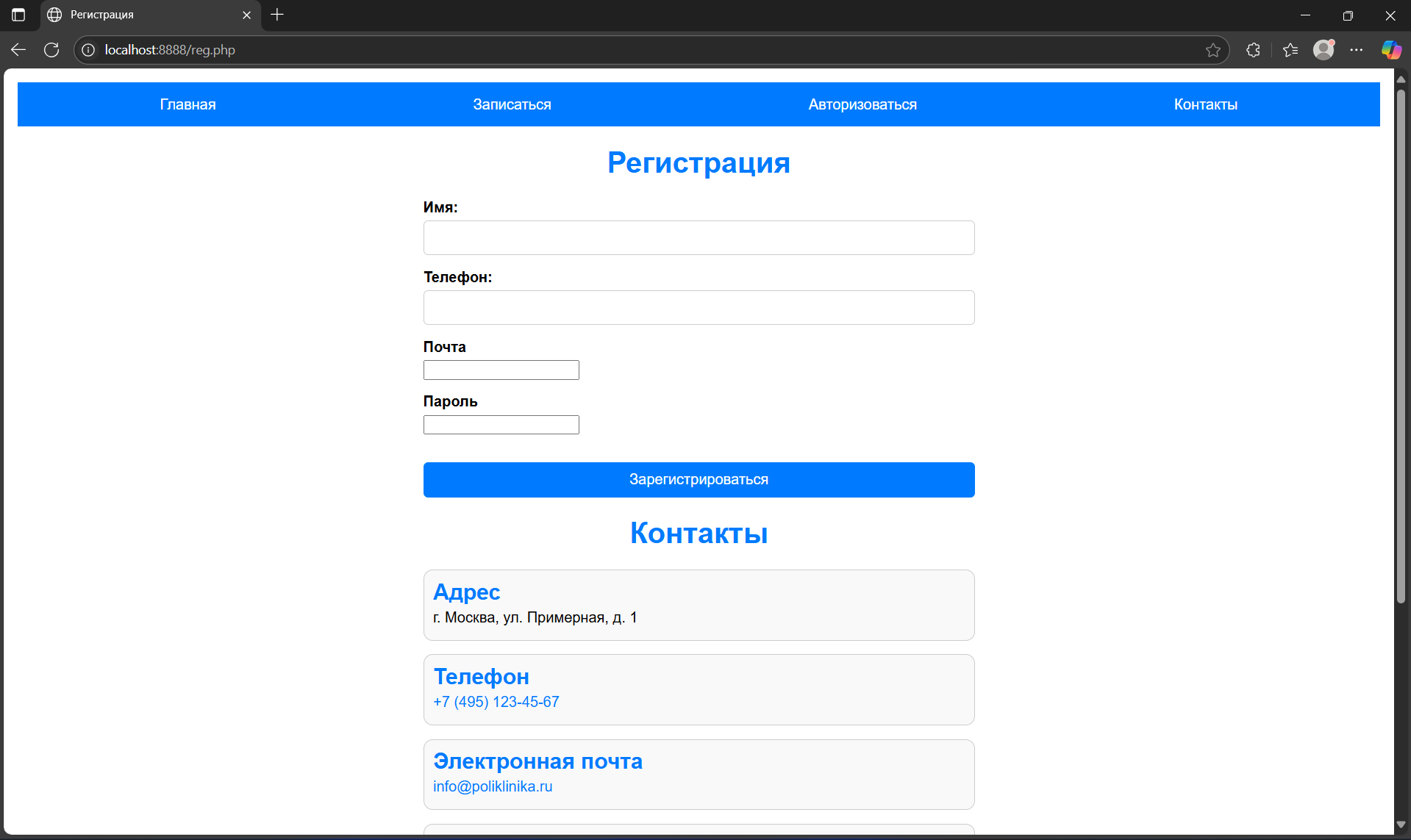


1. Главная страница: отзывы пациентов



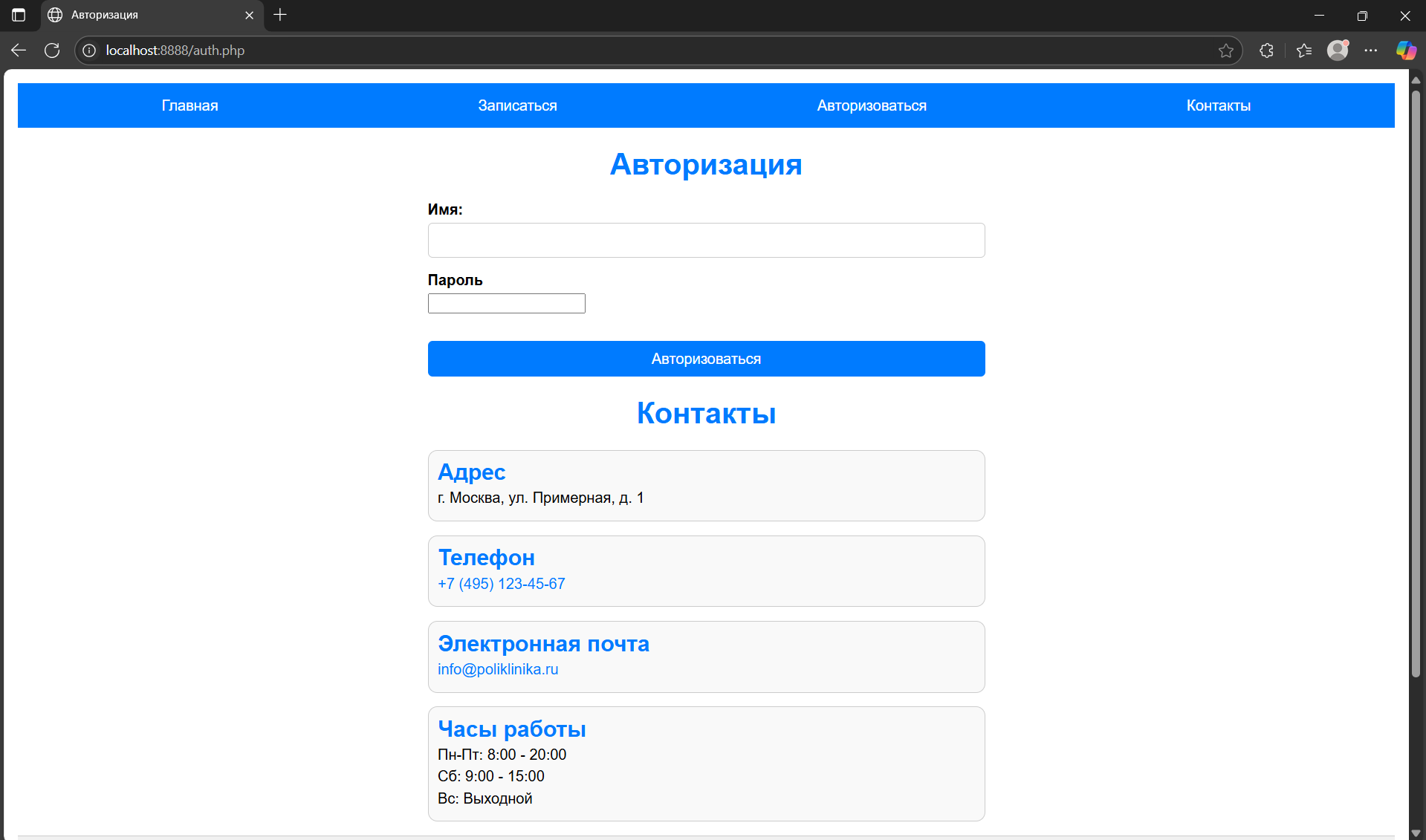
1. Главная страница: контакты и подвал сайта

