# значокГосударственное бюджетное образовательное учреждение среднего профессионального образования Московской области

**«ГБПОУ МО Щелковский колледж»**

**структурное подразделение №3**

Специальность 09.02.07 «Информационные системы и программирование»

КУРСОВОЙ ПРОЕКТ

По дисциплине МДК 09.03.01 проектирование и разработка веб-приложений

Тема: Разработка проекта облачного хранилища данных для ФБУЗ ИМЦ Роспотребнадзора

Выполнил студент: Евменов Егор Сергеевич (подпись)

Группа 3119 номер личного дела

Проверил преподаватель:

(подпись)(ФИО)

Оценка защиты

Дата защиты

г. Щелково 2025 год

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc188261436)

[1.АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc188261437)

[1.1. Чем занимается предприятие 4](#_Toc188261438)

[1.2. Задачи курсового проекта 5](#_Toc188261439)

[1.3. Технические термины в курсовом проекте 6](#_Toc188261440)

[1.4. Требования к разрабатываемому проекту 7](#_Toc188261441)

[1.5. Сравнительный анализ существующих решений 8](#_Toc188261442)

[1.6. Преимущества разработки собственного веб-приложения 10](#_Toc188261443)

[1.7. Используемые технологии для разработки проекта 11](#_Toc188261444)

[1.8. Анализ предметной области 13](#_Toc188261445)

[1.9. Анализ возможных проблем и пути их решения 13](#_Toc188261446)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 15](#_Toc188261447)

[2.1. Техническое описание разрабатываемого проекта 15](#_Toc188261448)

[2.2. Этапы разработки проекта 16](#_Toc188261449)

[ЗАКЛЮЧЕНИЕ 31](#_Toc188261450)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 32](#_Toc188261451)

# ВВЕДЕНИЕ

В современном мире обеспечение информационной безопасности и эффективное управление данными играют ключевую роль для государственных учреждений, особенно для таких организаций, как Федеральное бюджетное учреждение здравоохранения "Информационно-методический центр" Роспотребнадзора (ФБУЗ ИМЦ Роспотребнадзора). В условиях растущих объемов данных и необходимости их централизованного хранения, обработки и анализа, внедрение облачного хранилища данных становится неотъемлемой частью стратегии цифровой трансформации.

Целью настоящего курсового проекта является разработка проекта облачного хранилища данных для ФБУЗ ИМЦ Роспотребнадзора. Данный модуль предназначен для автоматизации процессов хранения, обработки и анализа медицинских, эпидемиологических и исследовательских данных, обеспечения безопасности и доступности информации для сотрудников и ученых. Пользователи смогут легко находить нужные им данные и выполнять операции по их анализу и обработке.

Задачами данного курсового проекта являются:

* Исследование требований
* **Проектирование системы**
* **Выбор технологий для разработки проекта**
* **Разработка облачного хранилища данных**

# 1.АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1. Чем занимается предприятие

Федеральное бюджетное учреждение здравоохранения "Информационно-методический центр" Роспотребнадзора (ФБУЗ ИМЦ Роспотребнадзора) занимается предоставлением широкого спектра услуг в области общественного здоровья и гигиены, включая проведение исследований, мониторинг эпидемиологической обстановки, а также разработку методических рекомендаций для медицинских учреждений по всей территории России. Основная цель организации — обеспечение высокого уровня информационной безопасности, надежного хранения и эффективной обработки данных для удовлетворения потребностей сотрудников, ученых и других заинтересованных сторон.

Разработка проекта облачного хранилища данных для ФБУЗ ИМЦ Роспотребнадзора может иметь несколько целей:

1. Повышение уровня безопасности данных

* Сбор и анализ данных о хранении и обработке информации.
* Оценка качества выполненных работ на основании объективных показателей (безопасность данных, скорость обработки и др.).
* Улучшение взаимодействия с сотрудниками и учеными путем оперативного реагирования на замечания и предложения.

2. Совершенствование внутренних процессов

* Выявление слабых мест в работе сотрудников и подрядчиков.
* Анализ причин задержек, повреждений данных и других проблемных ситуаций.
* Введение стандартов качества и контроль их соблюдения.

3. Минимизация рисков и потерь

* Предупреждение повторения ошибок и инцидентов.
* Своевременное выявление недостатков в организации хранения и обработки данных.
* Уменьшение числа претензий со стороны сотрудников и ученых.

4. Автоматизация и оптимизация учета

* Быстрое получение аналитической информации для принятия управленческих решений.
* Сокращение временных и трудовых затрат на сбор и обработку данных.

Таким образом, разработка проекта облачного хранилища данных для ФБУЗ ИМЦ Роспотребнадзора направлена на улучшение качества предоставляемых услуг, повышение доверия сотрудников и укрепление своих позиций в области информационной безопасности и обработки данных.

Соответственно, целью данного курсового проекта является разработка проекта облачного хранилища данных для ФБУЗ ИМЦ Роспотребнадзора, где основными требованиями будут разработка системы хранения и акцент на безопасность данных.

## 1.2. Задачи курсового проекта

1. Исследование требований:
   1. Повышение уровня безопасности хранения данных.
   2. Совершенствование внутренних процессов обработки и анализа данных.
   3. Минимизация рисков и потерь данных.
   4. Увеличение эффективности работы сотрудников.
   5. Автоматизация и оптимизация учета данных.
   6. Развитие конкурентоспособности в области цифровой трансформации.
2. Проектирование системы:
   1. Разработка структуры базы данных для хранения медицинских, эпидемиологических и исследовательских данных.
   2. Проектирование интерфейса для внутреннего использования (для сотрудников и ученых).
3. Выбор технологий для разработки проекта:
   1. HTML
   2. CSS
   3. PHP
   4. MySQL
   5. JavaScript
4. Разработка облачного хранилища данных:
   1. Создание серверной части для обработки и анализа данных.
   2. Реализация клиентской части для взаимодействия с облачным хранилищем.

