1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПОЛИКЛИНИКА

Поликлиника — это многопрофильное или специализированное лечебно-профилактическое учреждение для оказания амбулаторной медицинской помощи больным на приёме и на дому.

Это учреждение первичной медико-санитарной помощи.

На территории России поликлиники распределены по территориальному признаку и являются базовым уровнем оказания медицинской помощи населению.

Поликлиника является самостоятельной медицинской организацией или структурным подразделением медицинской организации (ее структурного подразделения), оказывающей первичную медико-санитарную помощь, и организуется для оказания первичной доврачебной медико-санитарной помощи, первичной врачебной медико-санитарной помощи, первичной специализированной медико-санитарной помощи, неотложной, а также паллиативной медицинской помощи населению.

Основными задачами и функциями Учреждения являются:

* оказание первичной (доврачебной, врачебной), специализированной медико- санитарной помощи, в том числе в неотложной форме, больным, проживающим на территории обслуживания и (или) прикрепленным на обслуживание, при острых заболеваниях, травмах, отравлениях и других неотложных состояниях:
* проведение профилактических мероприятий по предупреждению и снижению заболеваемости, выявление ранних и скрытых форм заболеваний, социально значимых заболеваний и факторов риска:
* проведение диспансеризации населения;
* диагностика и лечение различных заболеваний и состояний:
* восстановительное лечение и реабилитация:
* клинико-экспертная деятельность по оценке качества и эффективности лечебных и диагностических мероприятий:
* экспертиза временной нетрудоспособности, выдача и продление листков нетрудоспособности и направление граждан на медико-социальную экспертизу:
* диспансерное наблюдение за состоянием здоровья лиц. страдающих хроническими заболеваниями, в том числе отдельных категорий граждан, имеющих право на получение набора социальных услуг, функциональными расстройствами, иными состояниями, с целью своевременного выявления (предупреждения) осложнений, обострений заболеваний, иных патологических состояний, их профилактики и осуществления медицинской реабилитации:
* организация дополнительной бесплатной медицинской помощи, в том числе необходимыми лекарственными средствами, отдельным категориям граждан:
* установление медицинских показаний и направление в медицинские организации для получения специализированных видов медицинской помощи;
* организация и оказание паллиативной помощи больным, в том числе больным онкологическими заболеваниями, нуждающимся в наркотических и сильнодействующих лекарственных средствах в соответствии с рекомендациями врачей-специалистов:
* проведение всех видов медицинских осмотров (профилактические. предварительные, периодические).
  1. ЦЕЛЕВАЯ АУДИТОРИЯ «ПОЛИКЛИНИКИ»

Целевая аудитория медицинского центра может включать различные группы людей, в зависимости от предоставляемых услуг. К ним могут относиться:

* Взрослые мужчины и женщины, нуждающиеся в общей медицинской помощи;
* Пациенты с конкретными заболеваниями или хроническими состояниями;
* Женщины, ожидающие ребенка или нуждающиеся в акушерско-гинекологической помощи;
* Дети, требующие педиатрического обслуживания;
* Пациенты, нуждающиеся в стоматологической помощи;
* Спортсмены и активные люди, ищущие услуги по физиотерапии или спортивной медицине.

Социально-демографические характеристики целевой аудитории медицинского центра могут варьироваться в зависимости от местоположения и специализации центра. Среди них могут быть следующие:

* Возраст: от молодых взрослых до пожилых людей;
* Пол: мужчины и женщины;
* Образование: от неполного среднего до высшего образования;
* Доход: различные социально-экономические группы;
* Местоположение: жители города или пригорода;
* Семейный статус: одинокие, замужние/женатые, семьи с детьми;
* Предпочтения в области здоровья и медицины.

Психологический анализ целевой аудитории медицинского центра позволяет понять, какие мотивации и потребности могут привлечь клиентов. Некоторые общие характеристики психологического анализа могут включать:

* Стремление к сохранению и улучшению своего здоровья;
* Опасения и тревожность по поводу состояния здоровья;
* Желание получить профессиональную и квалифицированную медицинскую помощь;
* Потребность в доброжелательном, эмпатическом и внимательном отношении со стороны медицинского персонала;
* Приверженность к натуральным и альтернативным методам лечения;
* Потребность в комфорте и удобстве при посещении медицинского центра.