Переходя к разделу «Записаться» (рисунок 9), необходимо предусмотреть форму для регистрации, где пользователю будет предложено ввести имя, электронную почту, телефон и пароль. Валидация данных критически важна для предотвращения ошибок, а выделенная кнопка для отправки данных упростит взаимодействие. Сообщения об ошибках должны быть ясными, чтобы пользователь мог быстро устранить все недочеты. Для повышения удобства стоит добавить возможность быстрой регистрации через социальные сети.



1. Форма регистрации в соответствующем разделе

Далее следует раздел «Авторизоваться» (рисунок 10), который предназначен для пользователей, уже зарегистрированных в приложении. Он должен быть простым и интуитивно понятным. Здесь размещается форма для ввода логина и пароля с четкой сигнализацией обязательных полей. Кнопка «Войти» должна быть заметной и легко доступной, а также следует обеспечить ссылку для восстановления пароля, чтобы пользователи могли без труда восстановить доступ к своим аккаунтам.



1. Форма авторизации в соответствующем разделе

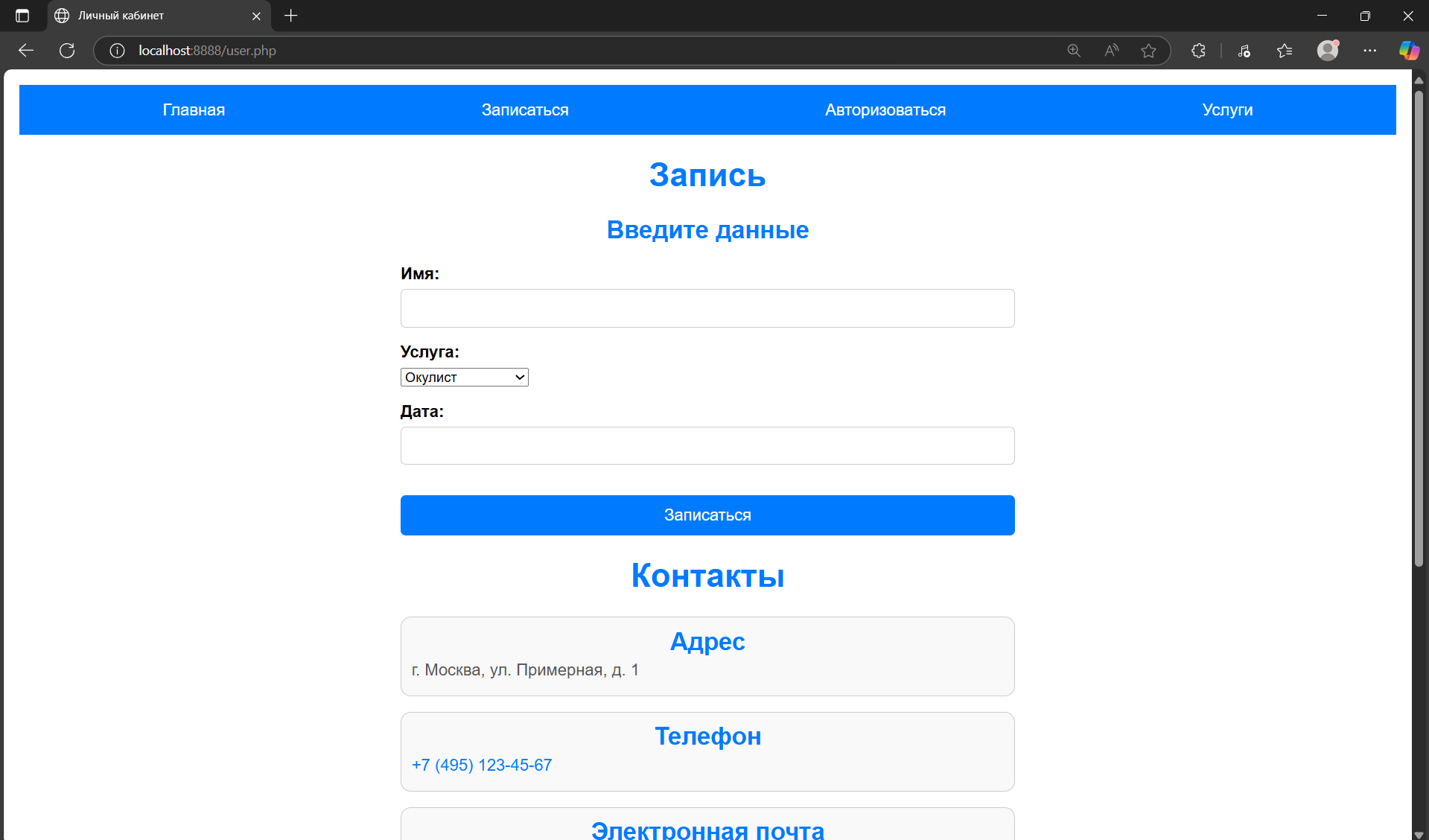
После успешной авторизации на сайте поликлиники пользователю открывается интуитивно понятная форма записи ко врачу (рисунок 11). На этом этапе пациента встречает приветливый интерфейс, в котором предусмотрены обязательные поля для заполнения.

Первое поле предназначено для ввода имени пациента. Здесь пользователь должен указать свое полное имя, что позволяет поликлинике идентифицировать его при записи на прием. Далее следует выпадающий список, где представлены различные специальности врачей. Это удобно и практично, так как позволяет быстро выбрать необходимого специалиста, например, терапевта, хирурга или педиатра.