Выполнение этих задач позволит успешно реализовать проект облачного хранилища данных для ФБУЗ ИМЦ Роспотребнадзора и предоставить организации систему, соответствующую поставленным целям.

## 1.3. Технические термины в курсовом проекте

* Облачное хранилище данных: Централизованное хранилище данных, которое позволяет хранить, обрабатывать и анализировать большие объемы информации в защищенном режиме.
* Серверная часть (бэкенд): Часть программного обеспечения, отвечающая за обработку данных, выполнение логики приложения и взаимодействие с облачным хранилищем. В нашем проекте серверная часть будет отвечать за хранение, обработку и анализ медицинских, эпидемиологических и исследовательских данных.
* Клиентская часть (фронтенд): Часть программного обеспечения, с которой пользователь взаимодействует непосредственно. В нашем случае это веб-интерфейс, через который сотрудники и ученые могут просматривать, добавлять и обрабатывать данные.
* Интерфейс пользователя (UI): Элементы графического интерфейса, предназначенные для взаимодействия пользователя с системой. Включает в себя дизайн, элементы управления, анимацию и другие компоненты, обеспечивающие удобство использования.
* Шифрование: Процесс преобразования данных в зашифрованный вид, чтобы сделать их недоступными для несанкционированного доступа. Используется для защиты конфиденциальной информации.
* Аутентификация: Процесс проверки подлинности пользователя, например, через ввод логина и пароля. Обеспечивает доступ к защищённым ресурсам только авторизованным пользователям.
* База данных: Организованное хранилище данных, используемое для хранения информации о пользователях, отзывах, настройках и других данных, необходимых для работы системы.

## 1.4. Требования к разрабатываемому проекту

На основе проведенного «исследования требований» были выявлены следующие ключевые аспекты:

1. Пользовательские требования:
   * Удобный и интуитивно понятный интерфейс для работы с облачным хранилищем данных.
   * Безопасность персональных и конфиденциальных данных.
2. Функциональные требования:
   * Возможность хранения, обработки и анализа медицинских, эпидемиологических и исследовательских данных.
3. Нефункциональные требования:
   * Высокая производительность системы для обработки больших объемов данных.
   * Надежная защита данных с использованием современных методов шифрования и аутентификации.
   * Простота использования для сотрудников и ученых.

## 1.5. Сравнительный анализ существующих решений

Для проведения сравнительного анализа облачных хранилищ данных необходимо учитывать множество факторов, включая функциональность, безопасность, удобство использования, масштабируемость и стоимость. Рассмотрим несколько популярных решений, таких как Google Cloud Storage, Amazon S3, Microsoft Azure Blob Storage и IBM Cloud Object Storage.

Сравнительный анализ:

* Google Cloud Storage
  + Функциональность : Поддерживает различные типы данных, включая объекты, блочные и файловые системы. Предоставляет инструменты для управления метаданными, политиками доступа и интеграции с другими облачными сервисами.
  + Удобство использования : Интуитивно понятный интерфейс и мощные API для разработчиков.
  + Стоимость : Стоимость зависит от объема хранимых данных и операций.
  + Безопасность : Высокий уровень безопасности, включая шифрование данных на уровне объектов, многофакторную аутентификацию и регулярные аудиты.
  + Преимущества и недостатки :
    - Преимущества : Высокая масштабируемость, интеграция с другими сервисами Google Cloud, поддержка глобального размещения данных.
    - Недостатки : Сложность для новичков, необходимость регулярного обновления и обслуживания.
* Amazon S3
  + Функциональность : Поддерживает хранение объектов, файлов и блочных данных. Предоставляет инструменты для управления метаданными, политиками доступа и интеграции с другими облачными сервисами.
  + Удобство использования : Интуитивно понятный интерфейс и мощные API для разработчиков.
  + Стоимость : Стоимость зависит от объема хранимых данных и операций.
  + Безопасность : Высокий уровень безопасности, включая шифрование данных на уровне объектов, многофакторную аутентификацию и регулярные аудиты.
  + Преимущества и недостатки :
    - Преимущества : Огромный рынок сторонних инструментов и сервисов, поддержка глобального размещения данных, высокая надежность.
    - Недостатки : Сложность для новичков, необходимость регулярного обновления и обслуживания.
* Microsoft Azure Blob Storage
  + Функциональность : Поддерживает хранение объектов, файлов и блочных данных. Предоставляет инструменты для управления метаданными, политиками доступа и интеграции с другими облачными сервисами.
  + Удобство использования : Интуитивно понятный интерфейс и мощные API для разработчиков.
  + Стоимость : Стоимость зависит от объема хранимых данных и операций.
  + Безопасность : Высокий уровень безопасности, включая шифрование данных на уровне объектов, многофакторную аутентификацию и регулярные аудиты.
  + Преимущества и недостатки :
    - Преимущества : Лучшая интеграция с другими службами Microsoft, поддержка глобального размещения данных, высокая надежность.
    - Недостатки : Сложность для новичков, необходимость регулярного обновления и обслуживания.
* IBM Cloud Object Storage
  + Функциональность : Поддерживает хранение объектов, файлов и блочных данных. Предоставляет инструменты для управления метаданными, политиками доступа и интеграции с другими облачными сервисами.
  + Удобство использования : Интуитивно понятный интерфейс и мощные API для разработчиков.
  + Стоимость : Стоимость зависит от объема хранимых данных и операций.
  + Безопасность : Высокий уровень безопасности, включая шифрование данных на уровне объектов, многофакторную аутентификацию и регулярные аудиты.
  + Преимущества и недостатки :
    - Преимущества : Высокая надежность, поддержка глобального размещения данных, интеграция с другими IBM Cloud сервисами.
    - Недостатки : Сложность для новичков, необходимость регулярного обновления и обслуживания.