Поведенческий анализ целевой аудитории медицинского центра помогает определить, как клиенты ищут и выбирают медицинские услуги, а также какие факторы влияют на их принятие решения о посещении центра. Некоторые характерные аспекты поведенческого анализа могут быть следующими:

* Использование интернета для поиска медицинских услуг и отзывов о медицинских центрах;
* Предпочтение рекомендаций и рецензий от друзей и семьи;
* Заключение медицинской страховки и возможность ее использования в данном центре;
* Расположение и доступность практики;
* Внешний вид и качество услуги;
* Цена и наличие акций или скидок.

1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. ВЫБОР ИНСТРУМЕНТАЛЬНЫХ И ПРОГРАММНЫХ СРЕДСТВ РАЗРАБОТКИ ВЕБ ПРИЛОЖЕНИЯ

2.1 ВЫБОР ЯЗЫКА И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

На начальных этапах процесса проектирования программного продукта необходимо принять принципиальные решения, во многом определяющие этот процесс, а также качество и трудоемкость разработки. К таким решениям относят:

– выбор архитектуры программного обеспечения;

– выбор типа пользовательского интерфейса;

– выбор подхода к разработке;

– выбор языка и среды программирования.

Языки веб-разработки - это языки программирования и технологии, используемые для создания веб-сайтов, веб-приложений и компонентов серверной части. Они составляют основу веб-разработки и определяют характер взаимодействия пользователей с веб-контентом. Существует два основных аспекта веб-разработки:

Front-End Development: Включает в себя проектирование и реализацию визуальных элементов, которые видят пользователи и с которыми они взаимодействуют в веб-приложении. Разработчики фронт-энда используют комбинацию HTML, CSS и JavaScript для создания визуально привлекательного и функционального пользовательского интерфейса.

Back-End Development: Работает с компонентами на стороне сервера и манипулирует данными, обеспечивая их хранение, обработку и получение. Это обеспечивает бесперебойную работу веб-приложения и его масштабирование при необходимости. К распространенным языкам back-end относятся Python, PHP, Ruby, Java и C#.

Для Front-End Development в качестве языка будет использоваться JavaScript как один из самый лидирующих языков в Front-End разработке.

Кроме языка программирования требуется выбрать используемые фреймворк.

Фреймворк (англ. framework — «каркас, структура») — это готовый набор инструментов, который помогает разработчику быстро создать продукт: сайт, приложение, интернет-магазин, CMS-систему.

В таблице 1 представлено сравнение Фреймворк для Front-End разработки.

Таблица 1 - сравнение Фреймворк для Front-End разработки.

|  |  |  |  |
| --- | --- | --- | --- |
| Фреймворк | Производительность | Сложность | Комьюнити |
| Angular | - | Сложный | + |
| Vue | + | Простой | - |
| React | + | Средней | + |
| Svelte | + | Простой | - |

В качестве выбора это не будет Angular из-за своей сложности, не Vue из-за не большого компьюнити что может привести к проблеме отсутствию информации о какой-либо проблеме, не Svelte из-за аналогичной проблемы что он так же вышел не давно и не прошел проверку временем.

Остается React, проверенный временем и наибольшим компьюнити что дает возможность найди ответ на любой вопрос, который так же прост в понимании.

2.2 ВЫБОР СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Система управления базами данных (СУБД) — это программное обеспечение, предназначенное для создания, управления, обновления и анализа баз данных. Она обеспечивает интерфейс для взаимодействия пользователя или приложения с данными, хранящимися в базе данных. СУБД позволяют структурировать данные таким образом, чтобы обеспечить их легкий доступ, безопасность и эффективное использование.

СУБД — комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать). Система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД.

В таблице 2 представлено сравнение СУБД.

Таблица 2 - сравнение СУБД.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СУБД | Бесплатность | Размер базы | Размер таблицы | Число пользователей |
| HSQLDB | нет | 28 ТB | 120 GB | Не ограничено |
| Microsoft SQL Server | нет | 16 ТВ | 532 GB | Не ограничено |
| MySQL | да | 256 TB | 256 ТB | Не ограничено |
| PostgreSQL | да | Неограничен | 32 TB | Не ограничено |

Так как количество обращение всегда будет увеличивать и их требуется хранить для документирования обращении то лучшей СУБД является PostgreSQL который не имеет ограничений на размере базы данных, так же является бесплатной СУБД, так же имеет возможность быстрого чтения среди множество данных и неограниченный размер хранение индексов.