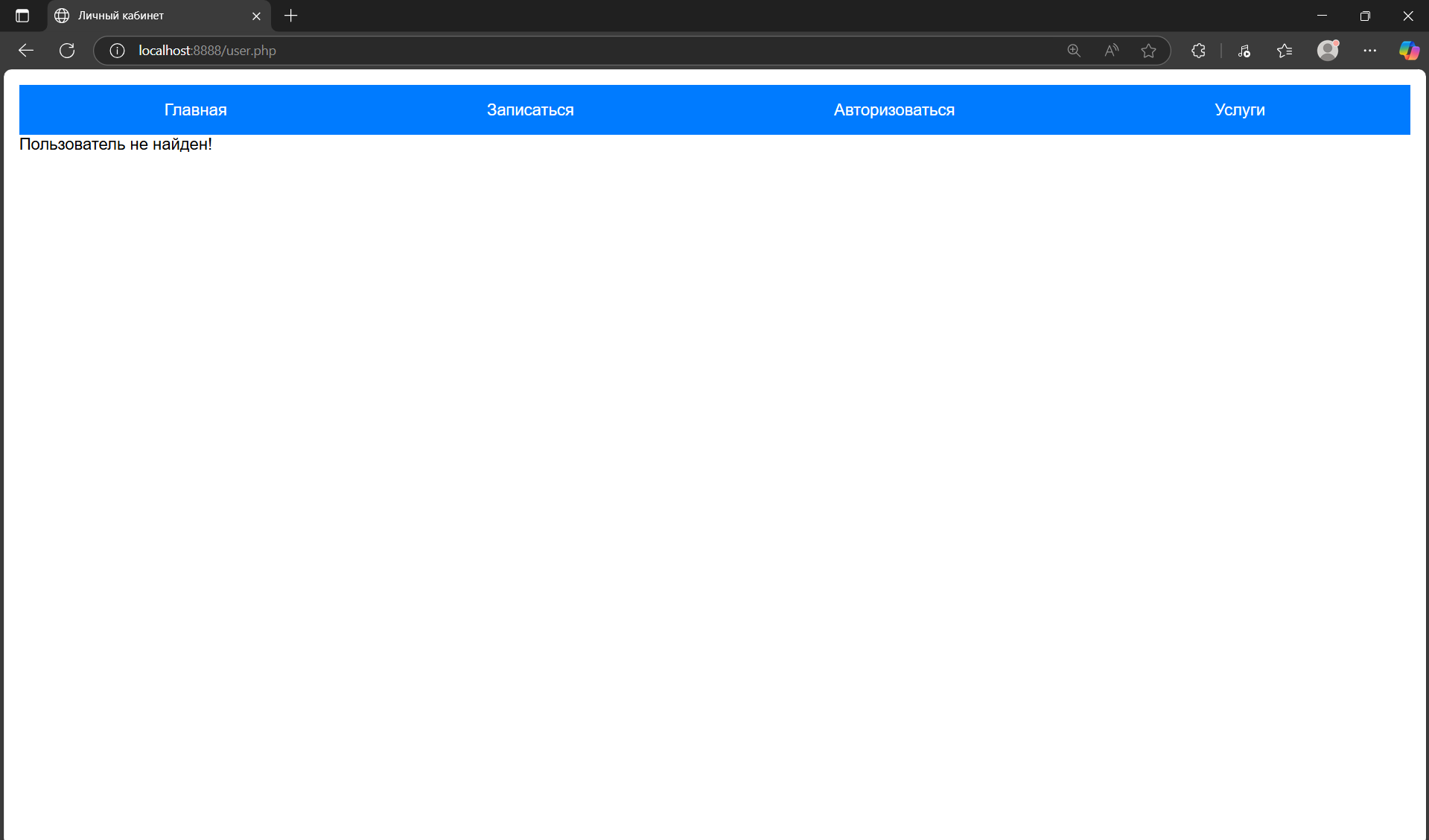
Следующим шагом служит выбор даты записи на прием. Поле для ввода даты оформлено так, чтобы исключить возможные ошибки: пользователь может просто выбрать нужную дату из календаря, что облегчает процесс записи и минимизирует вероятность опечаток.

Все поля формы являются обязательными для заполнения, что гарантирует правильность данных и их полное соответствие требованиям системы. Если пользователь попытается отправить форму, не заполнив одно из полей, он получит отчетливое оповещение, информирующее о необходимости до заполнения всех критериев. Если введённое имя пользователя не зарегистрировано в базе данных, то пользователь получает соответствующее оповещение (рисунок 12).

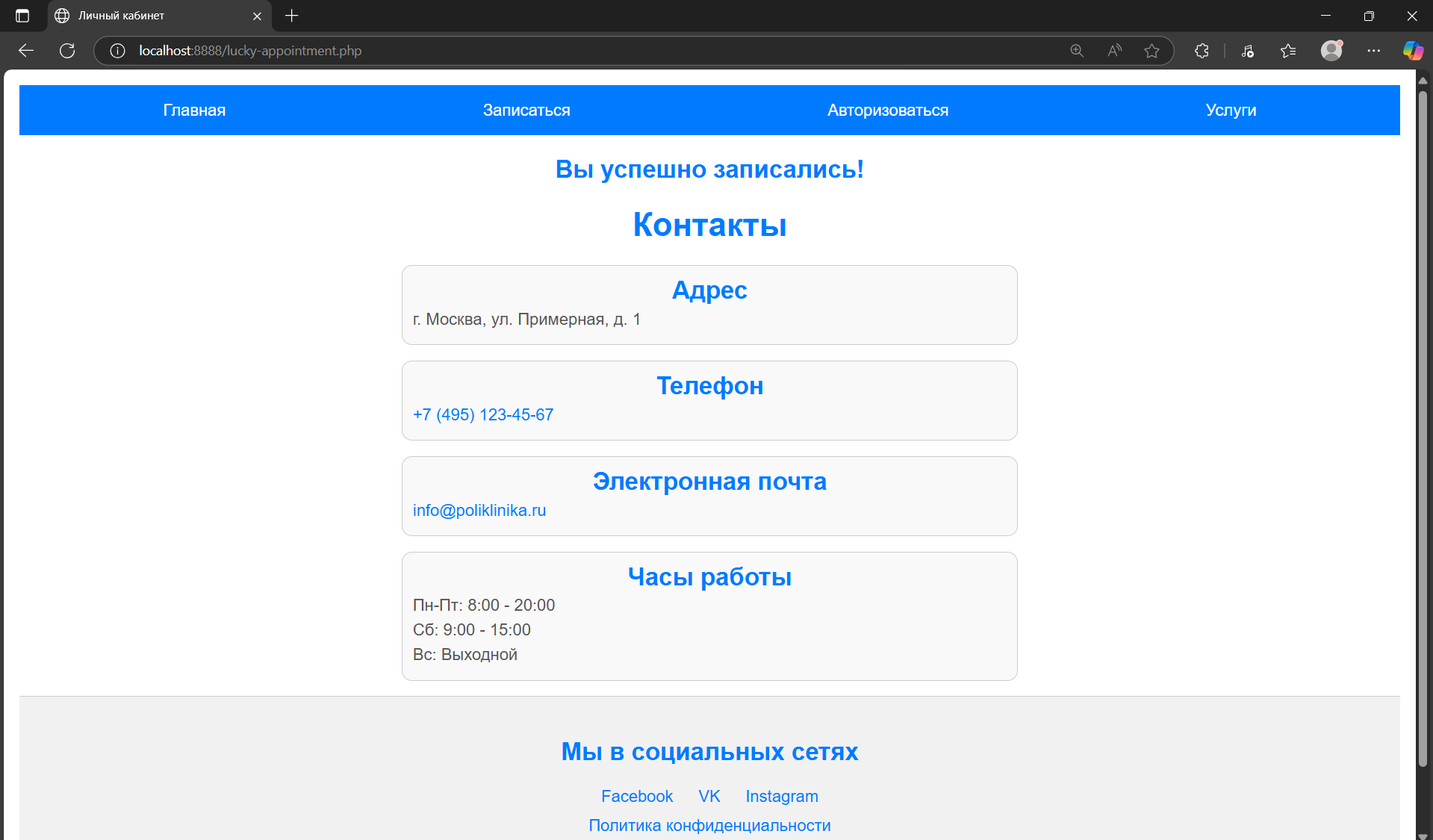
Однако, если введенное имя пользователя соответствует зарегистрированному в базе данных, и все данные корректны, система формирует надежное подтверждение успешной записи на прием (рисунок 13). В этот момент пациент получает уверенность в том, что его запрос обработан, и он получил желаемую дату и время консультации с врачом.



