Содержание

[Введение 7](#_Toc168317396)

[1 Аналитический обзор предметной области 9](#_Toc168317397)

[1.1 Обзор задачи 9](#_Toc168317398)

[1.2 Обзор методов для решения задач 10](#_Toc168317399)

[1.3 Постановка задачи 13](#_Toc168317400)

[2 Описание веб-приложения 15](#_Toc168317401)

[2.1 Обоснование выбора технологий 15](#_Toc168317402)

[2.2 Архитектура приложения и работы с данными 17](#_Toc168317403)

[2.3 Даталогическая модель 20](#_Toc168317404)

[2.4 Безопасность приложения 23](#_Toc168317405)

[3 Программная реализация модели 25](#_Toc168317406)

[3.1 Описание программной реализации 25](#_Toc168317407)

[Приложение А Листинг программы 33](#_Toc168317408)

Введение

Современные технологии значительно изменили методы взаимодействия и обучения. Одним из таких инновационных инструментов являются веб-приложения, направленные на дистанционную работу с обучающимися. В условиях стремительного развития цифровых технологий и увеличения потребности в удаленных формах взаимодействия, такие приложения становятся неотъемлемой частью образовательного процесса.

Данное веб-приложение предоставляет возможности для создания интерактивных досок, что позволяет не только улучшить организацию учебного процесса, но и повысить вовлеченность учащихся. Интерактивные доски служат эффективным инструментом для обмена идеями, совместной работы и наглядного представления информации. Будут рассмотрены основные функциональные возможности веб-приложения, его преимущества и значимость в контексте современного образования.

Интерактивные доски – это цифровые платформы, которые позволяют пользователям создавать, управлять и делиться информацией в визуально организованном формате. Подобные доски можно использовать для сбора идей, планирования проектов, обсуждения тем и совместной работы в реальном времени. Интерактивные доски предоставляют широкий спектр возможностей для эффективного взаимодействия между участниками образовательного процесса. Эти доски позволяют наглядно представлять идеи и концепции, что облегчает понимание и запоминание материала. Участники могут вносить свои идеи и правки, обеспечивая динамичное и продуктивное взаимодействие. Возможность структурировать информацию и задачи помогает эффективно планировать и отслеживать прогресс работы.

Использование веб-приложений для создания интерактивных досок значительно улучшает процесс обучения. Оно предоставляет обучающимся гибкость, позволяя работать в удобное для них время и в комфортной обстановке, что способствует более эффективному усвоению материала. Кроме того, веб-приложения позволяют подключаться к образовательному процессу из любой точки мира, что открывает доступ к качественному образованию для большего числа людей. Важно отметить, что использование современных технологий в обучении помогает развивать важные навыки работы с цифровыми инструментами.

Цели данной работы заключаются в детальном рассмотрении функциональных возможностей веб-приложения и его преимуществ для работы с обучающимися, а также в оценке его влияния на эффективность и качество образовательного процесса.

1. **Аналитический обзор предметной области**
   1. **Обзор задачи**

Современные образовательные учреждения сталкиваются с рядом вызовов, связанных с необходимостью адаптации к быстро меняющимся условиям и требованиям цифрового века. Существующие методы организации учебного процесса часто не соответствуют ожиданиям учащихся и преподавателей в плане гибкости, интерактивности и удобства использования. Возникла потребность в создании инструмента, который бы обеспечил более эффективное и продуктивное взаимодействие между всеми участниками образовательного процесса. Основные причины создания данного веб-приложения включают повышение эффективности учебного процесса, увеличение вовлеченности учащихся и поддержку дистанционного обучения.

Проект ориентирован на широкий круг пользователей, каждый из которых имеет свои специфические потребности и требования. Преподаватели и образовательные учреждения получают инструмент для улучшения организации учебного процесса, упрощения планирования занятий и повышения качества преподавания. Учащиеся и студенты могут воспользоваться удобным доступом к учебным материалам, улучшить взаимодействие с преподавателями и одногруппниками, а также более эффективно осваивать учебный материал.

Веб-приложение предоставляет пользователям широкий набор функций, направленных на улучшение организации и взаимодействия в образовательном процессе. Преподаватели могут создавать интерактивные доски для визуализации идей, планирования проектов и обсуждения тем. Эти доски могут содержать текстовые заметки, прикрепленные файлы, ссылки и другие элементы, помогающие наглядно представить информацию. Студенты могут комментировать доски, предоставленные преподавателями, и отчитываться по задачам, прикрепляя файлы с выполненными работами.

Возможность организовывать задачи и следить за их выполнением помогает поддерживать высокий уровень организации и продуктивности. Преподаватели могут создавать списки задач, устанавливать сроки и отслеживать прогресс их выполнения. Важной функцией веб-приложения является встроенный чат, который обеспечивает непрерывное взаимодействие между студентами и преподавателями. Чат позволяет оперативно решать вопросы, обсуждать учебные материалы и обмениваться мнениями, что способствует более тесному и продуктивному взаимодействию в процессе обучения.

Гибкость использования веб-приложения позволяет учащимся и преподавателям адаптировать учебный процесс под индивидуальные потребности и расписание каждого пользователя. Веб-приложение доступно из любой точки мира, что открывает доступ к образовательному процессу для большего числа людей. Учащиеся и преподаватели могут подключаться к платформе в любое удобное для них время, что позволяет адаптировать учебный процесс под индивидуальные потребности и расписание каждого пользователя. Такой подход способствует более эффективному усвоению материала и повышению качества образовательного процесса.

* 1. **Обзор методов для решения задач**

Для выполнения поставленной задачи и создания эффективного веб-приложения для дистанционной работы с обучающимися предусмотрен ряд ключевых этапов и методов.

Первым этапом является разработка функционального прототипа приложения. На этом этапе определяются основные функции, которые будут реализованы, такие как создание и управление интерактивными досками, возможность комментирования и прикрепления файлов студентами, а также интеграция чата для общения между преподавателями и студентами. Функциональный прототип позволяет получить первоначальные отзывы пользователей и внести необходимые коррективы до начала полноценной разработки.

Техническая реализация приложения осуществляется с использованием современных веб-технологий. Для разработки клиентской части применяются HTML5, JavaScript и CSS-фреймворк Bootstrap. HTML5 обеспечивает семантическую структуру веб-страниц и поддержку мультимедийных элементов, что способствует созданию богатого пользовательского интерфейса. JavaScript используется для реализации интерактивных функций и обеспечения динамического обновления содержимого страниц без перезагрузки. CSS-фреймворк Bootstrap помогает в быстрой и гибкой верстке адаптивного дизайна, который корректно отображается на различных устройствах.

Для связи с базой данных и обработки серверной логики используется PHP. Этот язык программирования известен своей эффективностью и широкими возможностями для интеграции с различными СУБД. В качестве системы управления базами данных выбрана PostgreSQL, которая предоставляет высокую производительность, надежность и поддерживает сложные запросы и транзакции, необходимые для работы веб-приложения.

Среда разработки Visual Studio Code выбрана благодаря своим мощным инструментам для написания, отладки и тестирования кода. Она поддерживает широкий спектр расширений и позволяет разработчикам настраивать рабочее окружение под конкретные задачи проекта. Использование этой среды способствует повышению продуктивности команды разработчиков и ускорению процесса разработки.

