

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №43

КУРСОВОЙ ПРОЕКТ
ЗАЩИЩЕН С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

канд. техн. наук, доцент

должность, уч. степень, звание

подпись, дата

Т.И. Беляя

инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ ПРОЕКТУ

на тему: “Создание информационной системы виртуальных лабораторий”

по дисциплине: ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № _____ 4232

подпись, дата

Н.В. Юрагин

инициалы, фамилия

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	3
ВВЕДЕНИЕ	5
1. АНАЛИЗ ИСПОЛЬЗОВАНИЯ ВИРТУАЛЬНЫХ ЛАБОРАТОРИЙ	
Ошибка! Закладка не определена.	
1.1. Особенности виртуальных лабораторий в образовании	6
1.2. Роль автоматизации и геймификации в виртуальных лабораториях....	6
1.3. Анализ конкурентов и существующих решений	7
1.4. Проблемы и перспективы развития ИС	8
2. ХАРАКТЕРИСТИКА ОБЪЕКТА РАЗРАБОТКИ	9
2.1. Функциональные требования к ИС	9
2.2. Нефункциональные требования к ИС	10
2.3. Целевая аудитория ИС	12
2.4. Ограничения и технические требования к системе	13
2.5. Перспективы развития ИС	14
3. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ.....	16
3.1. Архитектура разрабатываемой системы	16
3.2. Проектирование базы данных	16
3.3. Проектирование API.....	18
3.4. Проектирование интерфейсов для доступа к функционалу ИС	22
3.5. Используемые технологии обеспечения информационной безопасности	25
4. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ.....	26
4.1. Реализация серверной части	26
4.2. Реализация клиентской части администратора и пользователя	28
4.3. Проведённое тестирование и обеспечение качества	33
4.4. Развёртывание приложения на сервере с использованием технологий контейнеризации	35
ЗАКЛЮЧЕНИЕ	36
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	37
ПРИЛОЖЕНИЕ А	38
ПРИЛОЖЕНИЕ Б.....	39
ПРИЛОЖЕНИЕ В (РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ)	53

ПРИЛОЖЕНИЕ Г (РУКОВОДСТВО РАЗРАБОТЧИКА)	63
---	----

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ИС ВиртЛаб – Информационная система виртуальных лабораторий.

API – Application Programming Interface (программный интерфейс приложения).

JWT – JSON Web Token (токен для аутентификации).

REST – Representational State Transfer (архитектурный стиль взаимодействия компонентов сети).

LMS – Learning Management System (система управления обучением, например, Moodle).

AI – Artificial Intelligence (искусственный интеллект).

VR – Virtual Reality (виртуальная реальность).

AR – Augmented Reality (дополненная реальность).

SCORM – Sharable Content Object Reference Model (стандарт для электронного обучения).

ECTS – European Credit Transfer and Accumulation System (система зачетных единиц).

PostgreSQL – реляционная система управления базами данных.

Node.js – серверная платформа для выполнения JavaScript.

Express.js – фреймворк для разработки веб-приложений на Node.js.

React – библиотека JavaScript для создания пользовательских интерфейсов.

TypeScript – язык программирования, расширяющий JavaScript статической типизацией.

Docker – платформа для контейнеризации приложений.

WebGL – API для рендеринга 3D-графики в браузере.

TLS – Transport Layer Security (протокол шифрования данных).

AES-256 – алгоритм симметричного шифрования с ключом 256 бит.

WAF – Web Application Firewall (брандмауэр для веб-приложений).

ФГОС – Федеральный государственный образовательный стандарт.

GDPR – General Data Protection Regulation (регламент защиты данных).

ЕС).

152-ФЗ – Федеральный закон "О персональных данных" (РФ).

WCAG – Web Content Accessibility Guidelines (стандарты доступности веб-контента).

User Service – сервис управления пользователями.

Admin Service – сервис административных функций.

RBAC – Role-Based Access Control (управление доступом на основе ролей).

Prometheus + Grafana – инструменты мониторинга и визуализации метрик.

ВВЕДЕНИЕ

Современные образовательные процессы требуют инновационных подходов к организации лабораторных работ, особенно в условиях ограниченного доступа к физическим лабораториям или необходимости соблюдения строгих мер безопасности. Для решения этих задач была разработана информационная система виртуальной лаборатории «ИС ВиртЛаб», которая позволяет студентам проводить химические эксперименты в цифровой среде, а преподавателям — контролировать и оценивать результаты автоматически.

Актуальность проекта обусловлена возрастающей потребностью в дистанционных и гибридных формах обучения, а также необходимостью снижения рисков, связанных с проведением опасных химических опытов в реальных условиях. Виртуальные лаборатории уже доказали свою эффективность в образовательных учреждениях, предоставляя студентам возможность изучать сложные процессы в безопасной и контролируемой среде. Данная система сочетает в себе точность моделирования химических реакций с элементами геймификации, такими как автоматическая оценка результатов, что повышает мотивацию студентов и упрощает работу преподавателей.

«ИС ВиртЛаб» призвана сделать процесс обучения более интерактивным и доступным. В отличие от традиционных лабораторных работ, система предлагает не только виртуальное проведение экспериментов, но и мгновенную обратную связь, автоматическое оценивание выполненных заданий. Это способствует более глубокому усвоению материала и развитию практических навыков.

1. АНАЛИЗ ИСПОЛЬЗОВАНИЯ ВИРТУАЛЬНЫХ ЛАБОРАТОРИЙ

1.1. Особенности виртуальных лабораторий в образовании

Виртуальные лаборатории представляют собой цифровые среды, имитирующие реальные лабораторные условия для проведения экспериментов.

Их ключевые особенности:

- **Безопасность:** исключают риски, связанные с работой с опасными веществами или оборудованием.
- **Доступность:** позволяют проводить эксперименты в любое время и в любом месте, что особенно важно для дистанционного обучения.
- **Автоматизация оценки:** Система автоматически анализирует результаты, снижая нагрузку на преподавателей.

Однако традиционные виртуальные лаборатории часто ограничены:

- **Линейность сценариев:** Студенты следуют жестким инструкциям, что снижает творческий подход.

1.2. Роль автоматизации и геймификации в виртуальных лабораториях

Автоматизация и элементы геймификации решают ключевые проблемы классических виртуальных лабораторий:

1. **Снижение вовлеченности:** Монотонные эксперименты становятся интерактивными.
2. **Отсутствие обратной связи:** Студенты получают мгновенные оценки и рекомендации.

Эффективные механизмы, примененные в системе:

- **Динамическая оценка:** учитывает время выполнения, количество ошибок и сложность задания.
- **Визуализация процессов:** Анимации реакций и инструменты анализа (графики, спектры) делают обучение наглядным.
- **Прогресс и достижения:** Студенты видят свой рост, а преподаватели — общую успеваемость групп.

Исследования (например, Deterding et al., 2011) подтверждают, что такие подходы повышают вовлеченность на 40–60% и улучшают усвоение материала.

В «ИС ВиртЛаб» это реализовано через:

- Интерактивные сценарии: например, получение солей через реакции смешивания

1.3. Анализ конкурентов и существующих решений

На рынке образовательных технологий представлено несколько решений для виртуальных лабораторий, каждое из которых имеет свои особенности и ограничения:

1. Labster – предлагает высококачественные 3D-симуляции химических и биологических экспериментов с детализированной графикой. Однако система требует мощного оборудования, имеет высокую стоимость подписки и ограниченную адаптацию под индивидуальные учебные программы.
2. PhET Interactive Simulations – бесплатные интерактивные симуляции от Университета Колорадо. Отличаются простотой и доступностью, но охватывают только базовые эксперименты и не поддерживают автоматическую оценку работ студентов.
3. ChemCollective Virtual Labs – ориентированы на химические эксперименты с возможностью создания пользовательских сценариев. Недостаток – устаревший интерфейс и отсутствие интеграции с системами управления обучением (LMS).
4. MEL Chemistry VR – предлагает immersive-опыт в виртуальной реальности. Несмотря на инновационность, требует дорогостоящего VR-оборудования и не подходит для массового использования в учебных заведениях.