В результате анализа выше перечисленных систем управления сайтов, было принято решение разрабатывать сайт самостоятельного.

## 1.6. Преимущества разработки собственного облачного хранилища данных

* Контроль за данными: При разработке собственного облачного хранилища данных организация получает полный контроль над хранением и обработкой данных, что существенно повышает уровень безопасности и конфиденциальности информации.
* Подгонка под нужды организации: Облачное хранилище данных можно адаптировать под специфические потребности и бизнес-процессы организации, что обеспечит более эффективное взаимодействие как с сотрудниками, так и с учеными.
* Интеграция с внутренними системами: Разработанное облачное хранилище данных легко интегрируется с другими внутренними системами организации, такими как CRM, ERP и прочими, повышая общую производительность и удобство работы.
* Кастомизация интерфейса и функционала: Облачное хранилище данных позволяет настроить интерфейс и функционал точно под потребности организации, делая его максимально удобным и эффективным для использования.
* Экономия ресурсов в долгосрочной перспективе: В долгосрочной перспективе разработка и поддержка собственного облачного хранилища данных может оказаться экономически более выгодной, чем использование платных сервисов сторонних компаний.
* Уникальные функциональные возможности: Разработка собственного облачного хранилища данных даёт возможность внедрить уникальные функциональные возможности, специфические для организации, что может значительно повысить эффективность работы и привлечь больше сотрудников и ученых.
* Гибкость и скорость реакции на изменения: Имея собственное облачное хранилище данных, организация может быстро реагировать на изменения в своей деятельности и оперативно вносить необходимые изменения в структуру и содержание ресурса.

## 1.7. Используемые технологии для разработки проекта

Теперь, учитывая стоящие перед нами задачи, мы можем определить структуру облачного хранилища данных. Система будет разделена на модули для удобства разработки. Основные функции системы будут включать хранение, обработку и анализ медицинских, эпидемиологических и исследовательских данных. Безопасность данных будет обеспечиваться шифрованием информации и контролем доступа.

Для разработки облачного хранилища данных необходимо использовать определённые технологии, которые обеспечат функциональность, безопасность и удобство использования. В качестве технологий разработки будут использоваться: HTML, CSS, PHP, MySQL, JavaScript. Давайте рассмотрим, почему каждая из перечисленных технологий важна для создания облачного хранилища данных:

* HTML (HyperText Markup Language): используется для создания структуры веб-страниц, что позволяет организовать элементы интерфейса для взаимодействия с облачным хранилищем данных.
* CSS (Cascading Style Sheets): позволяет стилизовать веб-страницы, делая их более привлекательными и удобными для использования.
* PHP: является серверным языком программирования, который используется для обработки запросов от клиентов и взаимодействия с облачным хранилищем данных.
* MySQL: реляционная база данных, которая позволяет хранить информацию о пользователях, заказах и других данных, необходимых для работы сайта.
* JavaScript: используется для создания интерактивных элементов на веб-страницах, таких как динамическое обновление контента, асинхронная загрузка данных и валидация форм.

Использование всех этих технологий в совокупности позволит создать полнофункциональное облачное хранилище данных с удобным интерфейсом, безопасным хранением данных и широким спектром возможностей для взаимодействия с пользователями. Это решение обеспечит надежное хранение, обработку и анализ медицинских, эпидемиологических и исследовательских данных, что будет полезно для сотрудников и ученых ФБУЗ ИМЦ Роспотребнадзора.

## 1.8. Анализ предметной области

Предметная область «разработки облачного хранилища данных» включает в себя следующие основные компоненты информации:

1. ­Пользователи:
   * Идентификационные данные (логин, пароль);
   * Профили пользователей (имя, фамилия, должность, контакты);
   * Настройки уведомлений и приватности;
   * Участие в корпоративных мероприятиях и обсуждениях.
2. Контент:
   * Медицинские, эпидемиологические и исследовательские данные;
   * Отчеты и аналитика
   * Информационные страницы (о компании, о проекте).

Эти данные будут храниться в базе данных и облачном хранилище для обеспечения эффективного функционирования системы и обеспечения безопасности и доступности информации для сотрудников и ученых ФБУЗ ИМЦ Роспотребнадзора.

## 1.9. Анализ возможных проблем и пути их решения

При разработке и эксплуатации облачного хранилища данных могут возникнуть следующие проблемы:

1. Проблемы безопасности (утечка данных, перехват данных):

* Решение: использование защищенных протоколов передачи данных (например, HTTPS), шифрование данных, защита от SQL-инъекций и XSS-атак.
* Избежание: проведение аудита кода на наличие уязвимостей, регулярное обновление системы безопасности, использование библиотек с проверенной безопасностью.

1. Проблемы производительности (перегруз системы, замедление работы сайта):

* Решение: оптимизация кода, запросов к базе данных, кэширование данных, предусмотреть масштабирование приложения.
* Избежание: проведение тестирования производительности на ранних этапах разработки, использование инструментов мониторинга производительности.

