ВВЕДЕНИЕ

Информационный портал — крупный веб-сайт, организованный как многоуровневое объединение различных ресурсов и сервисов, обновление которых происходит в реальном времени.

Информационный портал содержит большое количество контента, как правило, уникального и рассчитан на большие нагрузки по посещаемости. Сайты такого типа имеют сложную структуру и навигацию и содержат различные интерактивные сервисы. Информационные порталы предназначены для большого количества посетителей.

Целью курсовой работы является применение на практике знаний, полученных в процессе изучения курса «Современные технологии интернет-программирования».

Задачами курсовой работы являются:

* анализ предметной области;
* формирование понимания назначения информационных порталов;
* умения производить обоснованный выбор программного обеспечения для разработки портала;
* разрабатывать структуру веб-ресурса, дизайн-макет веб-ресурса, разрабатывать модули веб ресурса.

В курсовой работе студент должен показать степень освоения теоретическим материалом, степень понимания назначения информационных порталов, умение производить обоснованный выбор программного обеспечения для разработки портала.

1 Анализ предметной области

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 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).

1.1 Целевая аудитория

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. Техническое задание

Разрабатываемый продукт предназначен для целевой аудитории предоставляя возможность взаимодействия с врачами.

Продукт представляет собой 2 веб приложении front-end и back-end, которые будут общаться между собой по протоколу https, front-end будет передать на back-end данные что делает пользователей, после чего back-end будет обрабатывать эти данные и возвращать front-end ответ.

Продукт предоставляет возможности:

* просмотреть свою историю болезни;
* создать пользователя на основе полиса;
* просмотреть историю приемов;
* просмотреть список врачей в поликлинике;
* возможность записаться к врачу на конкретную дату и время;
* добавить врачу запись в историю болезни.

Back-end –работает с компонентами на стороне сервера и манипулирует данными, обеспечивая их хранение, обработку и получение

front-end – это презентационная часть web-приложений, информационной или программной системы, её пользовательский интерфейс и связанные с ним компоненты.

Протокол https - протокол, который обеспечивает целостность и конфиденциальность данных при их передаче между сайтом и устройством пользователя.

Веб приложение - это компьютерная программа, которая для выполнения своих функций использует веб-браузеры и веб-технологии в сети Интернет.

Продукт должен предоставлять врачам и клиентов возможность удобнее взаимодействовать между собой, давай возможность пациентом в любое время записаться на прием или получить интересующую информацию.

Существует 3 ключевых критериев, характеризующих уровень юзабилити ресурса:

* простота. Насколько легко новому посетителю ориентироваться в пространстве и выполнять элементарные действия.
* эффективность. Сколько времени требуется на достижение цели.
* запоминаемость. Как быстро пользователь адаптируется к нюансам работы с сайтом, сможет ли оперативно разобраться при повторных посещениях.

Требования к производительности:

* быстродействие сайта;
* ожидание до 1 клика;

Требование к логической структуре БД:

* база данных должна включать совокупность информационных объектов различного типа.
* каждый информационный объект может иметь набор атрибутов и связей с другими информационными объектами.
* информационные объекты связаны друг с другом иерархически.
* при внесении изменений структура БД должна обеспечивать автоматическое изменение всех объектов и связей, затрагиваемых таким изменением.

Ограничения проектирования:

* работает только в браузере;
* требуется доступа к интернету;
* работает по http протоколу;
* срок 1-2 месяца разработки.

1. Выбор инструментальных и программных средств разработки

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 и так же данный фреймворк 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 данных всегда будет увеличивать и их требуется хранить 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.

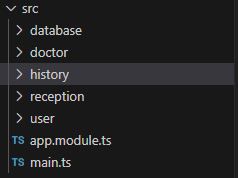


Рисунок 1 – директива src.

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.

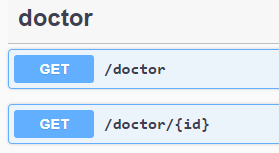


Рисунок 2 – api doctor.