1. Форма записи ко врачу после авторизации



1. Сообщение, выводящееся в случае, если введённый пользователь не существует



1. Сообщение об успешной регистрации

Таким образом, этот процесс записи ко врачу отличается удобством, прозрачностью и высокой степенью надежности, что делает его важным инструментом в управлении медицинскими записями и уходом за пациентами.

Завершая, стоит отметить, что создание программного интерфейса требует тщательной проработки дизайна и функционала. Каждый из разделов должен поддерживать единый стиль, что позволит избегать путаницы у пользователей. Удобство навигации, четкость визуальных элементов и доступность информации играют решающую роль в формировании успешного приложения. Единство дизайна, простота взаимодействия и ясность представления информации сделают интерфейс привлекательным для пользователей, что, в свою очередь, повысит их заинтересованность и лояльность, способствуя успешному использованию продукта.

# **Рекомендуемые системные требования**

Системные требования — это спецификации и характеристики, которые должны быть у компьютерной системы (например, персонального [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) или [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)), чтобы оно могло корректно функционировать и выполнять свои задачи. Эти требования определяют минимальные и рекомендуемые характеристики аппаратного и программного обеспечения, которые необходимы для оптимальной работы системы.

Для программного обеспечения системные требования могут включать в себя:

1. Операционную систему: Какая версия операционной системы требуется для запуска программы.
2. Процессор: Минимальный тип и тактовая частота процессора.
3. Оперативную память (RAM): Минимальное количество оперативной памяти, необходимой для работы программы.
4. Дисковое пространство: Свободное место на жестком диске, необходимое для установки и функционирования программы.
5. Графическую карту: в случае программ, требующих трехмерную графику, могут быть указаны требования к видеокарте.
6. Звуковое оборудование: Если программа использует звуковой вывод, могут быть указаны соответствующие требования.

Для аппаратного обеспечения, такого как компьютеры и игровые консоли, системные требования могут включать в себя:

1. Процессор: Тип, количество ядер и тактовая частота.
2. Оперативную память (RAM): Общий объем оперативной памяти.
3. Графическую карту: Тип и объем видеопамяти.
4. Жесткий диск или SSD: Объем свободного места для хранения данных.
5. Порты и подключения: Например, USB-порты, HDMI, Ethernet и др.

Для эффективной работы сайта поликлиники необходимы определенные системные требования, которые обеспечат стабильную и быструю работу всех функциональных возможностей. Вот рекомендуемые системные требования:

Серверные требования:

1. Операционная система: Linux (например, Ubuntu Server) – Linux является более надежной и безопасной системой для хостинга веб-приложений, а также требует меньших затрат на ресурсы.
2. Веб-сервер: Apache или Nginx – оба веб-сервера хорошо справляются с высокой нагрузкой и имеют обширную поддержку, что делает их хорошим выбором для динамических сайтов.
3. Процессор: Не менее 2 GHz, 2 ядра – многоядерный процессор позволяет обработать большее количество запросов одновременно и улучшает быстродействие сайта.
4. Оперативная память (RAM): 8 ГБ – достаточное количество оперативной памяти помогает обрабатывать несколько запросов одновременно и обеспечивает быструю загрузку страниц.
5. Дисковое пространство: Минимум 100 ГБ SSD – SSD-накопители обеспечивают более высокую скорость чтения и записи данных, что критично для быстрого доступа к информации на сайте.
6. База данных: MySQL или PostgreSQL – реляционные базы данных являются стандартными для веб-приложений и хорошо подходят для хранения структурированной информации, такой как записи пациентов.
7. SSL-сертификат: Обязательно – использование SSL (HTTPS) критически важно для обеспечения безопасности передачи данных между пользователями и сервером, особенно когда речь идет о медицинской информации.

Клиентские требования:

1. Браузеры: Последние версии Chrome, Firefox, Safari, Edge – современные браузеры поддерживают новейшие технологии и стандарты, что обеспечивает правильное отображение и функционирование сайта.
2. Операционная система: Windows, macOS, Linux, мобильные ОС (iOS, Android) – сайт должен быть доступен на всех популярных операционных системах, чтобы обеспечить доступность для максимального количества пользователей.
3. Интернет-соединение: Минимум 1 Мбит/с – хотя сайт будет работать медленнее при низкой скорости, такая скорость будет достаточной для загрузки большинства страниц и взаимодействия с функционалом.

Обоснование системных требований:

1. Надежность и безопасность: использование серверов на базе Linux и формирование надежной архитектуры с SSL-сертификатом позволяет защитить персональные данные пациентов, такие как медицинские истории и контактные данные.
2. Быстродействие: SSD-накопители и достаточный объем оперативной памяти улучшают производительность, что важно для обеспечения быстрого доступа к информации. Пациенты не любят ждать, и быстрая загрузка страниц привлекает больше пользователей.
3. Кроссплатформенность: адаптация сайта для различных операционных систем и браузеров гарантирует, что более широкий круг пациентов сможет получить доступ к услугам поликлиники. Это особенно важно в условиях увеличения числа мобильных пользователей.
4. Масштабируемость: выбор серверного аппаратной составляющей с учетом приростов нагрузки (например, увеличение количества пользователей, добавление новых услуг) позволяет избежать частой замены оборудования и переключения на более мощные серверы.