Ключевые проблемы существующих решений:

- Высокая стоимость (Labster, MEL Chemistry)
- Технические ограничения (требования к оборудованию)
- Недостаточная гибкость (фиксированные сценарии экспериментов)
- Отсутствие автоматизированной оценки (PhET, ChemCollective)

1.4. Проблемы и перспективы развития ИС

Основные проблемы современных виртуальных лабораторий:

1. Низкая адаптивность: большинство систем не учитывают индивидуальные ошибки студентов и не предлагают персонализированных подсказок.
2. Ограниченная обратная связь: проверка результатов часто ложится на преподавателей, что снижает оперативность обучения.
3. Слабая интеграция с образовательными стандартами и учебными планами.

ИС ВиртЛаб решает эти проблемы за счет:

- Автоматизированной системы оценки, учитывающей:
 - Правильность выполнения этапов эксперимента
 - Время выполнения
 - Количество ошибок
 - Сложность задания
- Гибких сценариев экспериментов с возможностью расширения (например, добавление новых химических реакций).

Перспективы развития системы:

- Интеграция с популярными LMS (Moodle, Blackboard)
- Добавление модулей для других научных дисциплин (физика, биология)
- Развитие AI-компонентов для персонализированных рекомендаций

2.ХАРАКТЕРИСТИКА ОБЪЕКТА РАЗРАБОТКИ

2.1.Функциональные требования к ИС

Система ИС ВиртЛаб реализует комплексный подход к проведению виртуальных химических экспериментов с элементами автоматизированного контроля и оценки. Ключевые функциональные возможности включают:

1. Управление лабораторными работами:

- Преподаватели могут создавать, редактировать и настраивать лабораторные задания, устанавливая параметры:
 - Цель эксперимента
 - Перечень необходимых реактивов и оборудования
 - Последовательность выполнения этапов
 - Критерии оценки (точность, время выполнения, безопасность)

2. Проведение виртуальных экспериментов:

- Студенты получают доступ к интерактивной лаборатории с возможностью:
 - Выбора химических веществ из виртуальной базы (кислоты, щелочи, соли, оксиды)
 - Проведения реакций в интерактивных колбах и пробирках
 - Наблюдения за ходом реакции (изменение цвета, выделение газа, образование осадка)

3. Система автоматической оценки:

- Анализ выполнения работы по параметрам:
 - Правильность последовательности действий
 - Время выполнения задания
- Формирование итоговой оценки с комментариями по ошибкам

4. Игровые элементы для повышения мотивации:

- Рейтинги:
 - Личные показатели прогресса
 - Сравнение результатов с другими студентами группы

5. Администрирование системы:

- Управление пользователями (студенты, преподаватели)
- Настройка прав доступа к лабораторным работам

2.2. Нефункциональные требования к ИС

1. Производительность:

- Время отклика системы не должно превышать:
 - 0.5 с для базовых операций (открытие интерфейса, выбор реагентов)
 - 1.5 с для сложных расчетов (моделирование реакций, генерация отчетов)
- Поддержка до 5 000 одновременных пользователей (с возможностью масштабирования)
- Оптимизация для работы на устройствах с ограниченными ресурсами (минимальные требования: 4 ГБ ОЗУ, 2-ядерный процессор)

2. Безопасность:

- Ролевая модель доступа (студент/преподаватель/администратор)
- Шифрование данных:
 - TLS 1.3 для передачи данных
 - AES-256 для хранения конфиденциальной информации
- Защита от угроз:
 - WAF (Web Application Firewall) для предотвращения DDoS-атак
 - Подготовленные SQL-запросы для исключения инъекций
- Резервное копирование:
 - Ежедневные бэкапы базы данных
 - Хранение резервных копий в геораспределенном хранилище

3. Удобство использования:

- Адаптивный интерфейс с поддержкой:
 - Desktopных браузеров (Chrome, Firefox, Edge)
 - Мобильных устройств (iOS/Android)

- Сенсорного ввода для планшетов
- Доступность:
 - Поддержка стандарта WCAG 2.1 (уровень AA)
 - Режим высокой контрастности
 - Совместимость с screen-readers (NVDA, VoiceOver)
- Локализация:
 - Русский и английский языки интерфейса
 - Возможность добавления новых языков без изменения кода

4. Масштабируемость:

- Микросервисная архитектура с выделенными модулями:
 - Сервис аутентификации
 - Движок виртуальных экспериментов
 - Система оценки результатов
- Горизонтальное масштабирование:
 - Автоматическое добавление серверов при росте нагрузки
 - Балансировка нагрузки между узлами
- API для интеграции:
 - RESTful API с документацией Swagger
 - Поддержка стандартов LTI для LMS

5. Надежность:

- Требования к uptime:
 - 99.95% доступности в рабочее время (08:00-22:00)
 - Плановые технические работы не чаще 1 раза в месяц
- Система мониторинга:
 - Prometheus + Grafana для сбора метрик
 - Автоматические оповещения при сбоях
- Стратегия восстановления:
 - Репликация базы данных в реальном времени
 - Возможность отката на стабильную версию в течение 15 минут

2.3.Целевая аудитория ИС

Система ИС ВиртЛаб предназначена для следующих ключевых групп пользователей:

1. Студенты

Профиль:

- Возраст: 16-25 лет
- Уровень подготовки: от школьников до студентов вузов
- Технические навыки: базовое владение компьютером

Цели использования:

- Выполнение лабораторных работ в дистанционном формате
- Подготовка к практическим занятиям
- Повторение и закрепление материала

Особые потребности:

- Наглядная визуализация химических процессов
- Точность результатов проведённых экспериментов

2. Преподаватели

Профиль:

- Преподаватели химии в школах и вузах
- Методисты образовательных учреждений
- Научные руководители

Функциональные возможности:

- Создание и настройка лабораторных работ
- Формирование учебных групп
- Автоматизированная проверка заданий
- Анализ успеваемости студентов

Ключевые требования:

- Гибкая система оценивания
- Подробная аналитика по группам
- Интеграция с электронными журналами

3. Администраторы системы

Профиль:

- IT-специалисты образовательных учреждений
- Системные администраторы
- Техническая поддержка

Основные задачи:

- Управление пользователями и правами доступа
- Мониторинг работоспособности системы
- Обновление и сопровождение ПО
- Резервное копирование данных

Требования к системе:

- Централизованное управление
- Подробные логи действий
- Инструменты диагностики

2.4.Ограничения и технические требования к системе

Технические ограничения:

- Гарантии корректной работы в браузерах:
 - Chrome
 - Yandex
 - Edge
 - Safari
- Минимальные требования к оборудованию:
 - Для ПК: 4 ГБ ОЗУ, видеокарта с поддержкой WebGL
 - Для мобильных устройств: Android 8+/iOS 12+
- Требования к интернет-соединению:
 - Минимальная скорость 5 Мбит/с
 - Обязательное подключение для проведения экспериментов
 - Локальное кэширование данных для повторного просмотра результатов

Бизнес-ограничения:

- Первоначально доступно только для образовательных учреждений
- Ограниченный набор химических реакций в первой версии (3 базовых эксперимента)

Правовые аспекты:

- Соответствие:
 - ФЗ-152 "О персональных данных" (РФ)
 - GDPR (для международной версии)
 - Образовательным стандартам ФГОС
- Система модерации:
 - Контроль создаваемых преподавателями заданий
 - Журналирование всех действий пользователей

2.5. Перспективы развития ИС

Технологическое развитие:

1. Расширение функционала:

- Добавление новых категорий экспериментов:
 - Органическая химия (20+ реакций)
 - Биохимические процессы
 - Физико-химические методы анализа
- Введение многоуровневых сценариев:
 - Последовательные эксперименты
 - Альтернативные методы синтеза

2. Интеграционные возможности:

- Подключение к электронным журналам (Дневник.ру, ЭлЖур)
- API для интеграции с Moodle, Blackboard
- Поддержка SCORM-пакетов

3. Инновационные технологии:

- VR-режим для сложных экспериментов (2025 г.)
- AR-поддержка для визуализации молекул (2026 г.)
- AI-ассистент для подсказок (2025 г.)