На рисунке 3 представлены api reception.

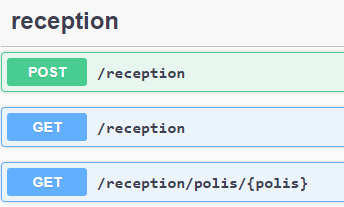


Рисунок 3 – api reception.

На рисунке 4 представлены api user.

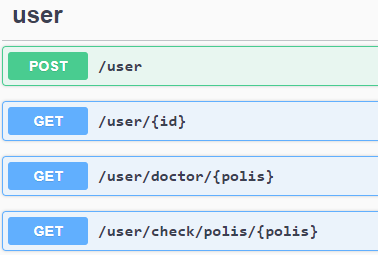


Рисунок 4 – api user.

На рисунке 5 представлены api history.

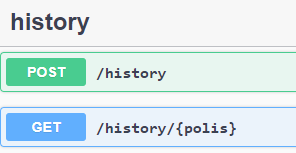


Рисунок 5 – api history.

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 6 представлена реализация страницы главная.

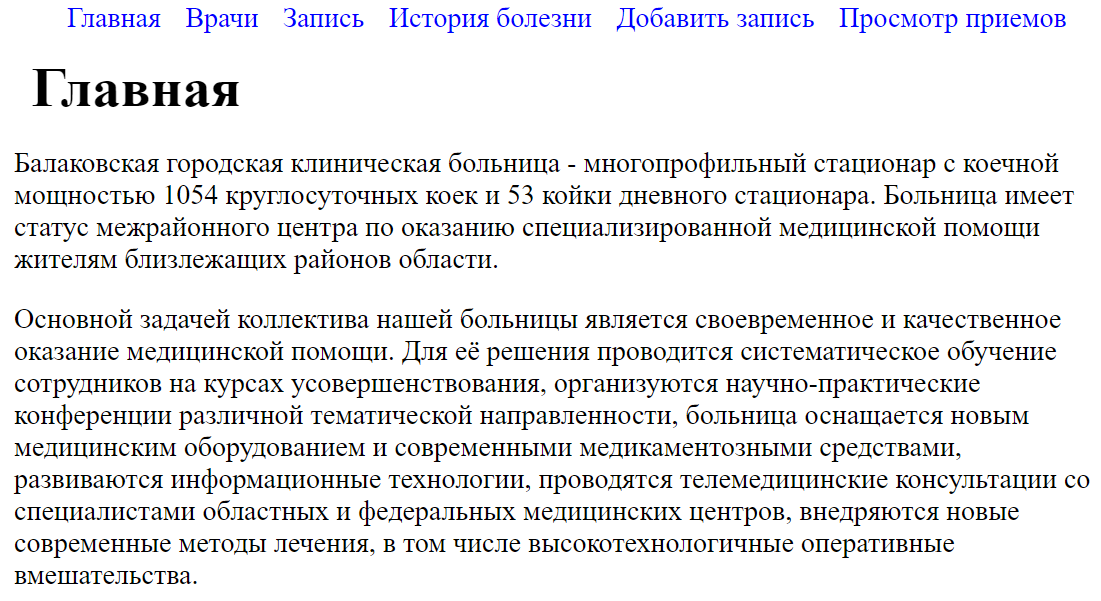


Рисунок 6 – Страница главная.

На рисунке 7 представлена реализация страницы врачи.