1. Проблемы совместимости (некорректное отображение на мобильных устройствах):

* Решение: тестирование на различных браузерах и устройствах.
* Избежание: соблюдать стандарты веб-разработки, тестировать на всех целевых платформах.

1. Проблемы обновлений и поддержки:

* Решение: регулярно обновлять приложение, внедрять механизмы автоматического тестирования и деплоя, проводить обучение персонала.
* Избежание: планировать обновления заранее, поддерживать документацию и комментарии в коде, использовать системы контроля версий.

На этапе разработки можно избежать многих проблем, проводя тщательный анализ требований, правильное проектирование архитектуры сайта, тестирование на всех этапах разработки и использование лучших практик веб-разработки. Также важно учитывать факторы безопасности и производительности уже на стадии планирования и разработки.

# 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1. Техническое описание разрабатываемого проекта

На основе собранной информации и проведённого анализа мы можем составить техническое описание к разрабатываемому облачному хранилищу данных для ФБУЗ ИМЦ Роспотребнадзора:

1. Архитектура

Приложение будет разработано на основе клиент-серверной архитектуры. Основные компоненты включают:

* Клиентская часть: Разработана с использованием HTML, CSS и JavaScript, что обеспечивает поддержку работы на различных устройствах и браузерах.
* Серверная часть: Реализована на PHP для обработки запросов и взаимодействия с облачным хранилищем данных.

2. Безопасность

Для обеспечения безопасности данных будут приняты следующие меры:

* Шифрование данных: передача данных между клиентом и сервером будет осуществляться по защищенному каналу с использованием протокола HTTPS.
* Хранение паролей: пароли пользователей будут кэшированы в базу данных.
* Защита от атак: будут применены меры защиты от SQL-инъекций и XSS-атак путем использования подготовленных выражений и фильтрации входных данных.

3. Производительность

Для достижения оптимальной производительности приложение будет оптимизировано следующим образом:

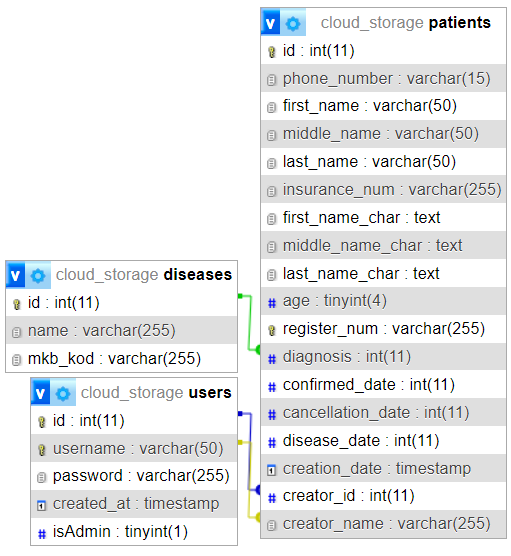
* Оптимизация кода: Код приложения будет минимизирован и сжат для быстрой загрузки и работы на клиентской стороне. Статические ресурсы будут загружаться через CDN.
* Оптимизация запросов к базе данных: Запросы к базе данных будут оптимизированы, а наиболее востребованные данные будут кэшироваться для ускорения доступа.
* Тестирование производительности: На начальных этапах разработки будут проведены тесты производительности для выявления и устранения потенциальных проблем.

4. Совместимость

* Приложение будет протестировано на различных браузерах и устройствах для обеспечения полной совместимости. Для этого будет использован подход прогрессивного улучшения, позволяющий гарантировать базовую функциональность на всех целевых платформах.

## 2.2. Этапы разработки проекта

Первым этапом в разработке проекта является база данных (рисунок1). Таблица diseases – это справочник, отвечающий за корректное наименование диагнозов. Таблица users хранит в себе зарегистрированных пользователей. Таблица patients хранит данные о пациентах и случаях заболевания.