Образовательные перспективы:

- Разработка готовых учебных курсов:
 - Базовый курс химии (10-11 класс)
 - Практикум для вузов
 - Олимпиадная подготовка
- Создание системы сертификации:
 - Электронные дипломы
 - Система баллов ECTS

Разработка ведется с учетом как текущих образовательных потребностей, так и перспектив цифровизации естественнонаучного образования. Платформа имеет модульную архитектуру, позволяющую легко добавлять новые функции и направления развития.

3.ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

3.1.Архитектура разрабатываемой системы

Система реализована по модульному принципу с четким разделением на уровни:

1.Клиентский уровень:

Технологический стек:

- React (v18+) с TypeScript для строгой типизации
- HeroUI как компонентная библиотека
- React Router для навигации
- Axios для HTTP-запросов
- DRAG AND DROP – для проведения химических экспериментов

2.Серверный уровень:

- Node.js (v18 LTS)
- Express.js (REST API)
- JavaScript на сервере
- JWT для аутентификации

Микросервисная архитектура:

- Сервис администратора
- Сервис пользователя

3.Уровень данных:

- PostgreSQL (основное хранилище)

3.2.Проектирование базы данных

Пользователи (User)

- Основные учетные данные (email, пароль, ФИО)
- Ролевая модель (role_name: student/teacher/admin)
- Статус аккаунта (is_blocked, email_is_verify)
- Активность (created_at, last_login)

Группы (Group)

- Учебные группы (group_number)

- Уникальный хэш для вступления (hash_code_login)
- Привязка к преподавателю (ID_user)
- Пользователи в группах (Users_in_group)

Задания (Task)

- Базовые параметры (title, description)
- Доступность (is_available)

Задания для групп (Task_for_group)

- Параметры доступа (is_open, deadline)

Результаты (User_make_task)

- Оценка (score)
- Комментарии (comment_user, comment_teacher)
- Временные метки (date_start, date_finish)
- Контроль удаления (is_deleted)

Поддержка (Support)

- Обращения пользователей (user_text)
- Ответы администраторов (admin_response)
- Статус обработки (status)

Черный список (Blacklist)

- Заблокированные пользователи (id_user)
- Причина блокировки (reason)
- Дата добавления (date_added)

Платежи (Purchases)

- Доступ к платным заданиям (id_task)
- Статус оплаты (is_paid)
- Метод оплаты (payment_method)
- Финансовые метрики (price)

Администраторы (Admin)

- Отдельная таблица для админсостава
- Контроль активности (is_active)

3.3.Проектирование API

API-эндпоинты администратора:

1. Авторизация и аутентификация

POST <https://localhost:5001/auth/login>

- Аутентификация администратора
- Принимает: {email, password}
- Возвращает: JWT-токен и базовую информацию о пользователе

GET <http://localhost:5001/api/auth/getInfoAboutSelf>

- Получение информации о текущем авторизованном администраторе
- Требуется авторизации

GET <https://localhost:5001/auth/check>

- Проверка валидности токена
- Используется для проверки сессии

2. Управление черным списком

POST <https://localhost:5001/blacklist/addToBlacklist>

- Добавление пользователя в черный список
- Параметры: {id_user, reason}

POST <https://localhost:5001/blacklist/removeFromBlacklist>

- Удаление пользователя из черного списка
- Параметр: {id_user}

GET <https://localhost:5001/blacklist/getAll>

- Получение полного списка заблокированных пользователей
- Возвращает массив записей с причинами блокировок

3. Работа с обращениями в поддержку

GET <https://localhost:5001/support/getAllAppeal>

- Получение всех обращений в техподдержку
- Возвращает список с статусами и текстами обращений

POST https://localhost:5001/support/sendResponseToSupport

- Отправка ответа на обращение
- Параметры: {id_support, admin_response, status}

4. Управление пользователями

GET https://localhost:5001/api/user/getAllStudents

- Получение списка всех студентов
- Возвращает массив с ФИО, email и статусом аккаунта

GET https://localhost:5001/user/getAllTeachers

- Получение списка всех преподавателей
- Включает информацию о привязанных группах

POST https://localhost:5001/user/editUserByID

- Редактирование данных пользователя
- Параметры: {id_user, email, lastname, firstname, middlename, role_name}

5. Управление заданиями

GET https://localhost:5001/task/getAll

- Получение всех лабораторных заданий
- Возвращает список с названиями и описаниями

POST https://localhost:5001/task/add

- Добавление нового задания
- Параметры: {title, description, is_available}

POST https://localhost:5001/task/changeAvailableById

- Изменение доступности задания
- Параметры: {id_task, is_available}

6. Управление платежами

GET https://localhost:5001/purchase/getAll

- Получение всех платежных транзакций

- Возвращает историю оплат с детализацией

POST https://localhost:5001/purchase/changelsPaidById

- Изменение статуса оплаты
- Параметры: {id_purchases, is_paid}

POST https://localhost:5001/purchase/changelsBlockedById

- Блокировка/разблокировка платежа
- Параметры: {id_purchases, id_blocked}

API-эндпоинты для пользователей (кроме администратора):

Аутентификация и профиль:

- POST /auth/registration - Регистрация нового пользователя
 - Параметры: {email, password, lastname, firstname, middlename, role}
 - Возвращает: JWT токен
- POST /auth/login - Вход в систему
 - Параметры: {email, password}
 - Возвращает: JWT токен
- POST /auth/editSelfFromToken - Редактирование своего профиля
 - Параметры: {email, lastname, firstname, middlename}
- GET /api/auth/getInfoAboutSelf - Получение информации о текущем пользователе
- GET /auth/check - Проверка валидности токена

Техническая поддержка:

- POST /support/sendToSupport - Отправка обращения в поддержку
 - Параметры: {message_text}
- GET /support/getListMyAppeal - Получение списка своих обращений

API-эндпоинты преподавателя

Управление группами:

- POST /teacher/group/create - Создание новой учебной группы

- Параметры: {group_name, subject}
- GET /teacher/group/getAllMyGroups - Получение списка своих групп
- POST /teacher/group/addUserToGroup - Добавление студента в группу
 - Параметры: {group_id, user_email}
- POST /teacher/group/removeFromGroupByEmail - Удаление студента из группы
 - Параметры: {group_id, user_email}
- POST /teacher/group/deleteById - Удаление группы
 - Параметры: {group_id}
- POST /teacher/group/grantRightsToGroup - Предоставление прав на группу другому преподавателю
 - Параметры: {group_id, teacher_email}
- POST /teacher/group/changeIsOpenById - Изменение статуса доступности группы
 - Параметры: {group_id, is_open}
- GET /teacher/group/getAllMyAccess - Получение списка групп с доступом
- GET /api/teacher/group/getGroupAttempts - Получение попыток выполнения заданий группой
 - Параметры: {group_id}

Управление заданиями:

- GET /teacher/tasks/getAllAvailable - Получение списка доступных заданий

Платежи и подписки:

- POST /teacher/purchase/add - Создание платежа за доступ к заданиям
 - Параметры: {task_id, group_id, price}
- POST /teacher/purchase/confirm - Подтверждение оплаты
 - Параметры: {purchase_id}
- GET /teacher/purchase/getAllMy - Получение своих платежей

API-Эндпоинты студента

Работа с группами:

- POST /student/group/addSelfToGroup - Вступление в группу
 - Параметры: {group_hash}
- POST /student/group/removeSelfFromGroup - Выход из группы
 - Параметры: {group_id}
- GET /api/student/group/getGroupsWhereiAmMember - Получение списка своих групп

Работа с заданиями:

- GET /student/task/getUserGroupsWithTasks - Получение групп с доступными заданиями
- GET /student/task/getTasksWithGroups - Получение заданий по группам
- GET /student/task/getUserTaskAttempts - Получение своих попыток выполнения заданий
- POST /student/task/createUserTaskAttempt - Начало выполнения задания
 - Параметры: {task_id}
- POST /student/task/finishUserTaskAttempt - Завершение выполнения задания
 - Параметры: {attempt_id, results_data}
- GET /api/student/task/getSelfAttempts - Получение истории своих попыток

3.4.Проектирование интерфейсов для доступа к функционалу ИС

Первичной точкой взаимодействия пользователя с сервисом является главная страница (см. Приложение Б, рис. 1-2).