PostgreSQL заработал прочную репутацию благодаря своей проверенной архитектуре, надежности, целостности данных, надежному набору функций, расширяемости и преданности сообщества открытого исходного кода, стоящего за программным обеспечением, для последовательного предоставления производительных и инновационных решений. PostgreSQL работает во всех основных операционных системах, совместим с ACID с 2001 года и имеет мощные надстройки, такие как популярный расширитель геопространственных баз данных PostGIS. Неудивительно, что PostgreSQL стала предпочитаемой реляционной базой данных с открытым исходным кодом для многих людей и организаций.

2.3 ВЫБОР СРЕДСТВ РАЗРАБОТКИ ВЕБ СЕРВЕРА

Backend (бэкенд) – серверная часть сайта. Отвечает за быструю загрузку страниц, обработку данных, безопасность, интеграцию с другими системами. Пользователь не видит всего этого, но благодаря backend веб-продукт или ПО нормально функционируют.

Backend-разработчик пишет код, организует хранение и передачу данных. Создает и поддерживает механизмы и алгоритмы, которые позволяют посетителям взаимодействовать со страницей. Например, делает так, чтобы платежи в интернет-магазине были безопасными.

В таблице 3 представлено сравнение фрейморков для разработки веб сервиса.

Таблица 3 - сравнение фрейморков для разработки веб сервиса.

|  |  |  |  |
| --- | --- | --- | --- |
| Фреймворк | Сложность | Документация | Производительность |
| Nest(js) | Средний | + | + |
| Django(python) | Легкий | - | - |
| Spring(java) | Сложный | - | + |

Из представленных Фреймворков выбран Nest(js) из-за своей простоты и хорошо написанной документации в сравнение с др. фреймворками, так же используется тот же язык что и в frond-end разработке что упрощает разработку.

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВЕБ ПРИЛОЖЕНИЯ

Веб приложение состоит из 2 модулей frond-end и back-end приложения, которые общаются между собой при помощи сетевого протокола http/ https.

Backend и frontend взаимодействуют между собой по следующему принципу. Пользователь попадает на Web-сайт и взаимодействует с пользовательским интерфейсом. Нажав на какую-либо кнопку frontend отправляет запрос на серверную сторону сервиса. Backend принимает, обрабатывает запрос и придает запросу определенный вид. Обработанный запрос возвращается на сайт и отображается пользователю в виде frontend.

В Backend используется ORM typeORM с помощью которого осуществляется взаимодействие с базой данных и swagger для автоматического документирование проекта.

Структура Backend приложения в папке src где хранится весь код, в корне находятся модули, в которых описан код контроллеров, сервисов, сущностей.

На рисунке 1 представлена директива src

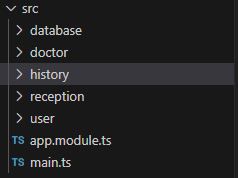


Рисунок 1 - директива src

main.ts – точка входа приложения.

database – директива с конфигурацией подключение к бд.

doctor – директива с логикой для взаимодействия с врачами;

user – директива с логикой для взаимодействия с пользователями;

history – директива с логикой для взаимодействия с историей книжки болезней пациентов;

reception – директива с логикой для записи на прием к врачу.

Ниже представлены конечные точки с помощью swagger.

На рисунке 2 представлено api doctor.

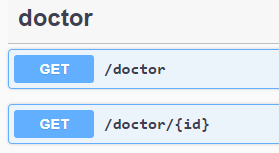


Рисунок 2 - api doctor

На рисунке 3 представлены api reception.

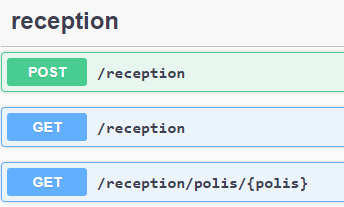


Рисунок 3 - api reception

На рисунке 3 представлены api user.

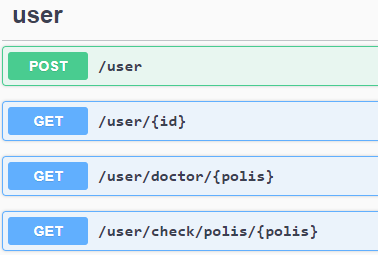


Рисунок 3 - api user

На рисунке 4 представлены api history.