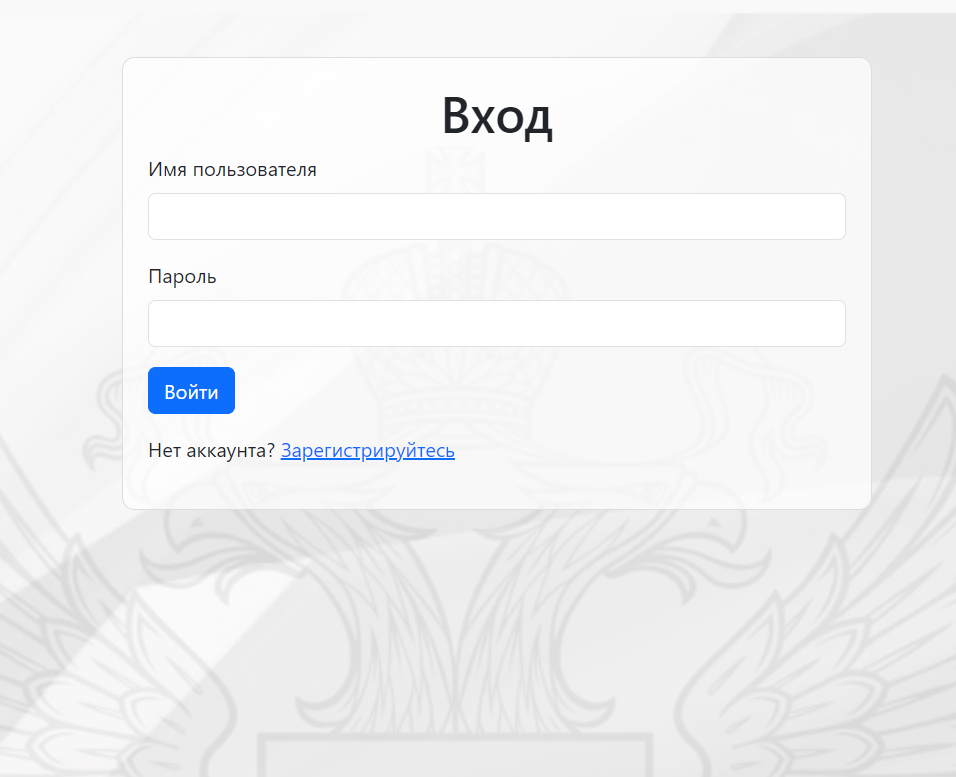
**Рисунок 1. База данных**

Вторым этапов в разработки проекта является программный код (рисунок 2). Данный код отвечает за регистрацию новых пользователей и занесения их в базу данных.

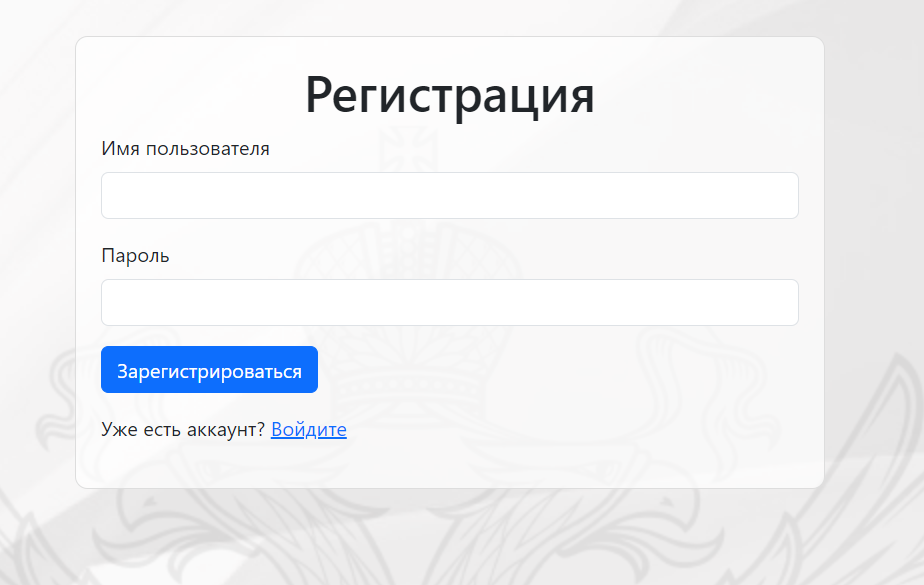
**Рисунок 2. Программный код в файле register\_process.php**

Теперь перейдем к описанию разработанного проекта. Начнем с раздела авторизации пользователя (рисунок 3). В данном разделе мы видим:

* кнопку «Войти», нажав на которую после валидации данных произойдёт переход на главную страницу
* гипер ссылку «Зарегистрируйтесь», нажав на которую произодёт переход на страницу регистрации (рисунок 4)
* Поля «Имя пользователя» и «Пароль», данные из которых используются для валидации при нажатии на кнопку «Войти»