При нажатии на кнопку “Зарегистрироваться” пользователь попадает на страницу регистрации, где он должен пройти соответствующую процедуру, введя данные о себе, а также электронную почту и пароль для последующего входа в аккаунт.

Для пользователей с разными ролями предусмотрены соответствующие требуемому функционалу web-интерфейсы. (см Приложение Б, рис. 5-6, 18).

Для пользователя с ролью “студент” предусмотрены следующие вкладки с соответствующим функционалом:

- Главная – Общая информация об аккаунте и возможность редактирования аккаунта (см. Приложение Б, рис. 5-6).
- Мои группы – Возможность просмотра и присоединения к группам с другими студентами для последующего получения прав на выполнение заданий (см. Приложение Б, рис. 11).
- Задания – Возможность просмотра и перехода к выполнению доступных студенту заданий (см. Приложение Б, рис. 12).
- Мои результаты – Возможность просмотра результатов выполненных заданий текущим аккаунтом (см. Приложение Б, рис. 13).
- Поддержка – Возможность отправить обращение в поддержку (см. Приложение Б, рис. 10)

Для пользователя с ролью “Преподаватель” предусмотрены следующие вкладки с соответствующим функционалом:

- Все вкладки, доступные пользователю с ролью “Студент”
- Управление группами – Возможность просмотра, создания, удаления, редактирования состава групп, а также наделение группы правами на выполнение доступных заданий. (см. Приложение Б, рис. 7)
- Подписка – Возможность оформления подписки для предоставления аккаунту доступа к полному функционалу сервиса. (см. Приложение Б, рис. 8)
- Результаты группы – Возможность просмотра результатов выполнения студентами заданий. (см. Приложение Б, рис. 9)

Для пользователя с ролью “Администратор” точкой входа является

другая форма авторизации, которая располагается по другому URL-адресу (см. Приложение Б, рис. 17). Для администратора предусмотрены следующие вкладки с соответствующим функционалом:

- Главная – Возможность просмотра общей информации об аккаунте администратора. (см. Приложение Б, рис. 18)
- Задания – Возможность управления доступности заданий. (см. Приложение Б, рис. 19)
- Подписки – Управление подписками пользователей. (см. Приложение Б, рис. 20)
- Пользователи – Просмотр, редактирование, добавление пользователей в чёрный список. (см. Приложение Б, рис. 21)
- Чёрный список – Управление пользователями в чёрном списке. (см. Приложение Б, рис. 22)
- Поддержка – Управление обращениями пользователей, обработка и отправка ответов на обращения. (см. Приложение Б, рис. 23)

3.5.Используемые технологии обеспечения информационной безопасности

Токен-система:

- JWT Access Token (HS512, срок жизни 15 мин)
 - Хранение: in-memory (клиент)
 - Полезная нагрузка: {userId, role, sessionId}
- Refresh Token (срок жизни 7 дней)
 - Хранение: HttpOnly cookie (Secure, SameSite=Strict)
 - Механизм ротации при каждом использовании

Защитные механизмы:

- Rate Limiting (оконный алгоритм):
 - 5 запросов/мин для /auth/*
 - 30 запросов/мин для API-эндпоинтов
- Защита от brute force:
 - Блокировка после 5 неудачных попыток (на 30 мин)
 - CAPTCHA для повторного входа
- Анализ аномалий:
 - Детекция необычной геолокации
 - Контроль устройств (fingerprinting)

2. Защита данных

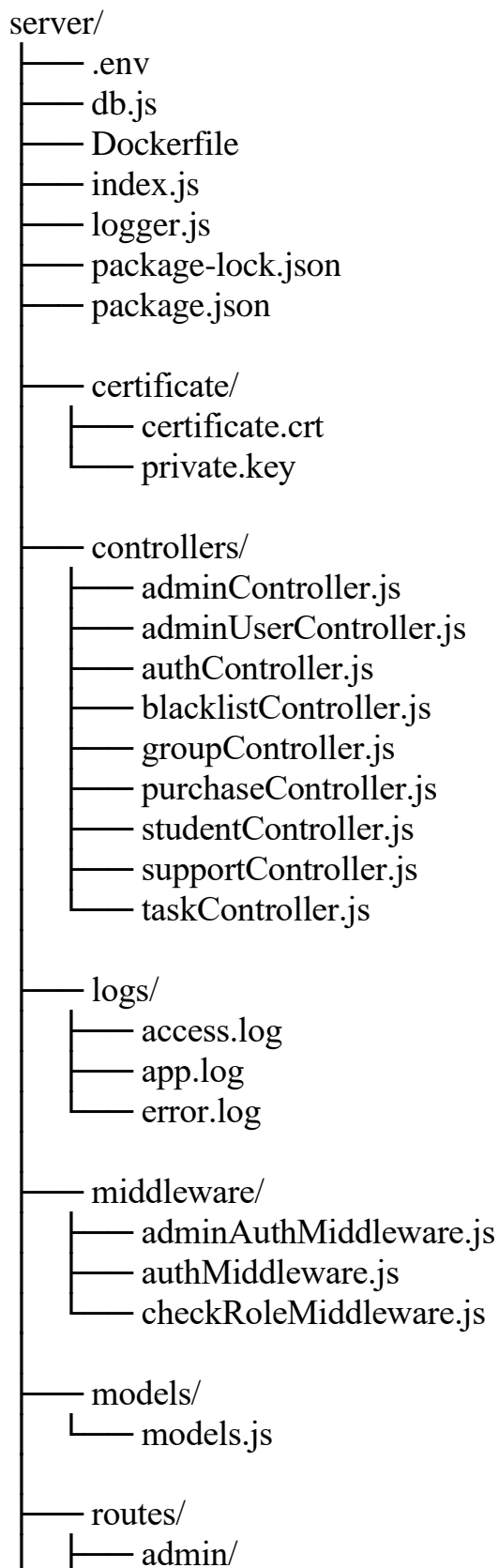
Шифрование:

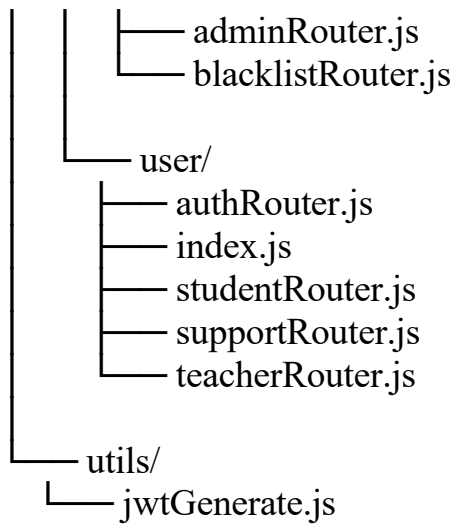
- Транзитные данные:
 - TLS 1.3 с PFS (Perfect Forward Secrecy)
 - Принудительное HSTS (Strict-Transport-Security)
- Хранение паролей:
 - Алгоритм: bcrypt (cost factor=12)
 - Перед хешированием: ключевое растяжение PBKDF2
- Конфиденциальные данные:
 - Полевое шифрование (AES-256-GCM)
 - Для ФИО, email, учебных результатов