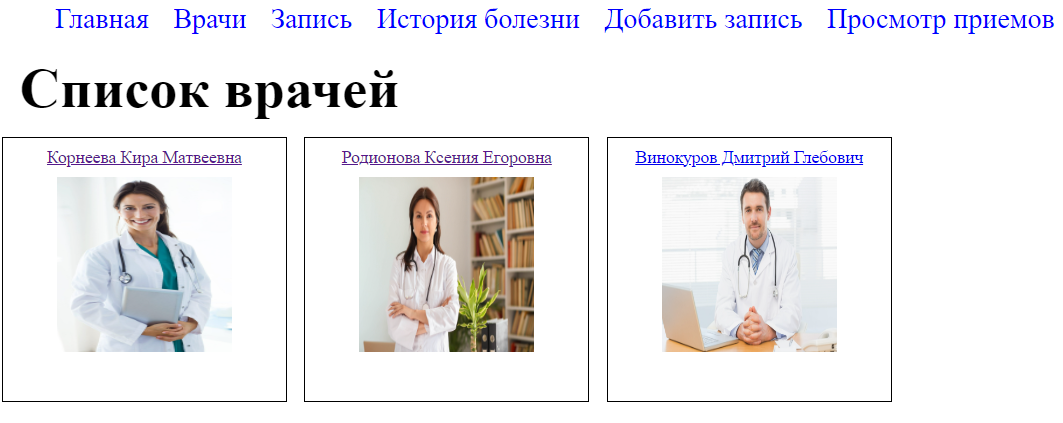


Рисунок 7 – Страница врачи

На рисунке 8 представлена реализация страницы запись к врачу.