**Рисунок 3. Страница авторизации**

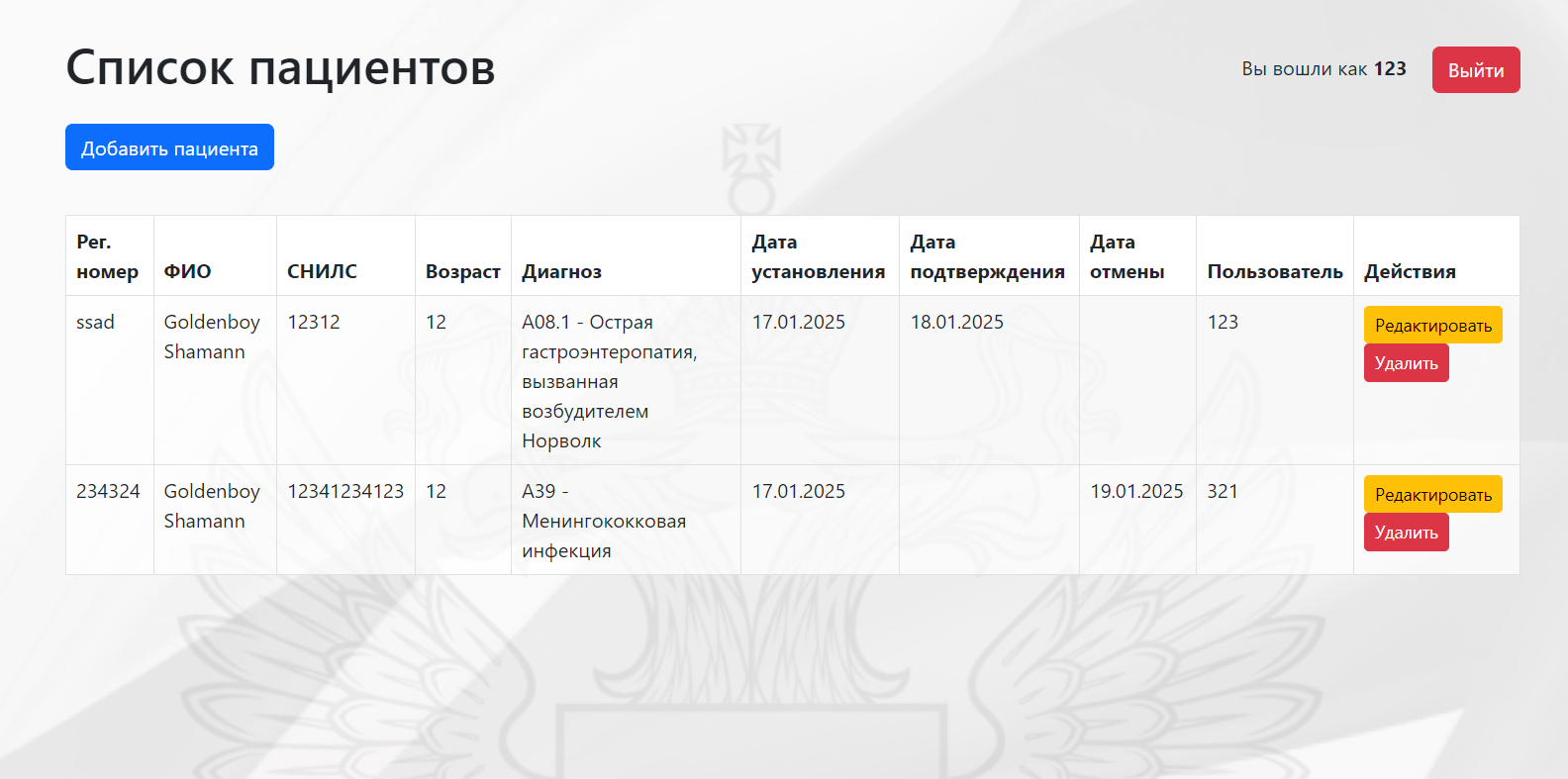
****

**Рисунок 4. Страница регистрации**

Давайте рассмотрим поподробнее кнопку войти. После валидации данных нас перенесёт на главную страницу сайта (рисунок 5), на этой странице имеется:

* кнопка «Выйти», нажав на которую пользователь перейдет на страницу авторизации (рисунок 3)
* кнопка «Добавить пациента», нажав на которую пользователя перенесет на страницу создания нового случая заболевания (рисунок 6)
* кнопка «Редактировать», нажав на которую пользователя перенесет на страницу редактирования случая заболевания (рисунок 7)
* кнопка «Удалить», нажав на которую запись удалится

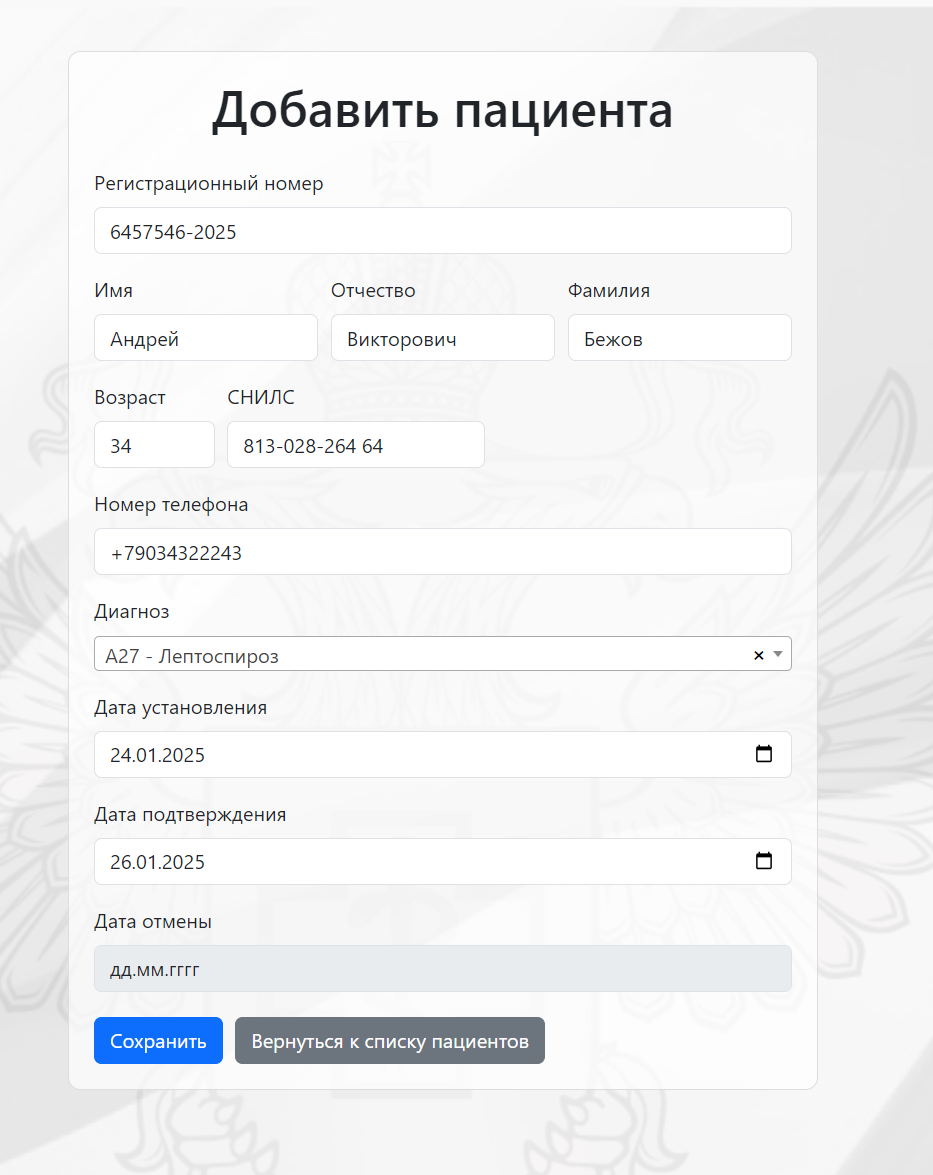
Стоит заметить, что видимость всех пациентов и столбец «Действия» в таблице пациентов доступен только для администратора. Обычный пользователь видит только свои карточки для соблюдения конфиденциальности.