Эти системные требования помогут обеспечить не только стабильную работу сайта поликлиники, но и высокий уровень безопасности и удобства для пациентов.

# **Обеспечение безопасности информации**

Обеспечение безопасности информации на сайте поликлиники требует комплексного подхода. Вот основные меры, которые можно принять:

1. Шифрование данных. Использование протокола HTTPS для защиты данных, передаваемых между пользователями и сервером. Это предотвратит перехват информации третьими сторонами.
2. Защита паролей. Пароли пользователей должны храниться в зашифрованном виде с использованием надежных алгоритмов хеширования. Рекомендуется реализация политики сложных паролей.
3. Регулярные обновления ПО. Обновляйте все компоненты сайта (CMS, плагины, библиотеки) до последних версий, чтобы устранить известные уязвимости.
4. Защита от DDoS-атак. Использование инструментов и сервисов, которые могут защитить ваш сайт от распределенных атак, направленных на перегрузку системы.
5. 6. Регулярное резервное копирование. Проведение регулярного резервного копирования данных, чтобы в случае инцидента можно было восстановить информацию.
6. 7. Мониторинг и аудит безопасности. Реализованные системы мониторинга, которые отслеживают подозрительную активность и проводите регулярные аудиты безопасности.
7. 8. Обучение персонала. Проведение обучение для сотрудников по вопросам кибербезопасности, чтобы они знали о возможных угрозах и способы их предотвращения.
8. 9. Защита от SQL-инъекций и XSS. Обеспечение защиты от распространенных атак, таких как SQL-инъекции и межсайтовые скрипты, используя валидированные вводы и фильтрацию данных.

В рамках разработки сайта было реализовано ряд ключевых мер, направленных на обеспечение безопасности пользовательских данных и защиту информации от несанкционированного доступа. Прежде всего, акцент был сделан на защиту паролей пользователей. Для этого применялась технология хеширования, что позволяет надежно хранить пароли в зашифрованном виде. Вместо хранения исходного пароля в базе данных, используется специальный алгоритм, который генерирует уникальный хеш, не подлежащий обратному преобразованию. Это гарантирует, что даже в случае утечки данных злоумышленники не смогут восстановить оригинальные пароли.

Кроме того, реализована политика создания сложных паролей, что существенно затрудняет их подбор. Пользователи обязаны выбирать пароли, содержащие комбинацию букв верхнего и нижнего регистра, цифр и специальных символов. Также предусмотрена возможность включения многофакторной аутентификации, что добавляет дополнительный уровень безопасности при входе в систему.

Важным аспектом защиты данных стала реализация протокола HTTPS, обеспечивающего шифрование информации, передаваемой между пользователем и сайтом. Это предотвращает перехват данных третьими сторонами во время их передачи.

Регулярные обновления программного обеспечения (как серверного, так и клиентского) также находятся в центре внимания нашей стратегии безопасности, поскольку это позволяет закрывать уязвимости, которые могут быть использованы злоумышленниками.

Дополнительно был разработан механизм защиты от распространенных атак, таких как SQL-инъекции и межсайтовые скрипты (XSS), с использованием валидированных вводов и фильтрации данных, что позволяет минимизировать риски несанкционированного доступа и манипуляций с базой данных.

Все эти меры направлены на создание безопасной среды для пользователей, что способствует доверию и уверенности в надежности нашего сайта.

# **Определение экономической эффективности внедрения программного продукта**

Внедрение созданного программного продукта становится выгодным и рациональным решением для организации, так как оно приносит ряд финансовых и организационных преимуществ. Во-первых, данный продукт позволяет значительно сократить операционные затраты. Например, автоматизация рутинных задач уменьшает потребность в ручном труде, что, в свою очередь, снижает расходы на заработную плату.

Во-вторых, программный продукт способствует повышению производительности труда сотрудников. За счет оптимизации процессов и улучшения взаимодействия между отделами, работники смогут уделять больше времени ключевым задачам, что сказывается на общей эффективности бизнеса.

Кроме того, внедрение программного продукта снижает вероятность ошибок, связанных с человеческим фактором. Повышение точности выполняемых операций и уменьшение количества повторных работ также благоприятно сказывается на финансовом состоянии организации.

Организационные преимущества заключаются в улучшении качества обслуживания клиентов и увеличении скорости обработки заявок. Это приводит к повышению уровня удовлетворенности клиентов и, как следствие, к росту клиентской базы и увеличению прибыли.

Определение экономической эффективности внедрения программного продукта заключается в оценке затрат на разработку, внедрение и сопровождение сайта, а также в выявлении экономических выгод. Затраты на разработку включают оплату труда разработчиков (фронтенд и бэкенд) в размере от 100,000 до 300,000 рублей, дизайн и UX от 30,000 до 70,000 рублей и тестирование с отладкой от 20,000 до 50,000 рублей, что в итоге составит от 150,000 до 420,000 рублей. Затраты на внедрение включают обучение сотрудников (15,000 - 30,000 рублей), настройку серверов и хостинга (10,000 - 25,000 рублей) и первоначальную маркетинговую кампанию (20,000 - 50,000 рублей), что в сумме составит от 55,000 до 105,000 рублей. Затраты на сопровождение годовые, включая хостинг и доменное имя (5,000 - 15,000 рублей) и техническую поддержку с обновлениями (20,000 - 50,000 рублей), составляют от 25,000 до 65,000 рублей в год. Определяя экономические выгоды, можно выделить сокращение расходов за счет автоматизации процессов, что позволит сэкономить до 30% на операционных расходах, а также уменьшить потребность в физическом пространстве для хранения и обработки данных. Повышение производительности ожидается на уровне 25-40% благодаря более эффективным процессам, что ускорит обработку запросов и взаимодействие с клиентами, ведущее к увеличению объема продаж. Уменьшение ошибок достигается снижением их количества на 50% за счет автоматизации и валидации данных, что также повышает качество предоставляемых услуг. В целом, внедрение сайта будет способствовать значительному сокращению затрат и повышению общей эффективности бизнеса, с ожидаемым возвратом инвестиций (ROI) в течение первого года от 150% до 300%.