Для создания локального сервера применен серверный пакет Open Server. Это решение обеспечивает удобную настройку и управление сервером на локальной машине разработчика, что позволяет тестировать и отлаживать приложение в условиях, максимально приближенных к реальным. Такой подход сокращает время на разворачивание и настройку серверной инфраструктуры на начальных этапах разработки.

Интеграция системы управления пользователями является одним из ключевых аспектов проекта. Преподаватели и студенты могут легко регистрироваться и авторизоваться в системе, используя безопасные методы аутентификации и авторизации. Это обеспечивает защиту данных пользователей и позволяет управлять доступом к различным функциям и ресурсам приложения.

Функциональность интерактивных досок включает возможность добавления текстовых заметок, прикрепленных файлов, изображений и ссылок. Преподаватели могут создавать доски для визуализации учебного материала и планирования занятий. Студенты, в свою очередь, могут комментировать доски, делиться своими мыслями и прикреплять файлы с выполненными заданиями. Такая структура взаимодействия способствует более глубокому усвоению материала и поддерживает активное участие всех участников учебного процесса.

Встроенный чат играет важную роль в обеспечении непрерывного взаимодействия между студентами и преподавателями. Он позволяет оперативно решать возникающие вопросы, обсуждать учебные материалы и обмениваться мнениями. Чат интегрирован в веб-приложение таким образом, чтобы его использование было интуитивно понятным и удобным для всех пользователей. Это способствует созданию более продуктивной и кооперативной учебной среды.

Гибкость использования веб-приложения достигается благодаря его доступности на различных устройствах, включая компьютеры, планшеты и смартфоны. Адаптивный дизайн обеспечивает корректное отображение интерфейса на экранах разного размера и разрешения, что позволяет пользователям комфортно работать с приложением в любых условиях. Такая доступность расширяет возможности для использования веб-приложения в разных учебных сценариях и повышает его ценность для всех участников образовательного процесса.

* 1. **Постановка задачи**

Для достижения целей веб-приложения и удовлетворения потребностей его пользователей необходимо реализовать ряд ключевых функций. Эти функции обеспечат эффективное взаимодействие между преподавателями и студентами, а также повысят качество и организованность учебного процесса.

Одной из основных функций является создание и управление интерактивными досками. Это предполагает возможность для преподавателей создавать доски, которые будут служить центральным элементом для визуализации и организации учебных материалов. На досках можно размещать текстовые заметки, прикреплять файлы, изображения и ссылки. Каждая доска станет интерактивным пространством для обсуждения тем, планирования проектов и совместной работы. Создание текстовых заметок является ключевым аспектом этой функции, так как позволяет преподавателям добавлять важную информацию, инструкции и комментарии непосредственно на доске. Это облегчает восприятие и понимание материала студентами, а также способствует более структурированному представлению информации.

Важной функцией является возможность комментирования досок студентами. Это позволяет учащимся взаимодействовать с материалом, задавать вопросы, делиться своими мыслями и предлагать идеи. Комментирование способствует активному участию студентов в учебном процессе и создает условия для более глубокого усвоения материала через обсуждение и обмен мнениями.

Преподаватели должны иметь право создавать группы студентов. Эта функция необходима для организации учебного процесса по группам, что особенно важно в условиях большого количества студентов. Преподаватели смогут распределять студентов по группам, назначать групповые задания и управлять учебными мероприятиями внутри каждой группы. Это позволит более эффективно планировать и контролировать выполнение заданий, а также адаптировать учебные подходы к потребностям каждой группы.

Еще одной важной функцией является возможность общения в чате. Встроенный чат обеспечит непрерывное взаимодействие между студентами и преподавателями. Чат позволит оперативно решать вопросы, обсуждать учебные материалы, делиться мнениями и поддерживать активный диалог. Это особенно важно в условиях дистанционного обучения, где личное общение ограничено.

1. **Описание веб-приложения**
   1. **Обоснование выбора технологий**

Разработка веб-приложения для дистанционной работы с обучающимися требует использования надежных и проверенных технологий, которые обеспечат высокую производительность, безопасность и удобство использования. В этом разделе будут подробно рассмотрены средства, выбранные для разработки клиентской части, серверной части, базы данных, а также инструменты и среды разработки.

Для разработки клиентской части было принято решение использовать HTML5, JavaScript и Bootstrap. HTML5 является последней версией стандартного языка разметки для создания веб-страниц. Он обеспечивает поддержку мультимедийных элементов, таких как аудио и видео, а также улучшенную семантику и структуру документа. Это делает HTML5 идеальным выбором для создания современных, интерактивных и доступных веб-приложений. Благодаря своим возможностям HTML5 позволяет создавать структуру и содержание веб-страниц, что является фундаментом для всех остальных технологий, используемых в проекте.

JavaScript был выбран для создания динамичного и интерактивного пользовательского интерфейса. JavaScript является одним из наиболее широко используемых языков программирования для веб-разработки, и его использование позволяет создавать интерактивные элементы на страницах, такие как формы, анимации и события. JavaScript также поддерживается всеми современными браузерами, что обеспечивает широкую совместимость. В проекте активно используются технологии Ajax и DOM. Ajax (Asynchronous JavaScript and XML) позволяет обновлять части веб-страницы без перезагрузки всей страницы, что делает взаимодействие более быстрым и плавным. DOM (Document Object Model) предоставляет структуру документа, позволяя JavaScript изменять содержание, структуру и стиль страниц в реальном времени, что делает веб-приложение более интерактивным и динамичным.

Bootstrap, популярный фреймворк для разработки интерфейсов, был выбран для обеспечения адаптивного и эстетически приятного дизайна. Фреймворк – это набор готовых инструментов и компонентов, которые упрощают и ускоряют процесс разработки. Bootstrap предлагает набор готовых компонентов и стилей, таких как кнопки, формы, навигационные панели и другие элементы интерфейса. Эти компоненты легко настраиваются и интегрируются в проект, что значительно упрощает разработку визуальной части приложения. Важно отметить, что Bootstrap не вносит логических готовых решений, а предоставляет только визуальную составляющую, оставляя разработчикам полную свободу в реализации логики и функциональности приложения.

Для серверной части было решено использовать PHP. PHP является одним из самых популярных серверных языков программирования, который широко используется для создания динамических веб-страниц. Он обладает высокой производительностью, легкостью в изучении и использовании, а также большой библиотекой готовых решений и сообществом разработчиков. В проекте используется библиотека шифрования Sodium, которая обеспечивает высокий уровень безопасности для обработки и хранения данных. PHP также поддерживает множество других библиотек и фреймворков, таких как Laravel и Symfony, которые могут значительно упростить разработку сложных приложений, однако в данном проекте основной акцент делается на использование встроенных возможностей PHP и библиотеки Sodium для обеспечения безопасности.

Для работы с базой данных была выбрана PostgreSQL. PostgreSQL является мощной, открытой и объектно-реляционной системой управления базами данных (СУБД). Она известна своей надежностью, производительностью и соответствием строгим стандартам SQL. Это соответствие стандартам SQL помогает обеспечить совместимость и правильность обработки данных, что особенно важно для учебного приложения, где корректность и консистентность данных играют ключевую роль. PostgreSQL поддерживает сложные запросы, транзакции, расширяемость и множество встроенных функций, что делает её отличным выбором для работы с большими объемами данных и сложными структурами. Её возможности по масштабированию и производительности делают её подходящей для интенсивных нагрузок, что важно для поддержания стабильной работы приложения при большом количестве пользователей.