**Рисунок 5. Главная страница**

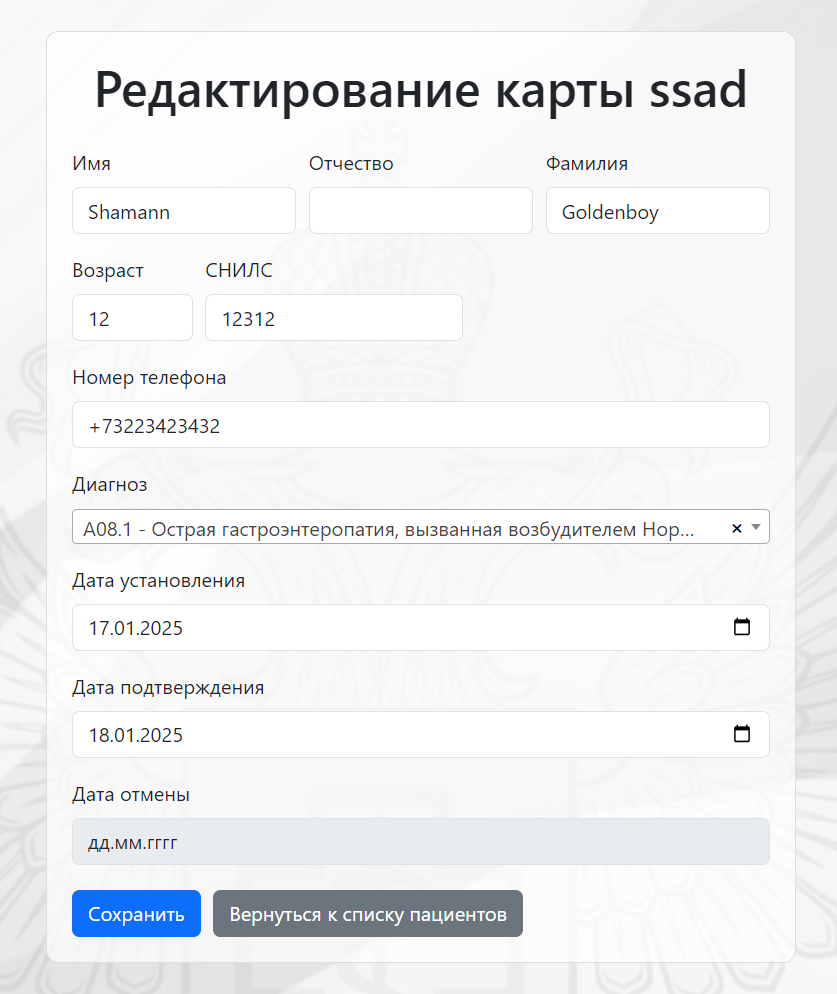
Перейдя на страницу создания пациента (кнопка «Добавить пациента») отображается форма создания. При заполненив форму можно нажать на кнопку «Сохранить». После сохранения происходит переход на главную страницу (рисунок 5), карточка сохраняется в таблицу patients и отображается в таблице на главной странице.

При заполнении необходимо учитывать, чтопри указании «Дата подтверждения» поле «Дата отмены» становится недоступным, и наоборот. Ни одна из перечисленных датне может быть раньше «Дата установления».

****

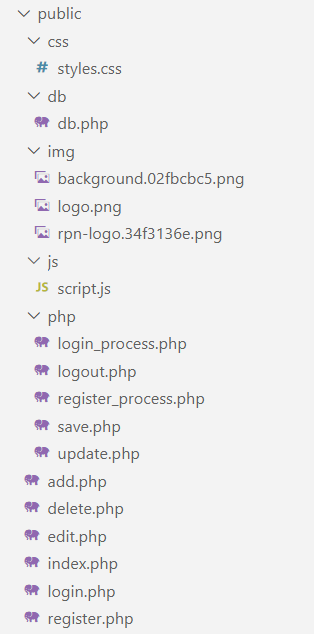
**Рисунок 6. Страница создания случая заболевания**

После сохранения карточки открывается главная страница (рисунок 5). Если в карточке были указаны неверные данные, то их можно отредактировать (кроме регистрационного номера), нажав на кнопку «Редактировать». Произойдет переход на страницу редактирования карточки (рисунок 7).



**Рисунок 7. Страница редактирования карточки**

На рисунке 8 представлены все файлы проекта.