Экономическая эффективность внедрения сайта поликлиники с функциями записи, авторизации и просмотра услуг может быть оценена через несколько ключевых аспектов:

* 1. Снижение операционных затрат. Внедрение сайта позволяет автоматизировать процесс записи на прием, уменьшая необходимость в ручной работе сотрудников регистратуры. Это может привести к снижению расходов на зарплату и уменьшению очередей, что в свою очередь повышает уровень удовлетворенности пациентов.
  2. Увеличение числа записей. Удобство онлайн-записи может значительно увеличить количество пациентов, так как пользователи могут записываться в любое время, не завися от работы регистратуры. Это также может способствовать повышению общей посещаемости поликлиники.
  3. Улучшение качества обслуживания. Функции авторизации и предоставления информации о доступных услугах позволяют пациентам быстрее находить необходимую информацию и уменьшать время ожидания. Это улучшает качество обслуживания и способствует повышению лояльности клиентов.
  4. Сбор и анализ данных. Система может собирать данные о записавшихся пациентах, их предпочтениях и часто запрашиваемых услугах. Это поможет в дальнейшей оптимизации предоставляемых услуг и более точно настраивать маркетинговые стратегии поликлиники.
  5. Возможности для дополнительных функций. Внедрение сайта открывает возможности для расширения функционала, такого как онлайн-консультации, возможность получения уведомлений о записи и напоминаний, что дополнительно повышает уровень взаимодействия с пациентами.
  6. Экономия времени пациентов. Пациенты избавляются от необходимости ожидания в очередях и могут получать информацию о своих записях и услугах быстрее, что также улучшает общий опыт обращения в поликлинику.

В итоге, экономическая эффективность внедрения сайта поликлиники складывается из снижения затрат, увеличения числа обращений и улучшения качества обслуживания, что вместе приводит к повышению доходов и оптимизации работы учреждения.

# **Выводы по третьей главе**

В третьей главе углубленно рассматриваются ключевые аспекты структуры и разработки программного продукта, которые имеют решающее значение для успешной реализации проекта. В начале главы акцент сделан на разработке интерфейса программного продукта. Здесь обсуждаются принципы проектирования пользовательского интерфейса, включая удобство, логичность и интуитивность. Упоминаются основные элементы дизайна, такие как расположение кнопок, выбор цветовой схемы и адаптивность под различные устройства, что обеспечивает высокий уровень взаимодействия с пользователем.

Следующий пункт посвящен рекомендуемым системным требованиям. Этот раздел включает информацию о необходимых характеристиках оборудования и программного обеспечения, которые обеспечивают стабильную работу разработанного приложения. Указываются минимальные и рекомендуемые параметры, что помогает потенциальным пользователям понять, какие технические условия необходимы для корректной работы программы.

Третий пункт охватывает обеспечение безопасности информации. Важность защиты данных в современных информационных системах трудно переоценить, поэтому в этой части главы анализируются различные аспекты безопасности, такие как шифрование данных, аутентификация пользователей и защита от несанкционированного доступа. Освещаются методы, которые будут использоваться для обеспечения защиты конфиденциальной информации пользователей и устойчивости системы к внешним угрозам.

Завершающий раздел посвящен определению экономической эффективности внедрения программного продукта. Здесь рассматриваются потенциальные преимущества от использования системы, такие как сокращение затрат, увеличение производительности и улучшение качества обслуживания клиентов. Приводятся методы оценки экономической целесообразности, что позволяет понять, каким образом проект может повлиять на финансовые показатели организации.

Таким образом, третья глава обобщает важные аспекты разработки программного продукта, от проектирования интерфейса до экономической оценки, что создает полное представление о процессе разработки и внедрения системы, а также подчеркивает ее значимость для достижения поставленных целей.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения выпускной квалификационной работы был реализован одностраничный сайт для поликлиники, включающий все основные разделы, необходимые для информирования пациентов и организации работы учреждения в современном цифровом формате.

В разделе "О поликлинике" предоставлена подробная информация о медицинском учреждении, его истории, специализации, оказываемых услугах, режиме работы, что позволяет посетителям быстро получить представление о возможностях и уровне обслуживания в поликлинике. Этот раздел способствует формированию доверия со стороны новых и постоянных пациентов.

Раздел "Наши врачи" содержит фотографии и ФИО специалистов разных направлений, благодаря чему пациенты могут заранее ознакомиться с врачами, что положительно сказывается на уровне доверия и повышает готовность записаться на приём. Фотографии врачей и краткая информация о них делают взаимодействие с сайтом более личным и открытым.

Особое внимание было уделено функционалу записи на прием. Форма записи разработана таким образом, чтобы максимально упростить процедуру: пользователю достаточно лишь указать своё имя и контактный телефон, выбрать интересующего врача из предлагаемых вариантов, а также выбрать дату и время визита. После заполнения формы данные отправляются для дальнейшей обработки администрацией клиники, что ускоряет и облегчает процесс записи. Применение валидации данных снижает вероятность ошибок при заполнении и повышает точность информации.

Раздел "Отзывы пациентов" реализован для формирования положительного имиджа поликлиники и повышения её прозрачности. Возможность ознакомиться с опытом других пациентов способствует укреплению доверия, а публикуемые отзывы позволяют узнать о сильных сторонах работы учреждения и качестве оказываемых услуг глазами посетителей.

Раздел "Контакты" содержит всю необходимую информацию для связи с поликлиникой: телефон, адрес, электронную почту и карту с местоположением, что облегчает навигацию и позволяет быстро связаться с администрацией учреждения для решения всех возникающих вопросов.

В техническом плане сайт создан с использованием современных технологий фронтенд-разработки, что обеспечивает зону его адаптивности — корректная работа и привлекательное отображение сайта на различных устройствах, будь то ПК, ноутбук, планшет или мобильный телефон. Такой подход гарантирует удобство использования для максимально широкого круга пользователей. В процессе разработки также учитывались вопросы безопасности, надёжной обработки данных и защиты персональной информации пользователей.