В качестве среды разработки был выбран Visual Studio Code. Visual Studio Code – это бесплатный, легкий и мощный редактор кода, который поддерживает множество языков программирования и инструментов разработки. Он предоставляет расширенные функции, такие как отладка, поддержка Git, интеграция с различными системами контроля версий, а также многочисленные расширения, которые могут улучшить процесс разработки и повысить продуктивность. Visual Studio Code обеспечивает удобную и эффективную среду для написания кода, позволяя разработчикам сосредоточиться на создании функциональности и логики приложения.

Для создания локального сервера был использован серверный пакет Open Server. Open Server предоставляет полный набор инструментов для веб-разработки, включая сервер Apache, базы данных MySQL и PostgreSQL, поддержку PHP и других языков программирования. Этот пакет позволяет разработчикам быстро настроить локальную среду разработки, что упрощает процесс тестирования и отладки приложения перед его развертыванием на производственном сервере. Open Server обеспечивает гибкость и удобство, предоставляя все необходимые инструменты в одном пакете, что ускоряет процесс разработки и тестирования.

* 1. **Архитектура приложения и работы с данными**

Алгоритм работы веб-приложения для дистанционной работы с обучающимися включает несколько ключевых этапов, начиная с регистрации, аутентификации и авторизации пользователей.

При регистрации пользователь заполняет форму с такими данными, как логин, пароль, имя, фамилия и адрес электронной почты. Эти данные отправляются на сервер с помощью Ajax-запроса. Серверная часть на PHP обрабатывает запрос, проверяя уникальность логина и адреса электронной почты. Также осуществляется проверка сложности пароля, чтобы обеспечить безопасность аккаунта. Если все проверки пройдены успешно, все данные, кроме логина, шифруются (используя, например, библиотеку Sodium), и вносятся в базу данных. После успешной регистрации пользователь может войти в систему, введя свои учетные данные на странице входа.

Когда пользователь вводит свои учетные данные (логин и пароль) на странице входа, данные отправляются на сервер с помощью Ajax-запроса. Серверная часть обрабатывает запрос, проверяя учетные данные и создавая сессию пользователя при успешной аутентификации. После этого пользователь перенаправляется на главную страницу приложения.

Преподаватель выбирает опцию создания новой доски. Форма создания доски отображается на экране, где преподаватель вводит название доски и описание. Эти данные отправляются на сервер с помощью Ajax-запроса. Серверная часть обрабатывает запрос и вносит данные в базу данных. После подтверждения создания новая доска отображается в интерфейсе приложения.

Преподаватель открывает созданную доску и выбирает опцию добавления новой заметки или прикрепленного файла. Форма добавления заметки или файла отображается на экране, где преподаватель вводит текст заметки или загружает файл. Эти данные отправляются на сервер с помощью Ajax-запроса. Серверная часть обрабатывает запрос и вносит данные в базу данных. Новая заметка или файл отображается на доске после получения подтверждения добавления.

Студент открывает доску и выбирает опцию добавления комментария. Форма комментария отображается на экране, где студент вводит текст комментария. Эти данные отправляются на сервер с помощью Ajax-запроса. Серверная часть обрабатывает запрос и вносит данные в базу данных. После подтверждения добавления комментарий отображается на доске.

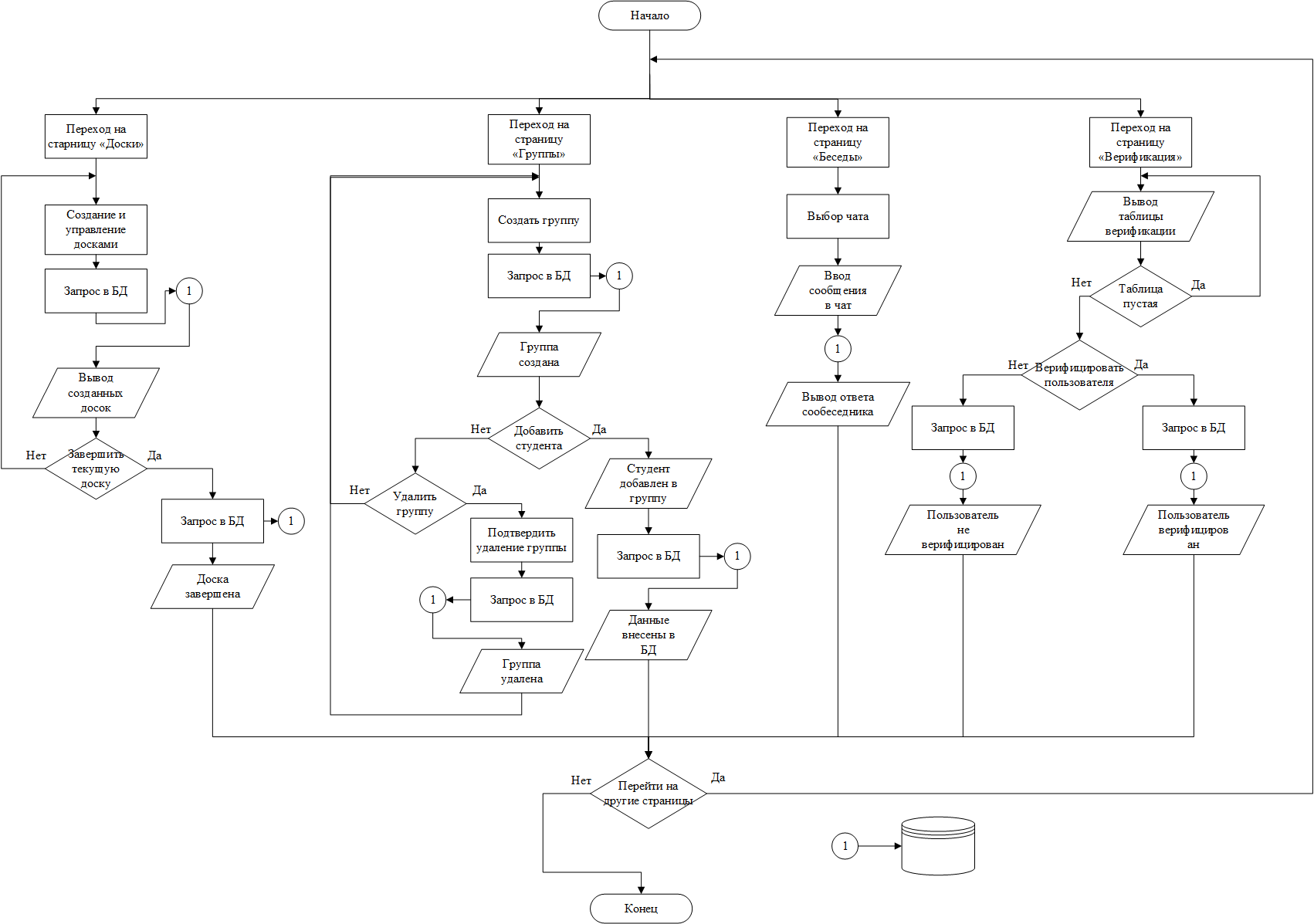
Преподаватель выбирает опцию создания новой группы студентов. Форма создания группы отображается на экране, где преподаватель вводит название группы и выбирает студентов из списка. Эти данные отправляются на сервер с помощью Ajax-запроса. Серверная часть обрабатывает запрос и вносит данные в базу данных. После подтверждения создания новая группа отображается в интерфейсе приложения.