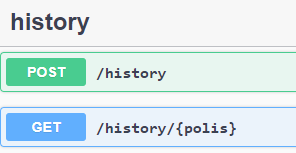


Рисунок 4 - api history

Веб приложение представляет собой следующие страницы:

* главная – общая информация об компании
* врачи – список врачей в поликлинике;
* запись – страница с помощью, которой можно записаться к врачу на прием;
* история болезни – страница, где пациент можно просмотреть свою историю болезни;
* добавить запись болезни – страница где врач может добавить запись в больничную книжку пациента;
* просмотр приемов – страница где пользователь может увидеть, историю свои приемов к врачу.

На рисунке 5 представлена реализация страницы главная.

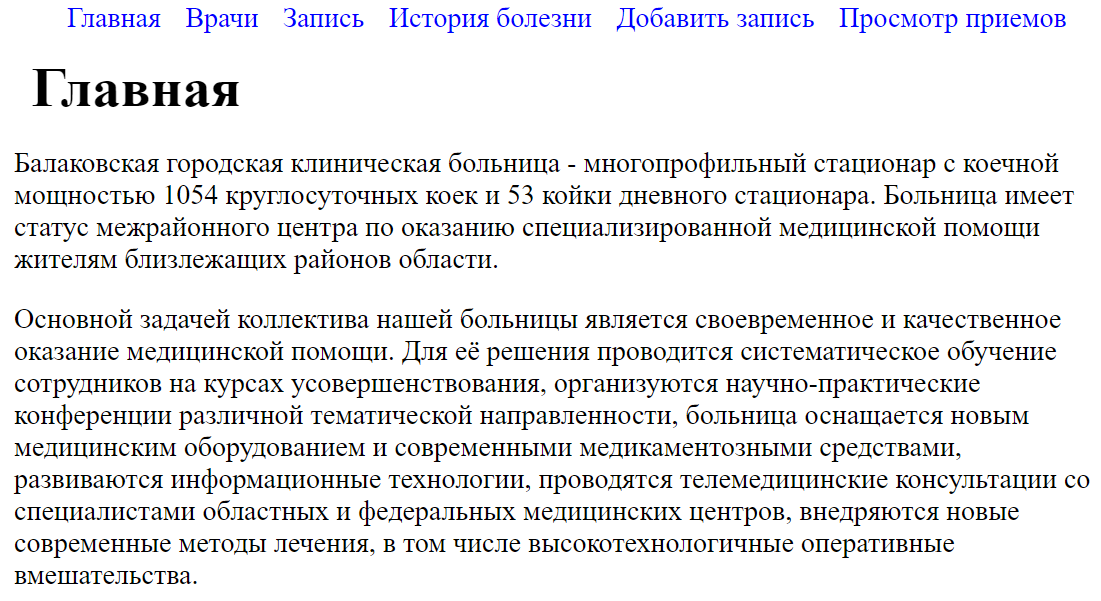


Рисунок 5 - Страница главная

На рисунке 6 представлена реализация страницы врачи.

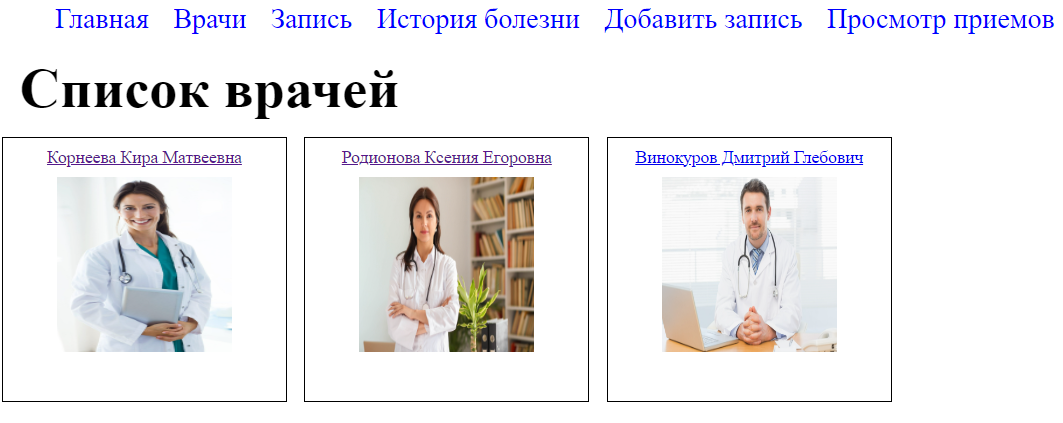


Рисунок 6 - Страница врачи

На рисунке 7 представлена реализация страницы запись к врачу.