В результате проделанной работы все поставленные задачи реализованы в полном объёме: созданный сайт обладает необходимой для современного медицинского учреждения функциональностью, прост в использовании, интуитивно понятен и готов к внедрению в реальную работу. Это позволит повысить цифровую доступность услуг поликлиники и улучшить качество обратной связи с пациентами, сделать процесс записи на приём и получения информации более быстрым, комфортным и современным.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 273 с. — (Профессиональное образование). — ISBN 978-5-534-20362-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562355> (дата обращения: 23.05.2025).
2. Грекул, В. И.  Проектирование информационных систем : учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 404 с. — (Профессиональное образование). — ISBN 978-5-534-19506-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566739> (дата обращения: 23.05.2025).
3. Зараменских, Е. П.  Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 119 с. — (Профессиональное образование). — ISBN 978-5-534-21417-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/571330> (дата обращения: 23.05.2025).
4. Григорьев, М. В.  Проектирование информационных систем : учебник для среднего профессионального образования / М. В. Григорьев, И. И. Григорьева. — Москва : Издательство Юрайт, 2025. — 278 с. — (Профессиональное образование). — ISBN 978-5-534-16847-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566741> (дата обращения: 23.05.2025).
5. Зараменских, Е. П.  Информационные системы: управление жизненным циклом : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 486 с. — (Профессиональное образование). — ISBN 978-5-534-21416-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/571329> (дата обращения: 23.05.2025).
6. Зыков, С. В.  Архитектура информационных систем. Основы проектирования : учебник для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2025. — 260 с. — (Высшее образование). — ISBN 978-5-534-21538-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/575500> (дата обращения: 23.05.2025).
7. Полуэктова, Н. Р.  Разработка веб-приложений : учебник для среднего профессионального образования / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-18644-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567621> (дата обращения: 23.05.2025).
8. Тузовский, А. Ф.  Проектирование и разработка web-приложений : учебник для среднего профессионального образования / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Профессиональное образование). — ISBN 978-5-534-16767-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/565693> (дата обращения: 23.05.2025).
9. Сысолетин, Е. Г.  Разработка интернет-приложений : учебник для среднего профессионального образования / Е. Г. Сысолетин, С. Д. Ростунцев. — Москва : Издательство Юрайт, 2025. — 80 с. — (Профессиональное образование). — ISBN 978-5-534-19603-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/565692> (дата обращения: 23.05.2025).
10. Советов, Б. Я.  Базы данных : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 403 с. — (Профессиональное образование). — ISBN 978-5-534-18784-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563146> (дата обращения: 23.05.2025).
11. Щербак, А. В.  Поддержка и тестирование программных модулей : учебник для среднего профессионального образования / А. В. Щербак. — Москва : Издательство Юрайт, 2025. — 145 с. — (Профессиональное образование). — ISBN 978-5-534-19290-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/580603> (дата обращения: 23.05.2025).
12. Казарин, О. В.  Надежность и безопасность программного обеспечения : учебник для вузов / О. В. Казарин, И. Б. Шубинский. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 352 с. — (Высшее образование). — ISBN 978-5-534-19386-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/580669> (дата обращения: 23.05.2025).

# **ПРИЛОЖЕНИЕ А**

Диаграмма деятельности сайта

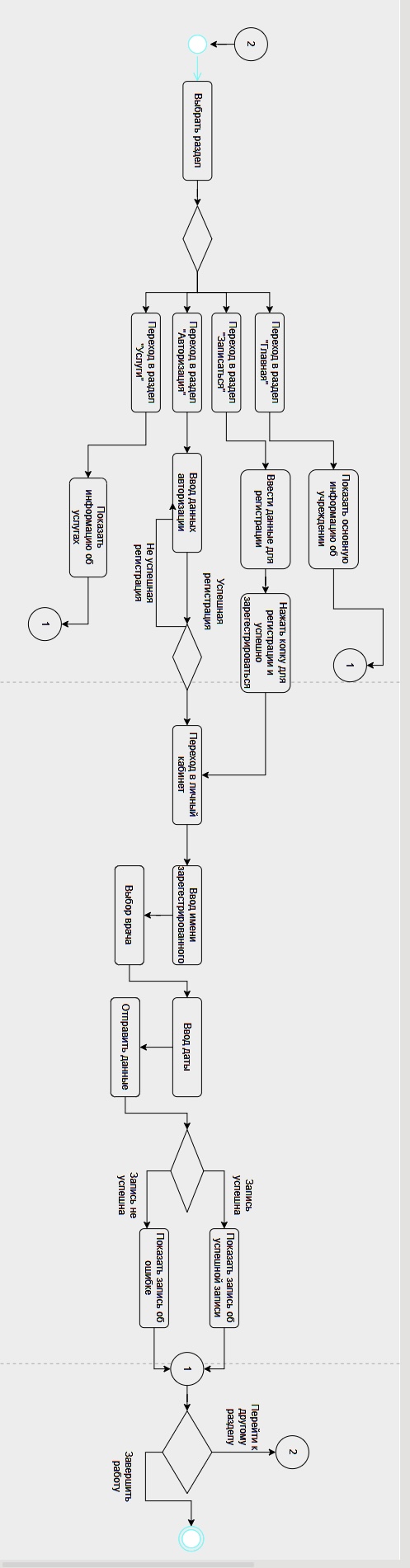
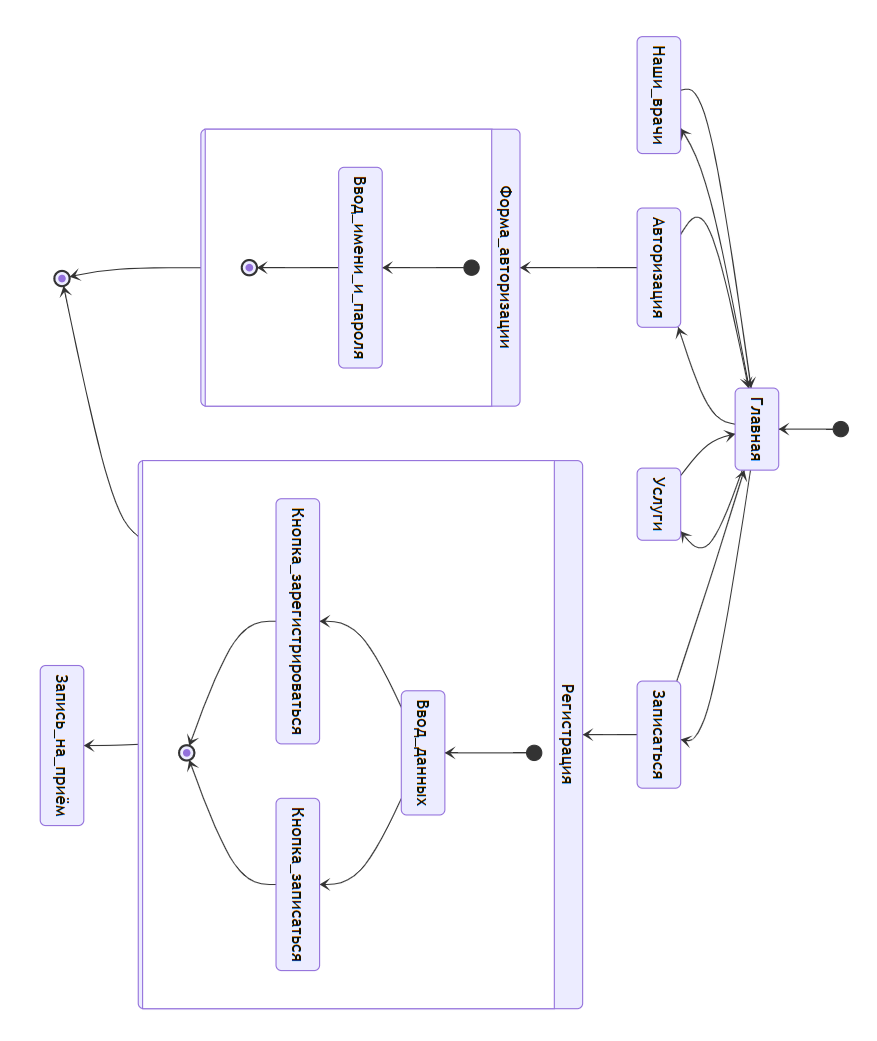


Диаграмма состояний сайта



# **ПРИЛОЖЕНИЕ Б**

Код файла index.php

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="/css/style.css">

    <title>Поликлиника</title>

</head>

<body>

    <!-- Навигационная панель -->

    <?php require\_once "blocks/header.php";?>

    <!-- О поликлинике -->

     <div class="about-section" id="about">

        <h1>О Поликлинике</h1>

        <p class="description">

            Наша поликлиника предоставляет широкий спектр медицинских услуг, обеспечивая комфорт и

            качество лечения для всех наших пациентов. Мы стремимся к улучшению здоровья

            и благополучия наших клиентов.