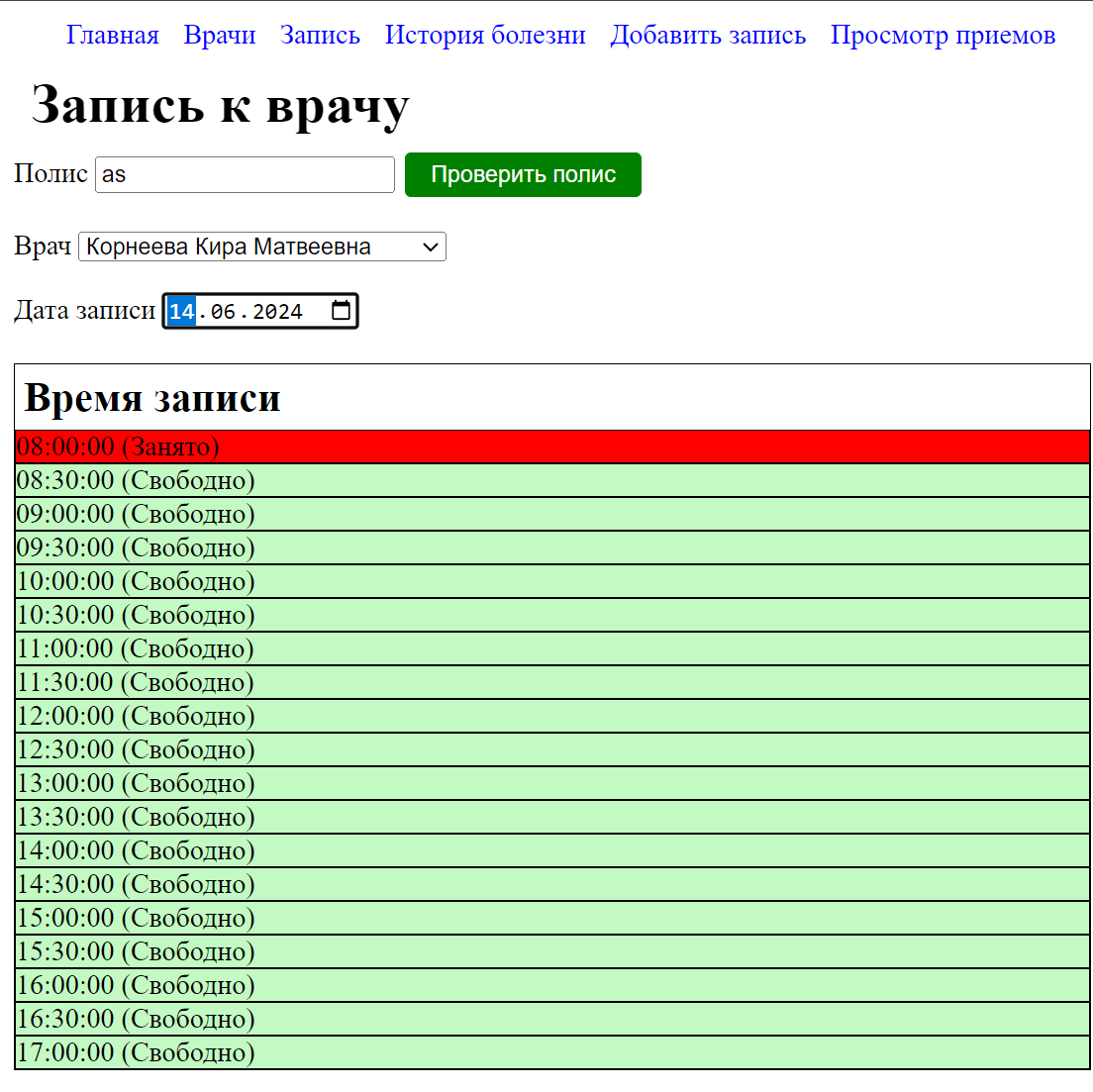


Рисунок 8 – Страница запись к врачу

На рисунке 9 представлена реализация страницы история болезни.

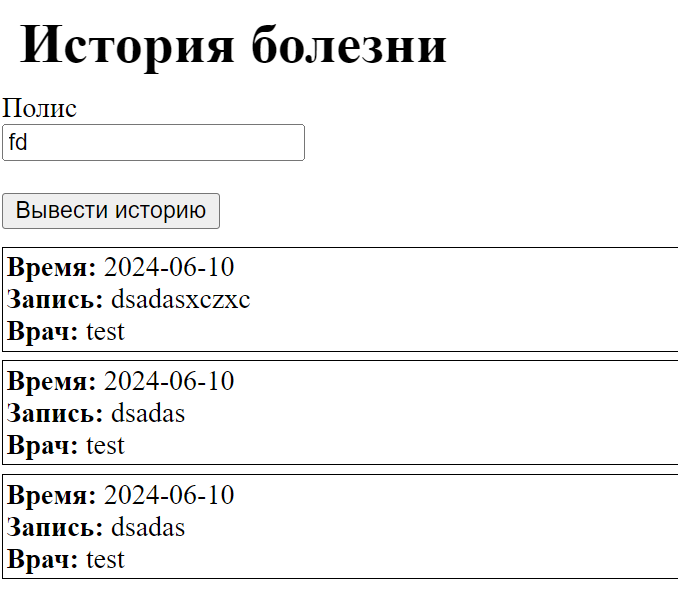


Рисунок 9 – Страница история болезни

На рисунке 10 представлена реализация страницы добавить запись.

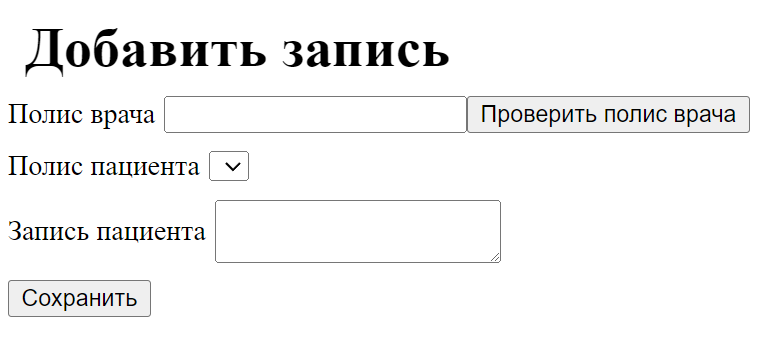


Рисунок – 10 страницы добавить запись.