4.РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

4.1.Реализация серверной части

Структура директорий серверной части:

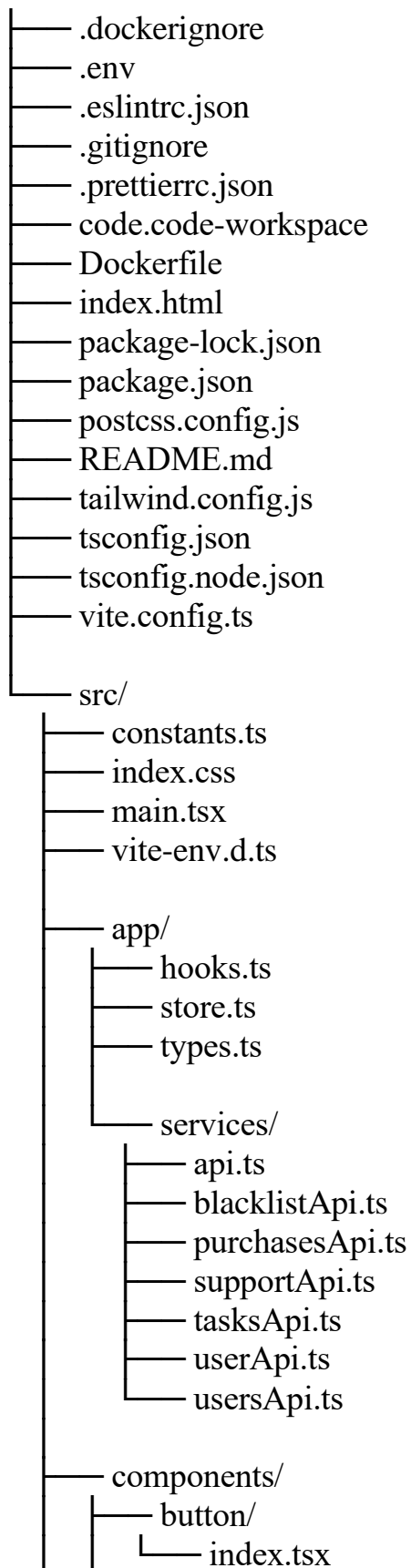


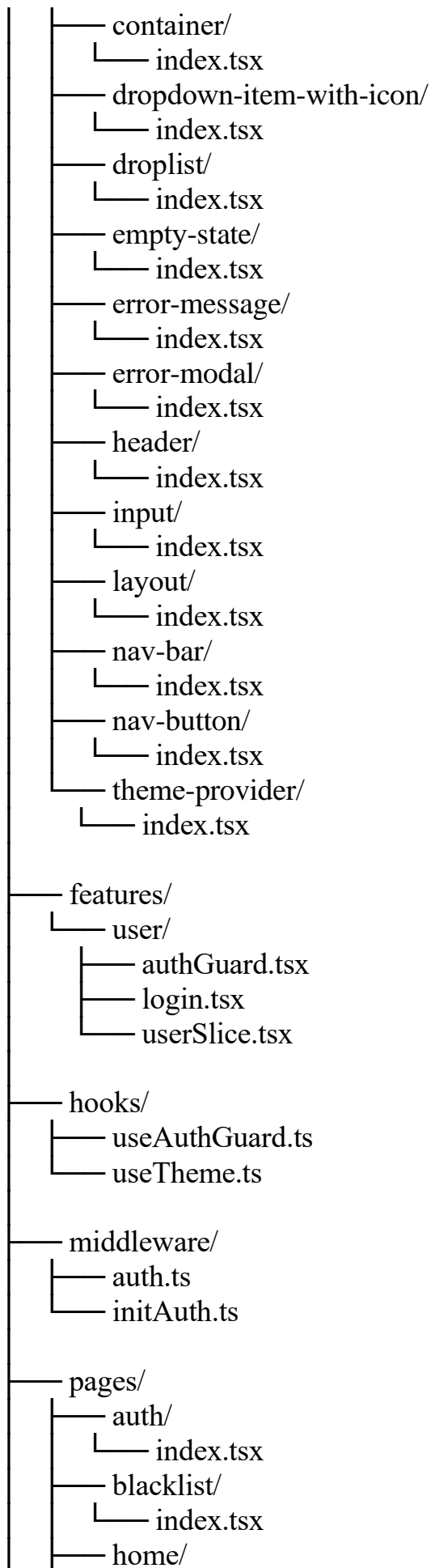


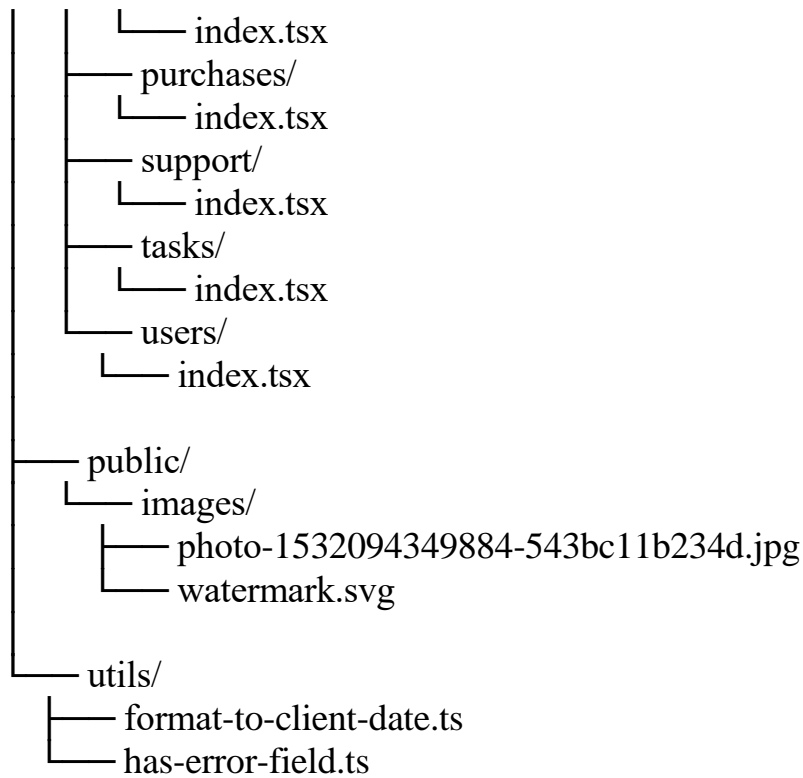
4.2. Реализация клиентской части администратора и пользователя

Структура директорий клиентской части администратора:

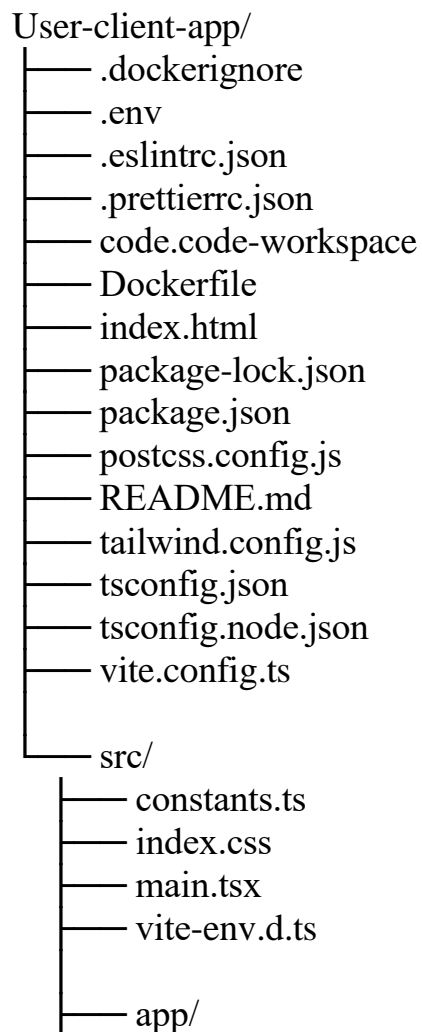
Admin/

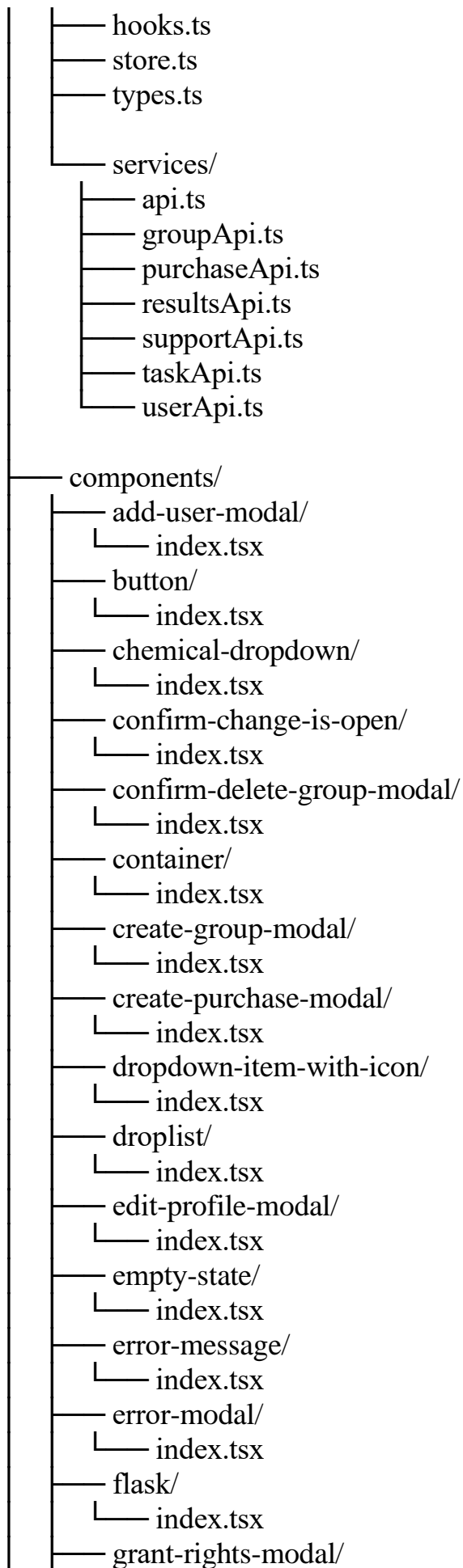


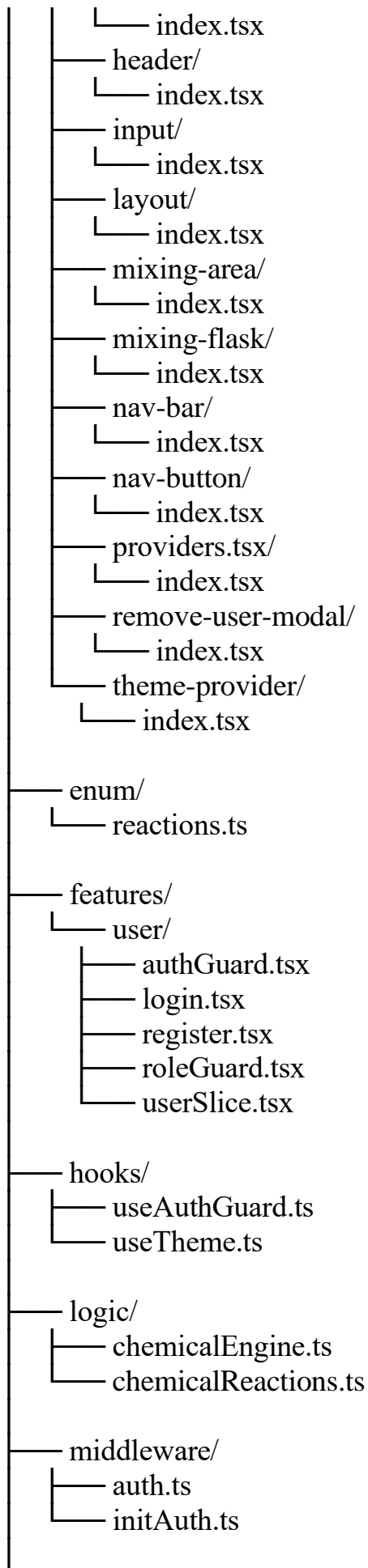


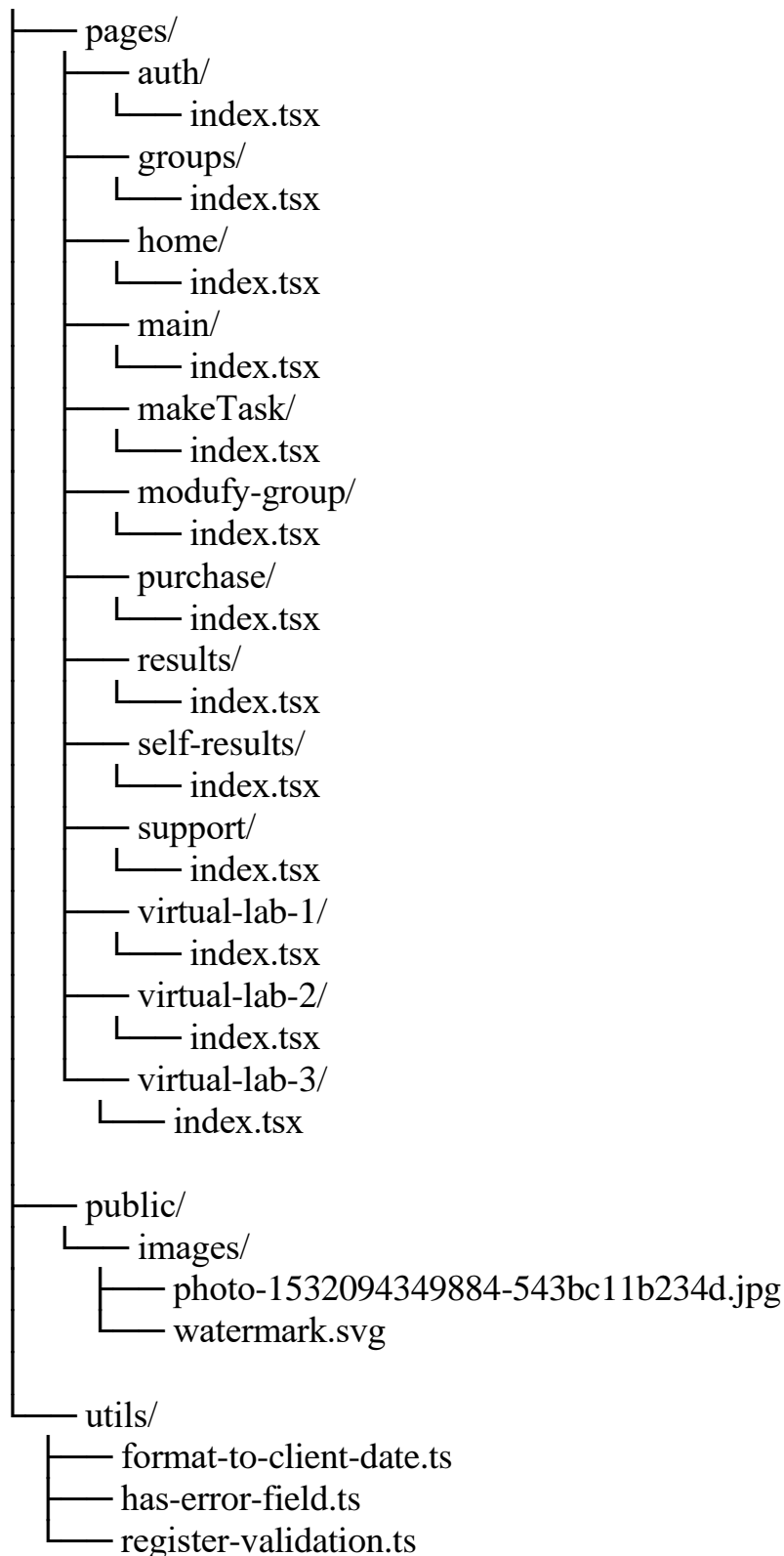


Структура директорий клиентской части пользователя:









4.3. Проведённое тестирование и обеспечение качества

В результате тестирования были проверены следующие функции, на основании чего гарантируется работоспособность всех основных функций системы:

1. Авторизация и профиль

- Регистрация нового пользователя (студент/преподаватель)
- Вход в систему по логину/паролю
- Просмотр и редактирование профиля
- Выход из системы

2. Функционал студента

- Просмотр доступных лабораторных работ
- Выбор и запуск эксперимента
- Работа с виртуальными реагентами (перетаскивание, смешивание)
- Наблюдение за ходом реакции (анимации, изменения)
- Сохранение промежуточных результатов
- Формирование отчета по эксперименту
- Просмотр истории выполненных работ
- Анализ ошибок и замечаний преподавателя
- Вступление/выход из учебных групп

3. Функционал преподавателя

- Создание/редактирование учебных групп
- Добавление/удаление студентов в группах
- Назначение лабораторных работ группам
- Просмотр прогресса студентов
- Проверка и комментирование работ
- Настройка критериев оценки
- Генерация сводных отчетов по группе
- Управление доступом к платным материалам

4. Административный функционал

- Управление пользователями (блокировка/разблокировка)
- Просмотр статистики использования системы
- Мониторинг обращений в поддержку
- Управление платежами и подписками
- Настройка системных параметров

5. Лабораторный интерфейс

- Выбор оборудования и реактивов
- Корректность визуализации реакций
- Работа с измерительными приборами
- Система подсказок и предупреждений
- Возможность отмены действий
- Сохранение эксперимента в любой момент

6. Дополнительные функции

- Доступ к методическим материалам
- Темный/светлый режим интерфейса

7. Безопасность и валидация

- Проверка вводимых данных в формах
- Ограничения доступа по ролям
- Защита от некорректных действий
- Валидация химических формул

8. Производительность

- Скорость загрузки интерфейса
- Стабильность при длительных сеансах
- Работа с медленным интернетом

4.4.Развёртывание приложения на сервере с использованием технологий контейнеризации

В проекте используется контейнеризация Docker. А благодаря Docker-compose развёртывание становится совсем простым. Необходимо лишь клонировать репозиторий:

git clone https://github.com/KarmaIIIe4ek/PPS_Kurovaya.git

И запустить процесс создания и запуска контейнеров

docker compose up -d --build

В результате будет запущено 5 docker-контейнеров.

ЗАКЛЮЧЕНИЕ

В ходе разработки системы "ИС ВиртЛаб" был создан комплексный образовательный продукт, сочетающий научную достоверность с современными технологиями. Проект реализован на стеке React/TypeScript для фронтенда и Node.js/PostgreSQL для бэкенда. Основные достижения включают полноценную виртуальную лабораторию с 3 базовыми экспериментами, систему автоматической оценки по трём критериям и механизмы контроля академической успеваемости. Особое внимание уделено UX-дизайну с применением принципов drag-and-drop подхода, что позволило создать многофункциональный интерфейс со специализированными компонентами, включая интерактивные элементы управления экспериментом.

Несмотря на успешную реализацию, в системе остались невоплощенными некоторые перспективные функции, такие как VR/AR-режимы, мобильное приложение и интеграция с лабораторным оборудованием. Текущие технические ограничения включают поддержку ограниченного набора реакций. Однако проект обладает значительным потенциалом для развития - в ближайшей перспективе планируется расширение библиотеки экспериментов до 30+ реакций, разработка мобильного клиента и внедрение ИИ-ассистента. "ИС ВиртЛаб" уже сейчас демонстрирует эффективность в повышении безопасности и доступности химического образования, а её дальнейшее развитие открывает новые возможности для цифровизации естественнонаучного обучения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
2. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированной системы.
3. ГОСТ 34.602-2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
4. Разработка Технического задания по ГОСТ 34 легко и просто – Москва – URL: <https://habr.com/ru/articles/432852>
5. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
6. ГОСТ 2.105-95. ЕСКД. Общие требования к текстовым документам.
7. Рудинский И. Д. – “Методические указания по выполнению курсового проекта для студентов специальности – Автоматизированные системы обработки информации и управления”. Калининград КГТУ 2003
8. ГОСТ Р. 50377-92 (МЭК 950-86) «Безопасность оборудования информационной технологии, включая электрическое конторское оборудование».

ПРИЛОЖЕНИЕ А

Реализация включает в себя более 35 тысяч строк, в связи с чем принято решение не размещать код в документе. Листинг и структуру проекта можно посмотреть в гитхаб-репозитории:

https://github.com/KarmaIIIe4ek/PPS_Kurovaya

ПРИЛОЖЕНИЕ Б

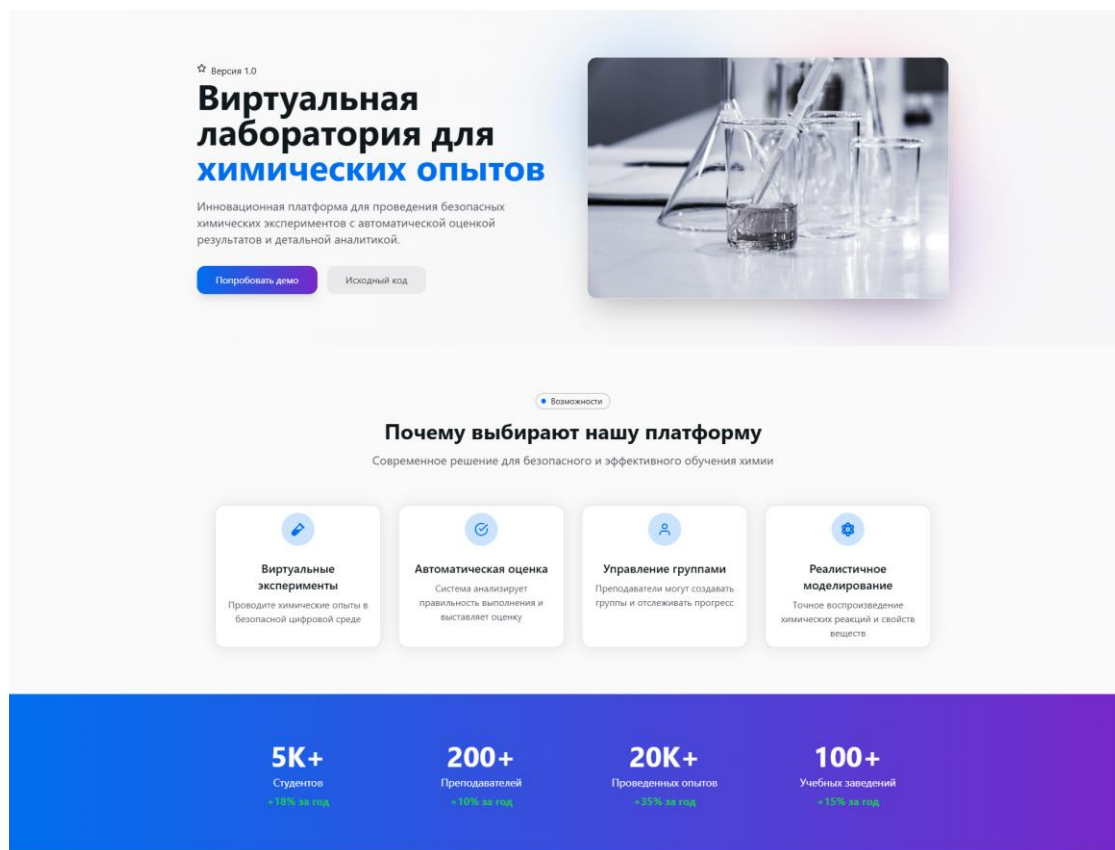


Рисунок 1 – Главная страница с общей информацией о проекте (часть 1)