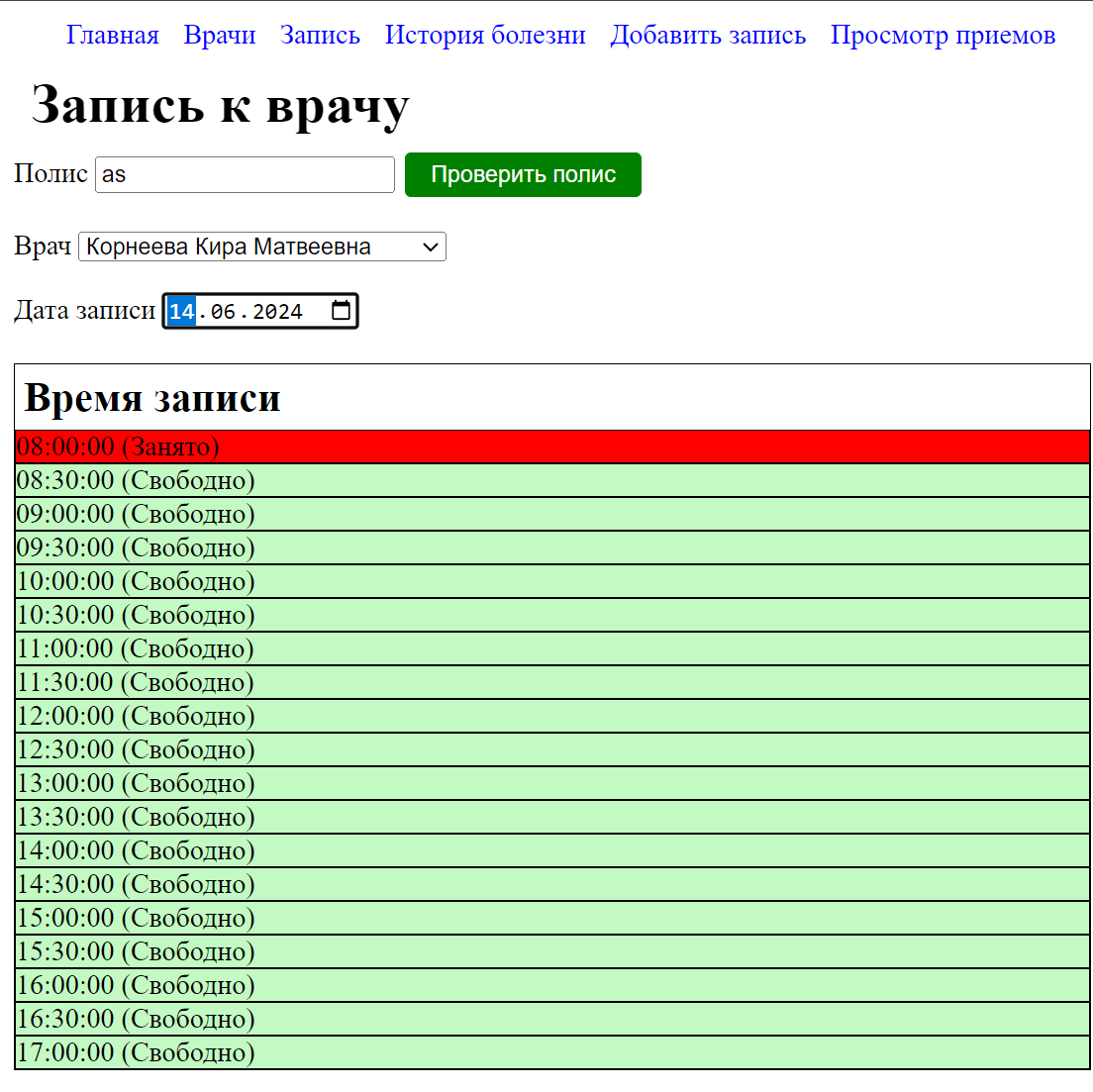


Рисунок 7 - Страница запись к врачу

На рисунке 8 представлена реализация страницы история болезни.