На рисунке 11 представлена реализация страницы добавить просмотр приемов.

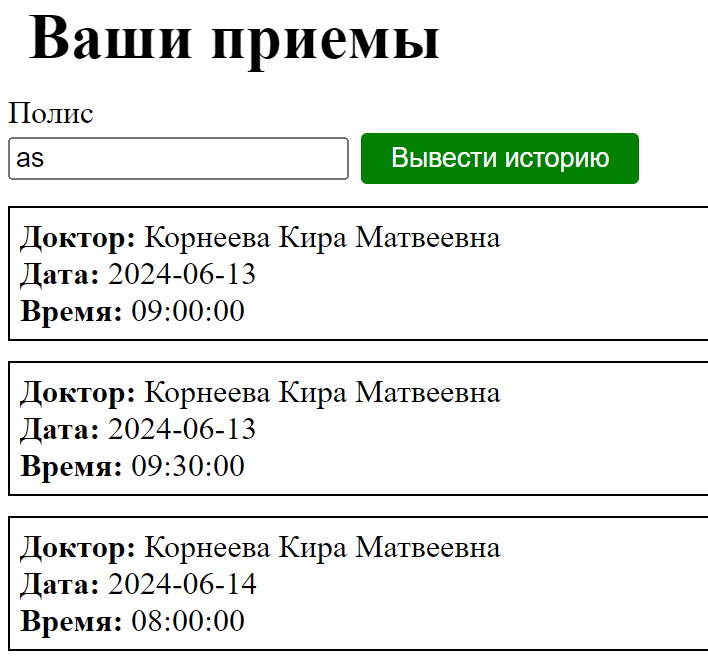


Рисунок – 11 страницы просмотр приемов.

ЗАКЛЮЧЕНИЕ

В ходе выполнение курсовой работы были выполненны следующий задачи:

* анализ предметной области «Поликлиника»;
* анализ целевой аудитории предметной области «Поликлиника»;
* разработано техническое задание;
* выбраны инструменты и ПО для разработки frond-end и back-end приложении;
* выбрана система управление базами данных;
* реализованы конечные точки api для back-end;
* реализован интерфейс веб приложения.

Цель курсовой работы достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лион, У. Разработка веб-приложений GraphQL с React, Node.js и Neo4j : практическое руководство / У. Лион ; пер. с англ. А. Н. Киселева. - Москва : ДМК Пресс, 2023. - 262 с. - ISBN 978-5-93700-185-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/2109522;

2. Солодушкин, С. И. Разработка программных комплексов на языке JavaScript : учебное пособие / С. И. Солодушкин, И. Ф. Юманова ; под общ. ред. В. Г. Пименова ; Министерство науки и высшего образования Россйской Федерации, Уральский федеральный университет. - Екатеринбург : Изд-во Уральского ун-та, 2020. - 132 с. - ISBN 978-5-7996-3034- 8. - Текст : электронный. - URL: https://znanium.com/catalog/product/1936353;

3. Стоянович, С. Бессерверные приложения на JavaScript : практическое руководство / С. Стоянович, А. Симович ; пер. с анг. А. Н. Киселева. - Москва : ДМК Пресс, 2020. - 394 с. - ISBN 978-5-97060-782-4. - Текст : электронный. - URL: https://znanium.com/catalog/product/1094948;

4. Хорстманн, К. С. Современный JavaScript для нетерпеливых : практическое руководство / К. С. Хорстманн ; пер. с англ. А. А. Слинкина. - Москва : ДМК Пресс, 2021. - 288 с. - ISBN 978-5-97060-177-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/2109482;

5. Хортон, А. Разработка веб-приложений в ReactJS : практическое руководство / А. Хортон, Р. Вайс ; пер. с англ. Р. Н. Рагимова. - 2-е изд. - Москва : ДМК Пресс, 2023. - 255 с. - ISBN 978-5-89818-503-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/2107174.