Пользователь открывает окно чата и вводит сообщение. Сообщение отправляется на сервер с помощью Ajax-запроса. Серверная часть обрабатывает запрос и вносит данные в базу данных. Сообщение отображается в окне чата у всех пользователей после получения подтверждения отправки.

Связь веб-приложения с сервером осуществляется через серверную часть, написанную на PHP. При запуске приложения PHP устанавливает соединение с сервером, используя параметры подключения. Для безопасного соединения используются подготовленные запросы, что предотвращает SQL-инъекции. Для каждой операции (создание доски, добавление заметки, комментирование и другие действия) PHP формирует соответствующий запрос, который отправляется на сервер и выполняется.

После выполнения запроса сервер возвращает результаты, такие как данные пользователя при аутентификации или подтверждение добавления новой записи. PHP обрабатывает результаты, преобразуя их в формат, который может быть использован клиентской частью (обычно JSON). Эти данные отправляются обратно на клиентскую сторону с помощью Ajax, где клиентская часть JavaScript получает их и обновляет интерфейс пользователя соответствующим образом. После успешного добавления комментария новый комментарий отображается на доске без перезагрузки страницы.

На рисунке ниже представлен общий алгоритм работы веб-приложения, иллюстрирующий основные этапы взаимодействия между клиентской и серверной частями, а также работу с базой данных:



ПРИМЕР АЛГОРИТМА

* 1. **Даталогическая модель**

Эффективная работа веб-приложения для дистанционной работы с обучающимися во многом зависит от правильного взаимодействия с базой данных PostgreSQL. PostgreSQL, как надежная и производительная система управления базами данных, обеспечивает стабильное хранение и обработку данных, что является ключевым для функциональности и безопасности приложения.

При регистрации нового пользователя данные, введенные в форму, такие как логин, пароль, имя, фамилия и адрес электронной почты, отправляются на сервер. Серверная часть на PHP обрабатывает эти данные, проверяя уникальность логина и электронной почты, а также сложность пароля. Если все проверки пройдены успешно, сервер шифрует пароль, имя, фамилию и адрес электронной почты, используя библиотеку Sodium. Затем шифрованные данные вносятся в таблицу «Users». Внесение данных осуществляется посредством SQL-запроса, который вставляет новую запись в таблицу.

Во время аутентификации пользователя на сервер отправляются логин и пароль. Серверная часть проверяет эти данные, извлекая соответствующую запись из таблицы «Users». Для этого используется SQL-запрос, который ищет запись по логину и сравнивает шифрованный пароль. Если учетные данные верны, создается сессия пользователя и информация о нем возвращается на клиентскую сторону.

Когда преподаватель создает новую доску, введенные данные, такие как название и описание доски, отправляются на сервер. Серверная часть обрабатывает эти данные и вносит их в таблицу «tabs». SQL-запрос добавляет новую запись в таблицу, включая идентификатор преподавателя, который создал доску.

Добавление текстовых заметок и прикрепленных файлов также связано с взаимодействием с базой данных. Преподаватель вводит текст заметки или загружает файл, после чего данные отправляются на сервер. Серверная часть обрабатывает данные и вносит их в таблицы «assignments» или «descriptions», в зависимости от типа добавляемого контента. SQL-запрос вставляет новую запись в соответствующую таблицу, связывая ее с идентификатором доски.

При комментировании доски студентом данные комментария отправляются на сервер, где обрабатываются и вносятся в таблицу «Comments». Серверная часть использует SQL-запрос для добавления новой записи, связывая комментарий с соответствующим заданием и пользователем.

Создание групп студентов также требует взаимодействия с базой данных. Преподаватель вводит название группы и выбирает студентов из списка, после чего данные отправляются на сервер. Серверная часть обрабатывает данные и вносит их в таблицу «groups». SQL-запрос добавляет новую запись в таблицу, связывая группу с преподавателем и идентификатором доски.

Общение в чате также связано с базой данных. Пользователь вводит сообщение, которое отправляется на сервер. Серверная часть обрабатывает сообщение и вносит его в таблицу «messenges». SQL-запрос вставляет новую запись, связывая сообщение с соответствующим чатом и пользователем.

Связь веб-приложения с базой данных PostgreSQL осуществляется через серверную часть, написанную на PHP. При запуске приложения PHP устанавливает соединение с базой данных, используя параметры подключения, такие как хост, имя базы данных, пользователь и пароль. Для безопасного соединения используются подготовленные запросы, что предотвращает SQL-инъекции.

Для каждой операции, такой как создание доски, добавление заметки, комментирование и другие действия, PHP формирует соответствующий SQL-запрос, который отправляется в PostgreSQL и выполняется. После выполнения запроса PostgreSQL возвращает результаты, такие как данные пользователя при аутентификации или подтверждение добавления новой записи. PHP обрабатывает результаты, преобразуя их в формат, который может быть использован клиентской частью, обычно JSON. Эти данные отправляются обратно на клиентскую сторону с помощью Ajax, где клиентская часть JavaScript получает их и обновляет интерфейс пользователя.

На рисунке ниже представлена даталогическая модель базы данных, иллюстрирующая структуру таблиц и их связи:

![Даталогическая модель](вставить\_путь\_к\_рисунку\_здесь)

Эта модель позволяет понять, как данные организованы и связаны между собой, что помогает обеспечить надежное и эффективное взаимодействие между различными компонентами веб-приложения.

* 1. **Безопасность приложения**

Обеспечение безопасности данных пользователей является приоритетом при разработке веб-приложения для дистанционной работы с обучающимися. Для защиты данных используются современные методы шифрования и предотвращения атак, что помогает сохранить конфиденциальность и целостность информации.

При регистрации нового пользователя данные, введенные в форму, такие как логин, пароль, имя, фамилия и адрес электронной почты, отправляются на сервер. После прохождения проверок уникальности логина и адреса электронной почты, а также сложности пароля, данные шифруются перед внесением в базу данных. Для шифрования используется библиотека Sodium, которая применяет алгоритм симметричного шифрования ChaCha20.

Алгоритм ChaCha20 обеспечивает высокую скорость шифрования и надежную защиту данных. Этот алгоритм работает, преобразовывая входной текст в шифрованный текст с помощью ключа и случайного значения, используемого для обеспечения уникальности шифрования. ChaCha20 выполняет серию математических операций над блоками данных, которые обеспечивают их надежное шифрование. В результате данных шифрования информация становится недоступной для несанкционированного доступа, что делает алгоритм ChaCha20 особенно эффективным для защиты личных данных.

Во время аутентификации пользователя серверная часть проверяет введенные учетные данные, извлекая соответствующую запись из базы данных. Процесс расшифровки позволяет проверить подлинность данных без компрометации их безопасности.