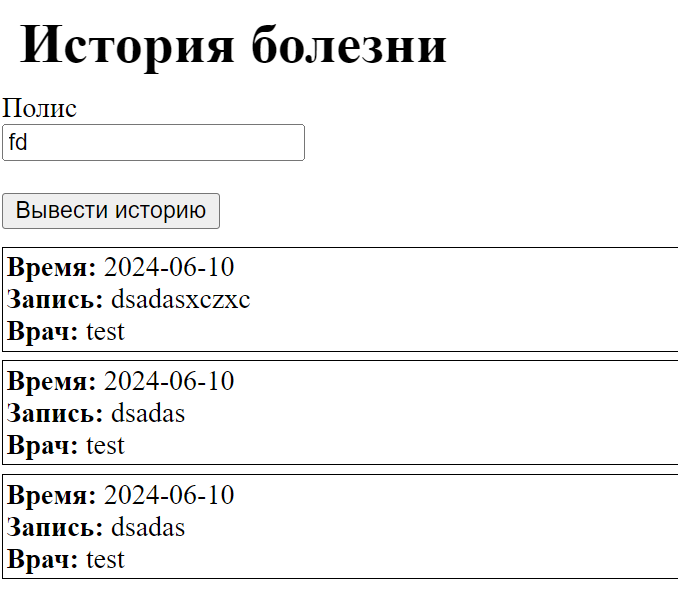


Рисунок 8 - Страница история болезни

На рисунке 9 представлена реализация страницы добавить запись.

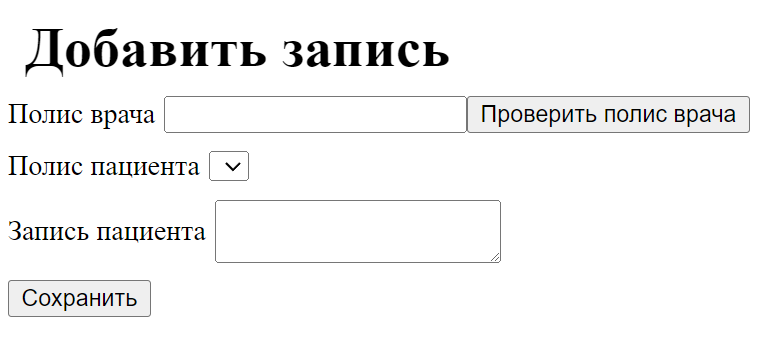


Рисунок - 9 страницы добавить запись.

На рисунке 10 представлена реализация страницы добавить просмотр приемов.

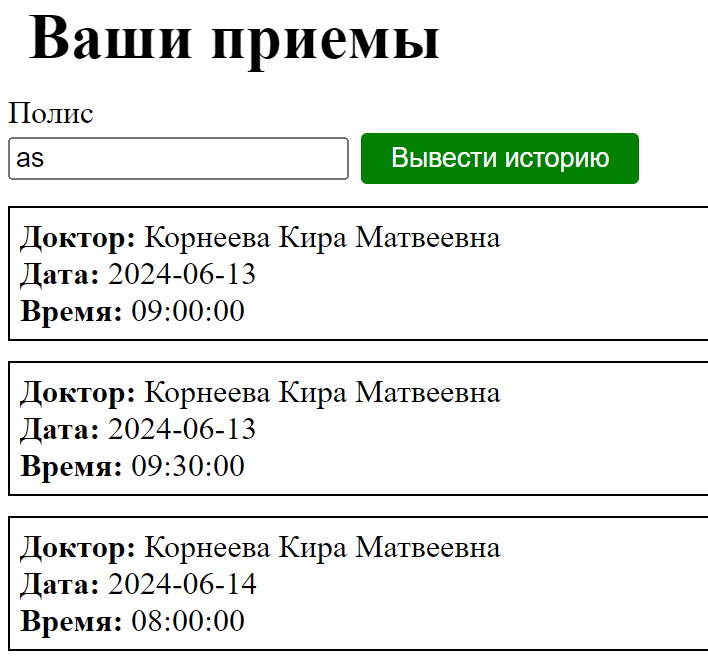


Рисунок - 10 страницы просмотр приемов.