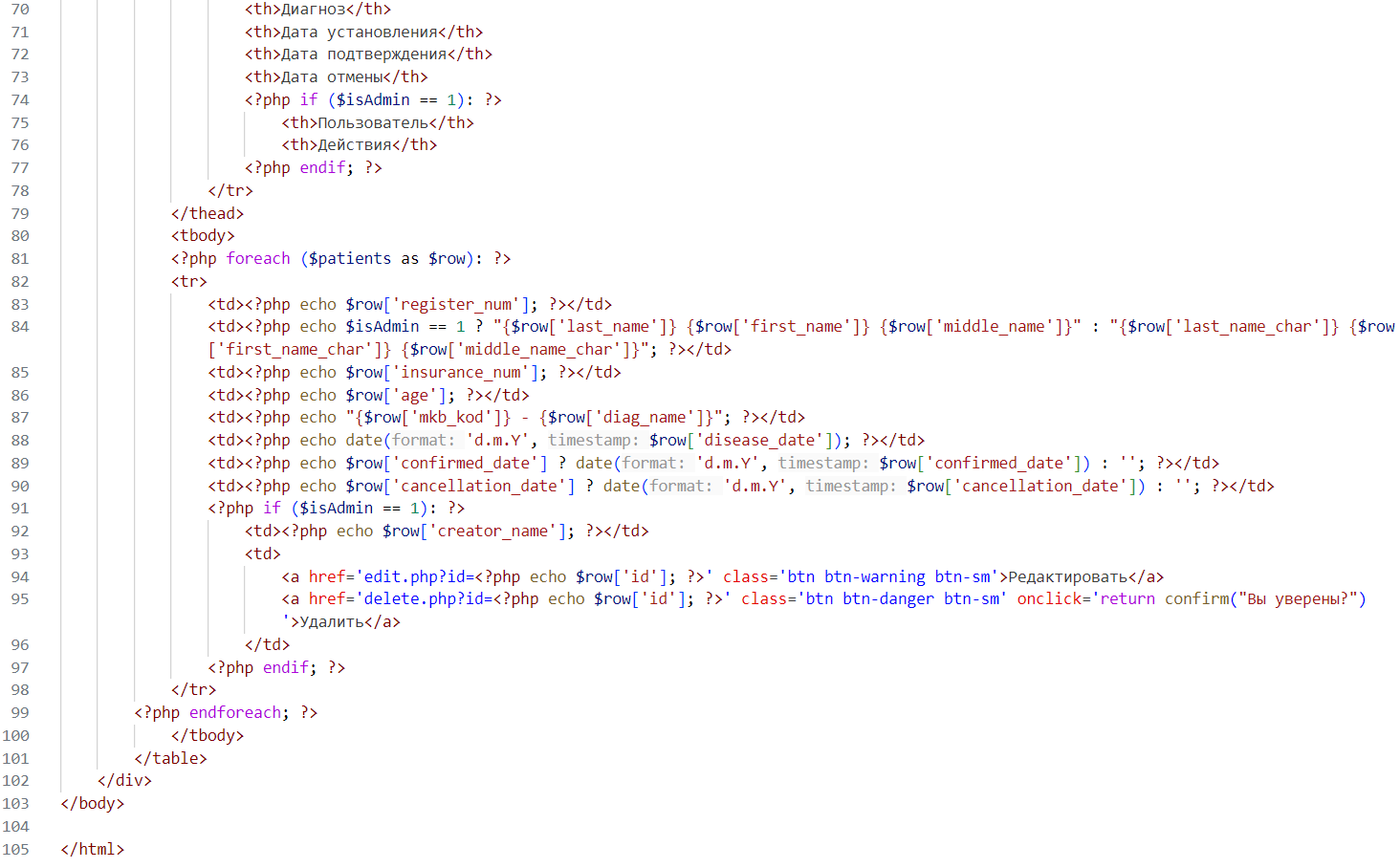


**Рисунок 8. Все файлы**

Теперь давайте рассмотрим один из файлов, который отвечает за отображение карточек на главной странице. Данный файл index.php находится в папке public(рисунок 9/1, 9/2, 9/3). Внешний вид данного кода можно увидеть на рисунке 5.

**Рисунок 9/1. Файл index.php в папке public**

**Рисунок 9/2. Файл index.php в папке public**

**Рисунок 9/3. Файл index.php в папке public**

# ЗАКЛЮЧЕНИЕ

В ходе работы над курсовым проектом были рассмотрены основные этапы создания облачного хранилища данных для ФБУЗ ИМЦ Роспотребнадзора.

Выполненные задачи:

* исследование требований,
* проектирование интерфейса,
* выбор технологий,
* разработка серверной и клиентской частей,
* интеграция с облачным хранилищем данных.

Курсовой проект по разработке облачного хранилища данных является важным шагом в изучении современных технологий разработки программного обеспечения и представляет собой отличную практическую работу, демонстрирующую умение применять теоретические знания на практике.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1.Автор:Алексеев А.В. «Разработка веб-приложений с использованием HTML, CSS, PHP, JavaScript и jQuery.» Год издания: 2019. Количество страниц: 280.

2. Авторы: Бен Фрейн, Дэвид Шоу "HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств", Год издания: 2020, Количество страниц: 400.

3.Авторы:Петрова О.И., Романов А.С., « Проектирование пользовательских интерфейсов веб-приложений с использованием HTML, CSS и JavaScript.», Год издания: 2020., Количество страниц: 312.

4. «Официальный сайт PHP» – <http://www.php.net/>

5. «HTMLBook» – https://htmlbook.ru/

6. «W3Schools» – <https://www.w3schools.com/>

7. Авторы: Дмитрий Котеров, Алексей Костарев,"PHP глазами хакера", Год издания: 2020., Количество страниц: 320.

8. Автор: Эрик Эллиотт, "Компиляция: современный JavaScript: шаблоны и идиомы для написания кода" (Compiling to JavaScript: Modern Patterns and Idoms for Writing Code), Год издания: 2020.. Количество страниц: 304.

9. Автор: Стив Сандерс, "Разработка веб-интерфейсов с использованием CSS: Практическое руководство", Год издания: 2023., Количество страниц: 352.

10. Автор: Эрик Эллиотт, "Компиляция: Современный JavaScript: Шаблоны и идиомы для написания кода" (Compiling to JavaScript: Modern Patterns and Idioms for Writing Code), Год издания: 2020., Количество страниц: 304.