Для предотвращения SQL-инъекций применяются подготовленные запросы. Подготовленные запросы обеспечивают безопасное выполнение SQL-запросов, исключая возможность вставки вредоносного кода. В подготовленных запросах параметры запроса передаются отдельно от самого SQL-кода, что предотвращает интерпретацию пользовательского ввода как части SQL-команды. Это достигается за счет использования параметризированных запросов, где значения параметров передаются безопасным образом, даже если пользовательский ввод содержит специальные символы.

Этот подход исключает возможность выполнения непреднамеренных команд, введенных злоумышленниками, защищая базу данных от атак SQL-инъекций. Все данные, вводимые пользователями, проходят обязательную валидацию и очистку, что также способствует повышению уровня безопасности приложения.

1. **Программная реализация модели**
   1. **Описание программной реализации**

В каталоге «Mainpage» также находится структура проекта, включающая файлы PHP и JavaScript. Файл index.php представляет разметку главной страницы и служит для перехода на страницы авторизации и регистрации. Вывод страницы index.php представлен на рисунке 3.

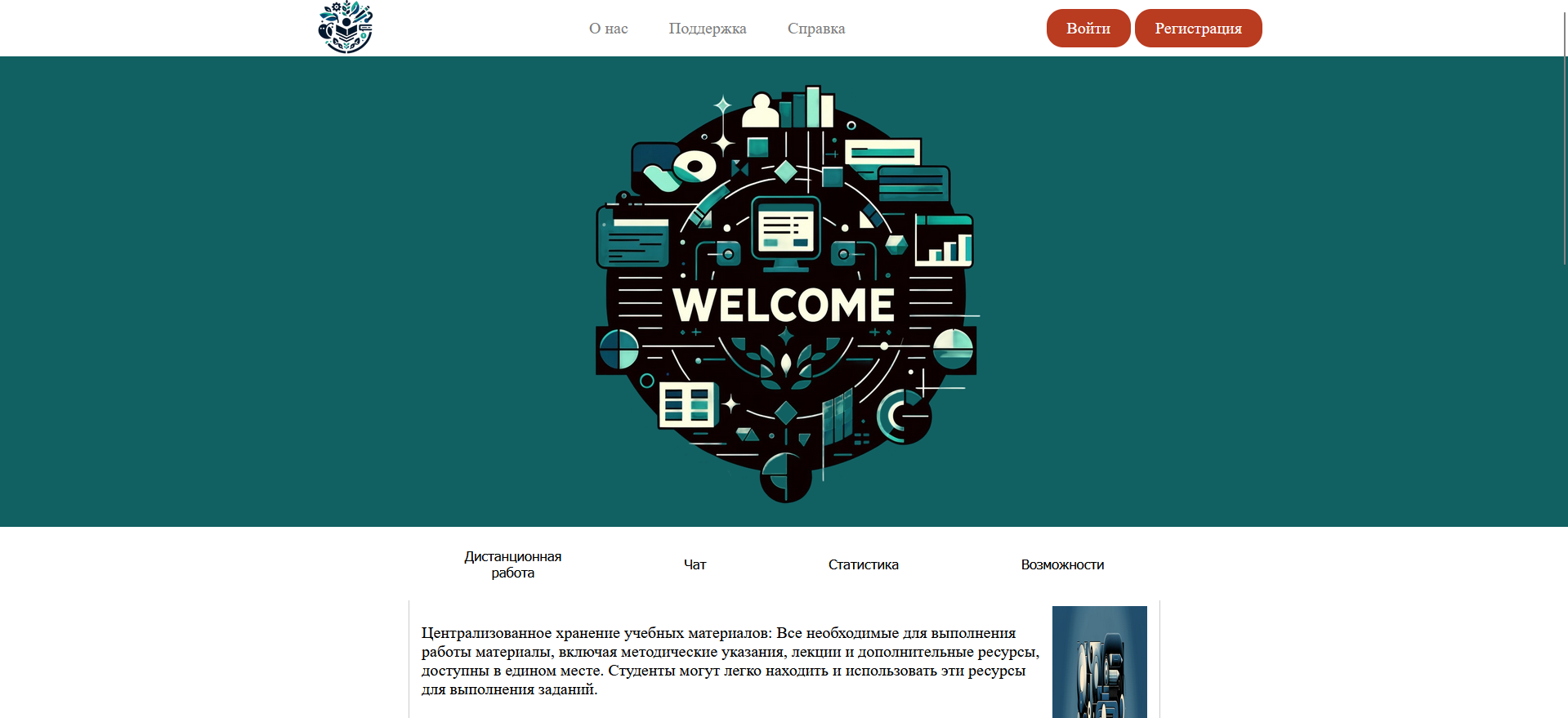


Рисунок 3. Страница index.php

Файл authorization.php представляет собой форму авторизации и передает данные в checking\_authorization.php, который проверяет данные пользователя, сохраняет информацию в сессии и перенаправляет его в зависимости от роли. Вывод страницы authorization.php представлен на рисунке 4.

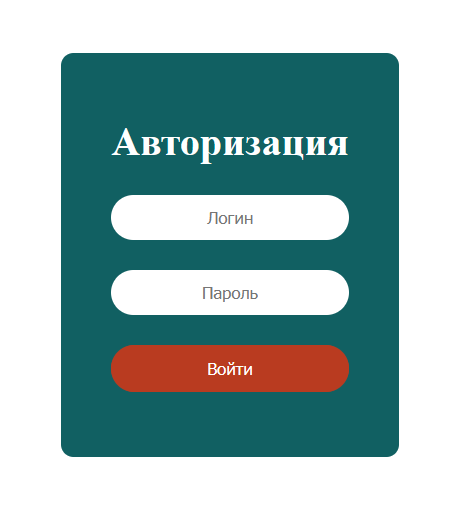


Рисунок 4. Страница authorization.php

Файл registration.php разметывает форму регистрации, данные из которой обрабатываются в checking\_form\_submit.php, проверяя заполненность полей, хешируя пароли и добавляя нового пользователя в базу данных. Вывод страницы registration.php представлен на рисунке 5.

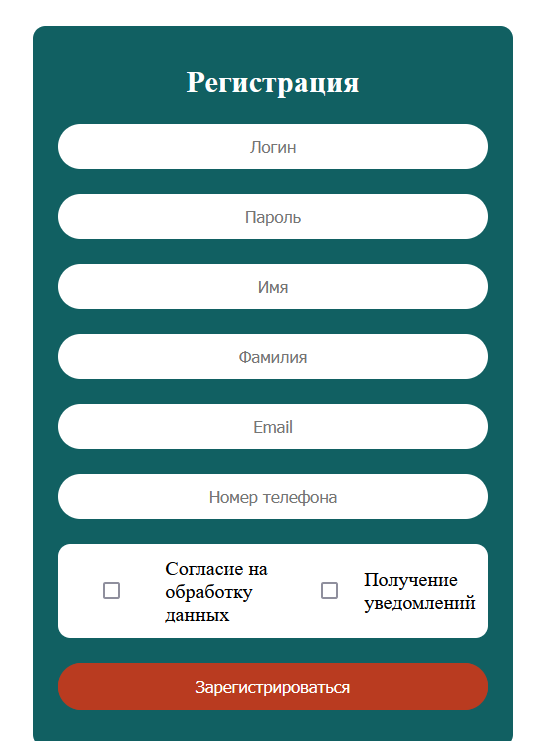


Рисунок 5. Страница registration.php

Файл roleVerification.php проверяет, является ли пользователь преподавателем, через данные сессии, а pdo\_connect.php подключается к базе данных PostgreSQL, создавая объект PDO с указанными параметрами. Файл user.php разметывает элементы профиля пользователя, выводит данные из базы и обеспечивает переход между страницами. Вывод страницы user.php представлен на рисунке 6.