        </p>

        <h2>Миссия и ценности</h2>

        <p class="mission">

            Миссия нашей поликлиники - предоставить высококачественные медицинские услуги в

            доступной форме, основываясь на принципах уважения, заботы и профессионализма.

        </p>

        <ul class="values">

            <li>Профессионализм</li>

            <li>Доступность</li>

            <li>Человечность</li>

            <li>Инновации</li>

        </ul>

        <h2>Фото</h2>

        <div class="gallery">

            <img src="/files/photo/photo\_hospital.jpg" alt="Здание поликлиники" class="gallery-item">

            <img src="/files/photo/team.jpg" alt="Команда врачей" class="gallery-item">

        </div>

    </div>

    <!-- Врачи: -->

     <div class="doctors-section" id="doctors">

        <h1>Наши Врачи</h1>

        <div class="carousel">

            <div class="carousel-item">

                <img src="/files/photo/doctor1.jpg" alt="Врач 1" class="doctor-image">

                <div class="doctor-info">

                    <h3>Анна Петрова</h3>

                    <p>Терапевт</p>

                    <p>Стаж: 10 лет</p>

                </div>

            </div>

            <div class="carousel-item">

                <img src="/files/photo/doctor2.jpg" alt="Врач 2" class="doctor-image">

                <div class="doctor-info">

                    <h3>Игорь Сидоров</h3>

                    <p>Хирург</p>

                    <p>Стаж: 15 лет</p>

                </div>

            </div>

            <div class="carousel-item">

                <img src="/files/photo/doctor3.jpg" alt="Врач 3" class="doctor-image">

                <div class="doctor-info">

                    <h3>Елена Васильева</h3>

                    <p>Педиатр</p>

                    <p>Стаж: 8 лет</p>

                </div>

            </div>

        </div>

        <!-- <button class="appointment-button">Записаться к врачу</button> -->

    </div>

    <script>

        // Простой скрипт для карусели

        let currentIndex = 0;

        const items = document.querySelectorAll('.carousel-item');

        const totalItems = items.length;

        function showItem(index) {

            items.forEach((item, i) => {

                item.style.display = (i === index) ? 'block' : 'none';

            });

        }

        showItem(currentIndex);

        setInterval(() => {

            currentIndex = (currentIndex + 1) % totalItems;

            showItem(currentIndex);

        }, 3000);

    </script>

    <!-- Отзывы пациентов -->

     <div class="reviews-section" id="reviews">

        <h1>Отзывы пациентов</h1>

        <div class="review">

            <p class="quote">"Отличное обслуживание! Врачи очень внимательные."</p>

            <div class="rating">

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star">☆</span>

            </div>

            <p class="author">— Анна К.</p>

        </div>

        <div class="review">

            <p class="quote">"Чувствую себя гораздо лучше благодаря этому центру."</p>

            <div class="rating">

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star filled">★</span>

            </div>

            <p class="author">— Игорь П.</p>

        </div>

        <div class="review">

            <p class="quote">"Комфортная обстановка и внимательные специалисты."</p>

            <div class="rating">

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star filled">★</span>

                <span class="star">☆</span>

                <span class="star">☆</span>

            </div>

            <p class="author">— Елена С.</p>

        </div>

    </div>

    <!-- Контакты: -->

     <div class="contacts-section" id="contacts">

        <h1>Контакты</h1>

        <div class="contact-item">

            <h2>Адрес</h2>

            <p>г. Москва, ул. Примерная, д. 1</p>

        </div>

        <div class="contact-item">

            <h2>Телефон</h2>

            <p><a href="tel:+74951234567">+7 (495) 123-45-67</a></p>

        </div>

        <div class="contact-item">

            <h2>Электронная почта</h2>

            <p><a href="mailto:info@poliklinika.ru">info@poliklinika.ru</a></p>

        </div>

        <div class="contact-item">

            <h2>Часы работы</h2>

            <p>Пн-Пт: 8:00 - 20:00</p>

            <p>Сб: 9:00 - 15:00</p>

            <p>Вс: Выходной</p>

        </div>

    </div>

    <!-- Подвал -->

     <?php require\_once "blocks/footer.php";?>

</body>

</html>

**Код файла reg.php**

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="/css/style.css">

    <title>Регистрация</title>

</head>

<body>

    <!-- Навигационная панель -->

    <?php require\_once "blocks/header.php";?>

        <!-- Форма для записи: -->

     <div class="appointment-section" id="appointment">

        <h1>Регистрация</h1>

        <form id="appointmentForm" method="post" action="../lib/reg.php">

            <label for="name">Имя:</label>

            <input type="text" id="name" name="name" required>

            <label for="phone">Телефон:</label>

            <input type="tel" id="phone" name="phone" required>

            <label for="mail">Почта</label>

            <input type="mail" id="mail" name="mail" required>

            <label for="password">Пароль</label>

            <input type="password" id="password" name="password" required>

            <button type="submit" class="submit-button">Зарегистрироваться</button>

        </form>

    </div>

    <!-- Контакты: -->

     <div class="contacts-section" id="contacts">

        <h1>Контакты</h1>

        <div class="contact-item">

            <h2>Адрес</h2>

            <p>г. Москва, ул. Примерная, д. 1</p>

        </div>

        <div class="contact-item">

            <h2>Телефон</h2>

            <p><a href="tel:+74951234567">+7 (495) 123-45-67</a></p>

        </div>

        <div class="contact-item">

            <h2>Электронная почта</h2>

            <p><a href="mailto:info@poliklinika.ru">info@poliklinika.ru</a></p>

        </div>

        <div class="contact-item">

            <h2>Часы работы</h2>

            <p>Пн-Пт: 8:00 - 20:00</p>

            <p>Сб: 9:00 - 15:00</p>

            <p>Вс: Выходной</p>

        </div>

    </div>

    <!-- Подвал -->

     <?php require\_once "blocks/footer.php";?>

</body>

</html>