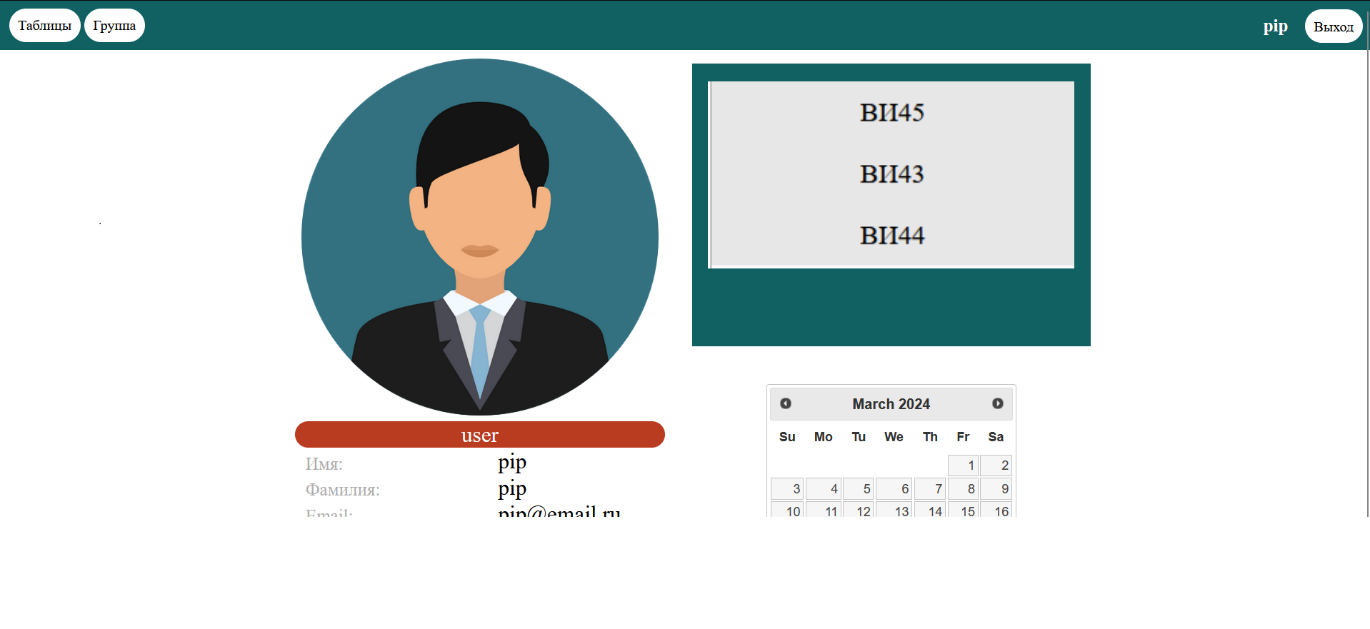


Рисунок 6. Страница user.php

Файл teacher.php выполняет аналогичные функции для профиля преподавателя, добавляя управление правами доступа. Вывод страницы teacher.php представлен на рисунке 7.

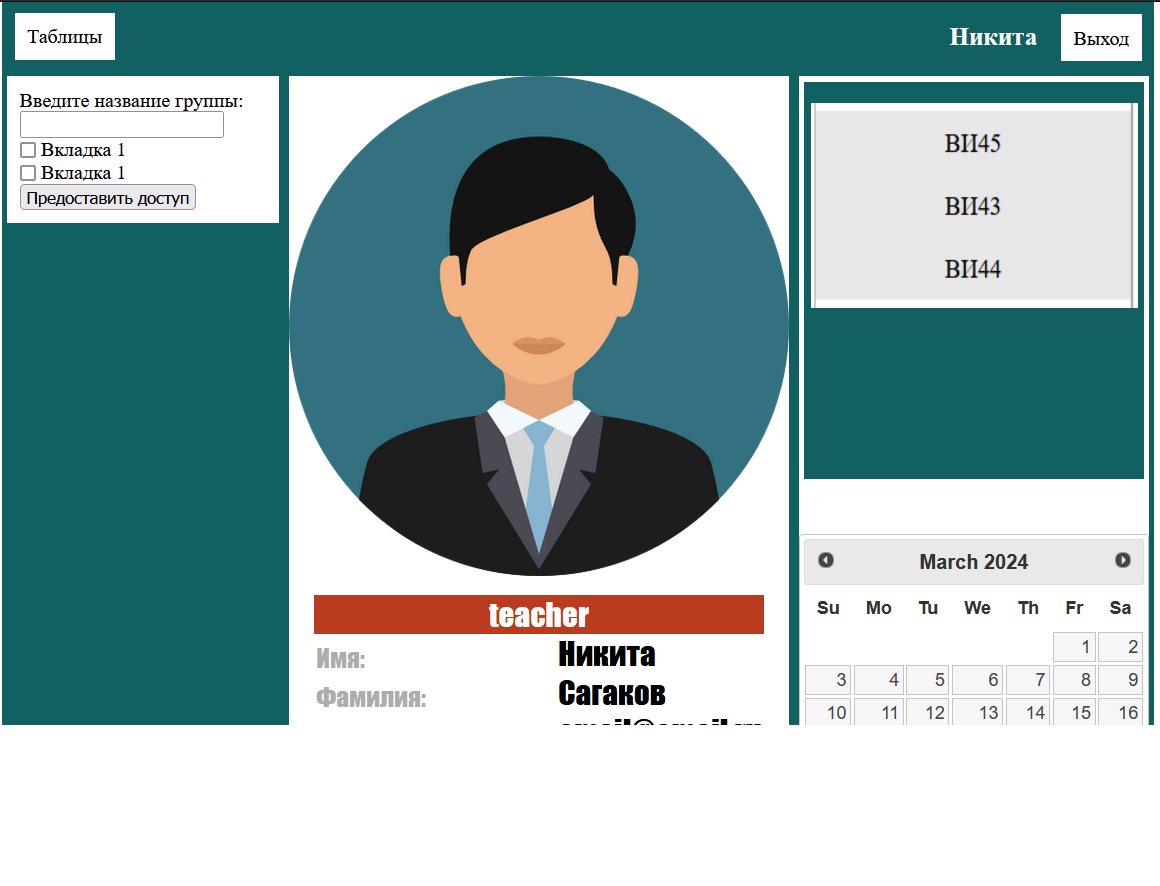


Рисунок 7. Страница teacher.php

Файл select\_checkbox.php получает список таблиц из базы данных для отображения на странице преподавателя, а workpage.php разметывает элементы страницы с таблицами заданий. Вывод страницы workpage.php представлен на рисунке 8.

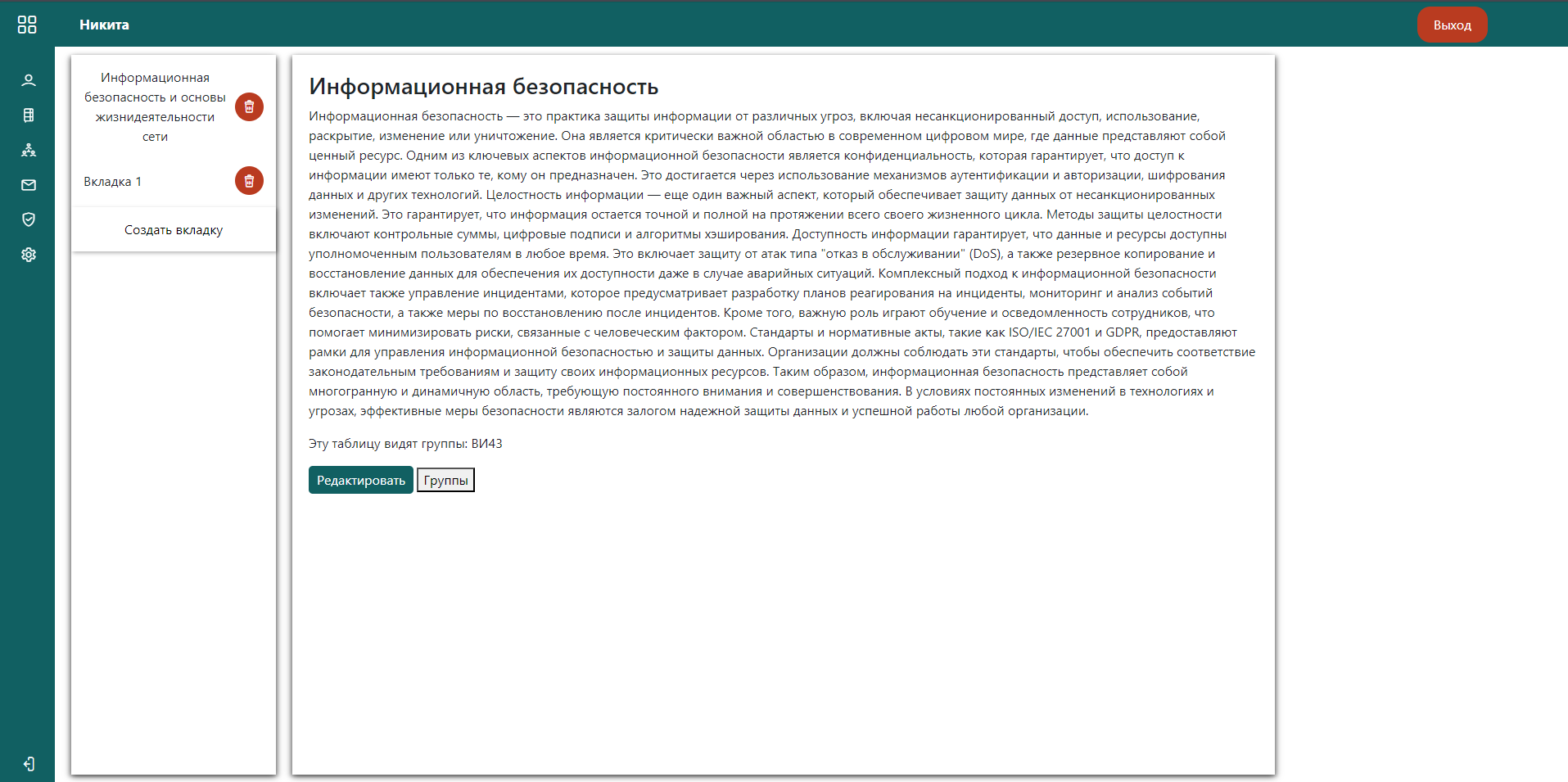


Рисунок 8. Страница workpage.php

Файл group.php разметывает страницу групп, включая формы для их создания и приглашения в группы. Вывод страницы group.php представлен на рисунке 9.

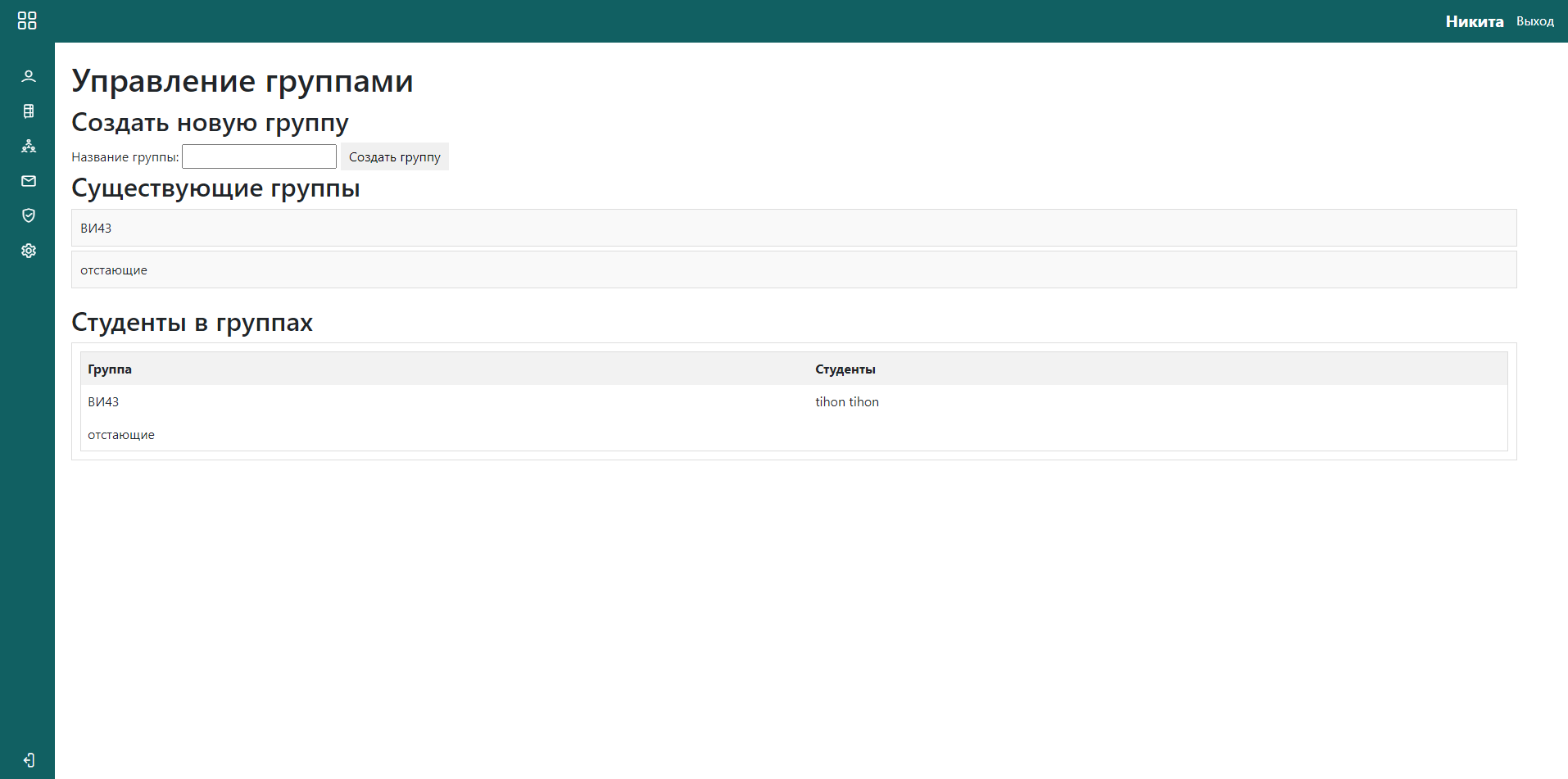


Рисунок 9. Страница group.php

Файл verification.php играет важную роль в системе безопасности веб-приложения. Он отвечает за проверку различных данных и действий, связанных с аутентификацией и авторизацией пользователей. Вывод страницы verification.php представлен на рисунке 10.

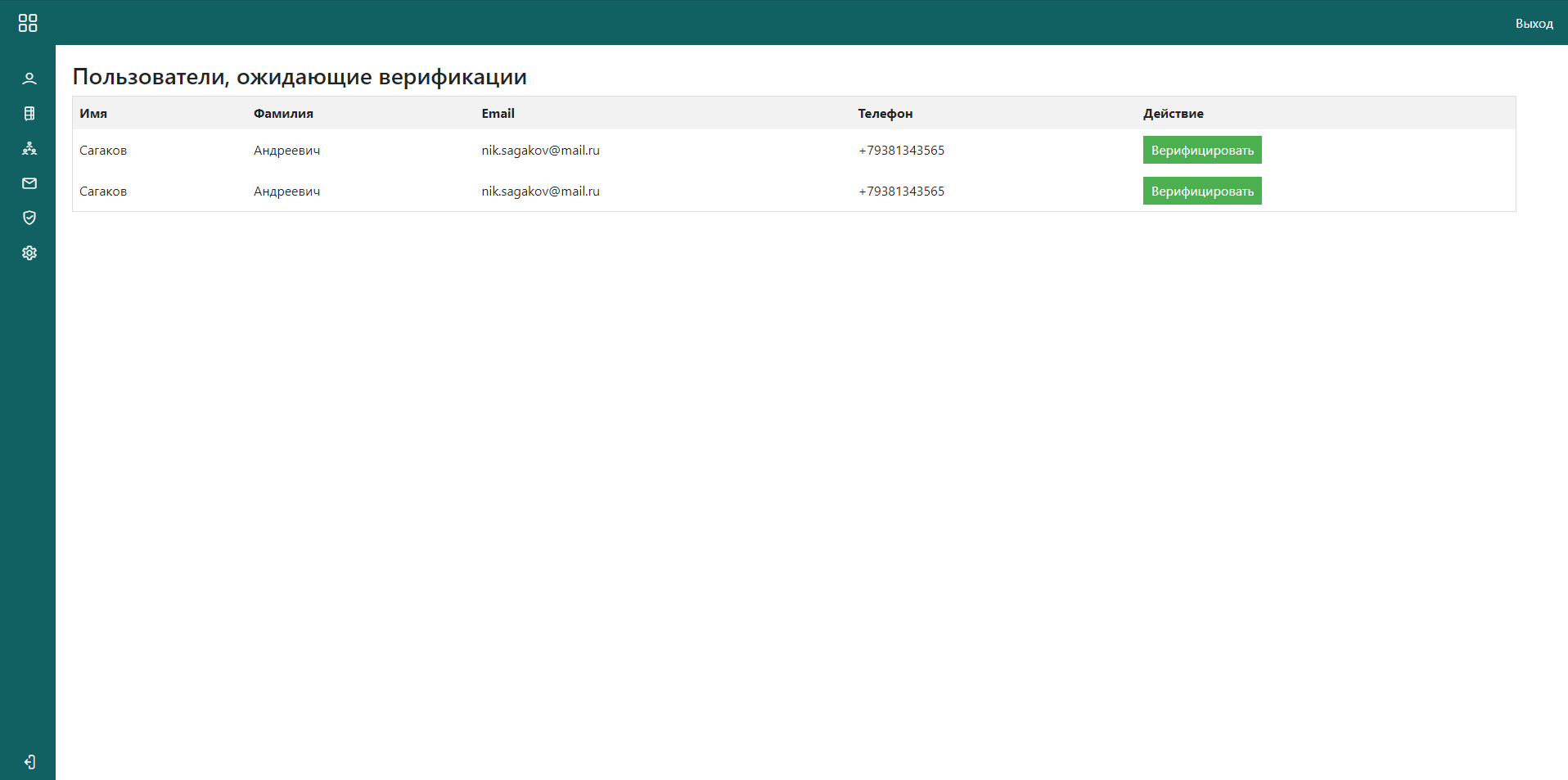


Рисунок 10. Страница verification.php

Скрипты getTabs.php и getRole.php возвращают список таблиц и проверяют роль пользователя, соответственно. Файлы addTab.php и addTask.php обрабатывают добавление новых таблиц и задач, создавая соответствующие SQL-запросы. create\_group.php и createAssignment.php создают новую группу и задачу, добавляя их в базу данных и связывая с пользователем. checkTabContent.php проверяет наличие заданий в таблице, deleteAssignment.php и deleteTab.php удаляют задачи и таблицы из базы данных, а editAssignment.php обновляет данные задачи. Скрипты fetch\_group.php и fetch\_users\_by\_group.php получают списки групп и пользователей через SQL-запросы.

JavaScript-скрипт invite\_user.js обеспечивает приглашение пользователей в группы через динамическую форму и AJAX-запросы, delete\_tab.js удаляет таблицы, а action\_teacher.js управляет доступом к таблицам и заданиям для учителей. Наконец, action\_on\_tab.js управляет таблицами и заданиями на странице, а action\_on\_group.js обрабатывает создание и управление группами чата через AJAX-запросы, улучшая пользовательский опыт за счет интерактивного взаимодействия.

Заключение

Перечень использованных информационных ресурсов

1. Официальная документация языка разметки HTML [Электронный ресурс], URL: <https://www.w3.org/2019/04/WHATWG-W3C-MOU.html> (дата обращения: xx.xx.xx)
2. Официальная документация каскадных таблиц стилей [Электронный ресурс], URL: <https://www.w3.org/Style/CSS/Overview.en.html> (дата обращения: xx.xx.xx)
3. Официальная документация языка программирования JS [Электронный ресурс], URL: <https://www.javascript.com> (дата обращения: xx.xx.xx)
4. Официальная документация языка программирования РНР [Электронный ресурс], URL: <https://www.php.net/manual/ru/index.php> (дата обращения: xx.xx.xx)
5. Официальная документация языка программирования PostgreSQL [Электронный ресурс], URL: <https://www.postgresql.org/docs/> (дата обращения: xx.xx.xx)
6. Официальная документация библиотеки Sodium для языка PHP [Электронный ресурс], URL: <https://www.php.net/manual/ru/book.sodium.php> (дата обращения: xx.xx.xx)

Приложение А Листинг программы