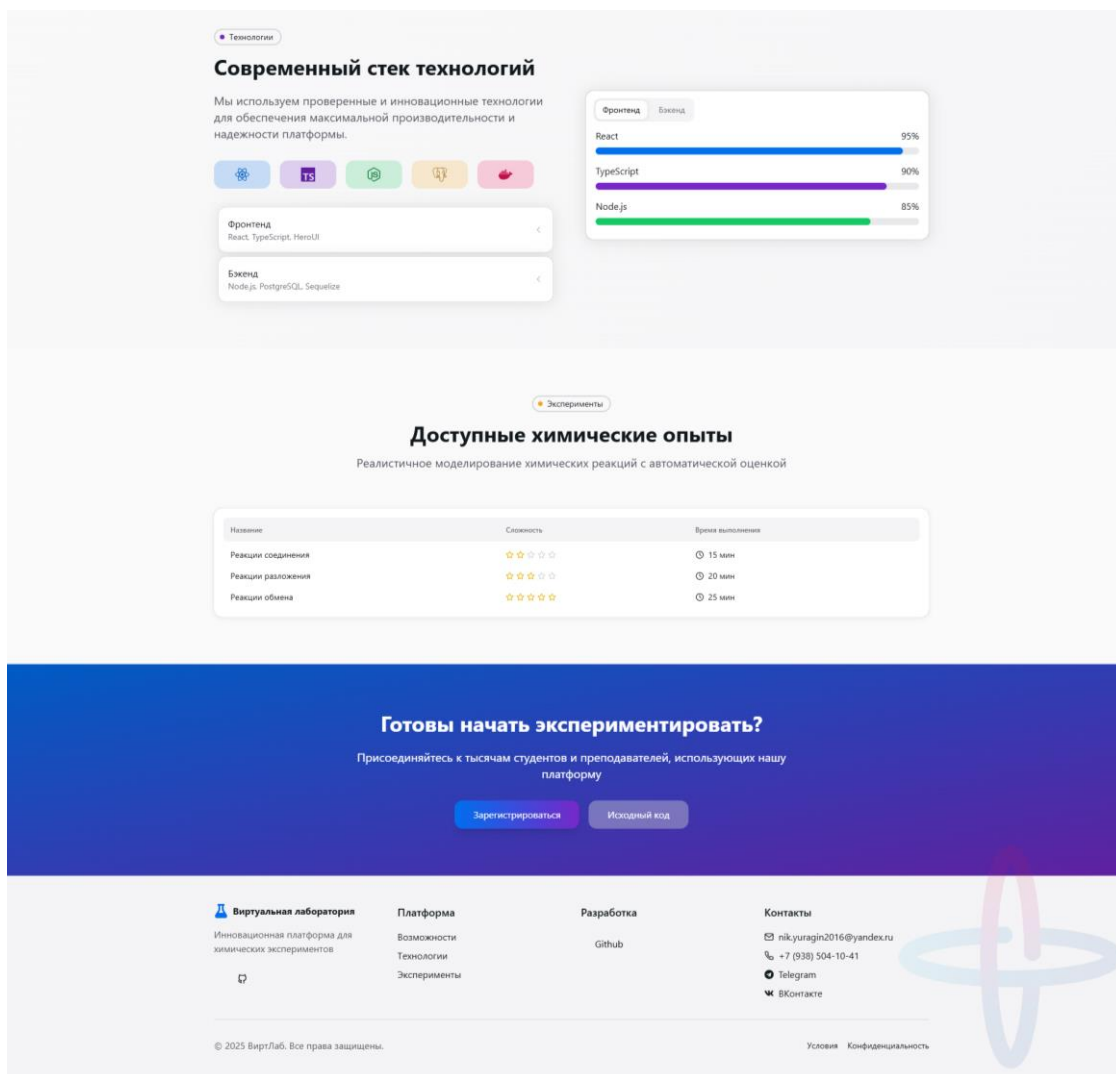


Рисунок 2 – Главная страница с общей информацией о проекте (часть 2)



Добро пожаловать!

→] Вход

👤+ Регистрация

Email

Пароль

Нет аккаунта? [Зарегистрируйтесь](#)

Войти

© 2025 Виртуальная лаборатория. Все права защищены.

Рисунок 3 – Форма авторизации пользователей



Создайте аккаунт

→ Вход

👤+ Регистрация

Фамилия

Имя

Отчество

Роль



Email

Пароль

Подтвердите пароль

Уже есть аккаунт? [Войдите](#)

Зарегистрироваться

© 2025 Виртуальная лаборатория. Все права защищены.

Рисунок 4 – Форма регистрации пользователей

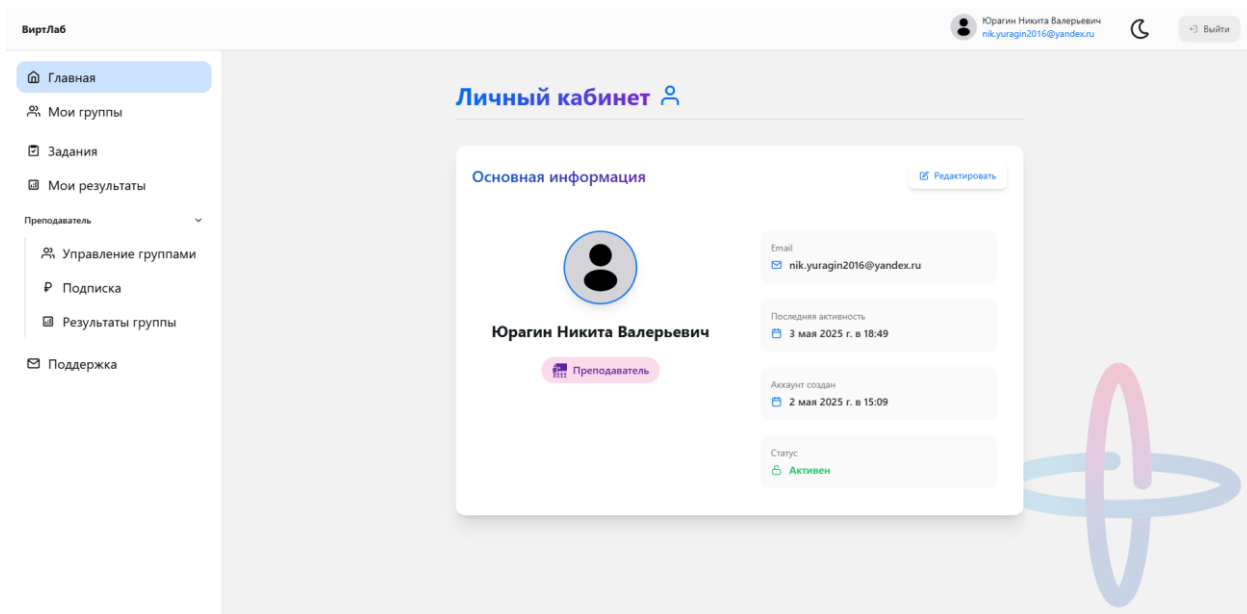


Рисунок 5 – Личный кабинет преподавателя (светлая тема)

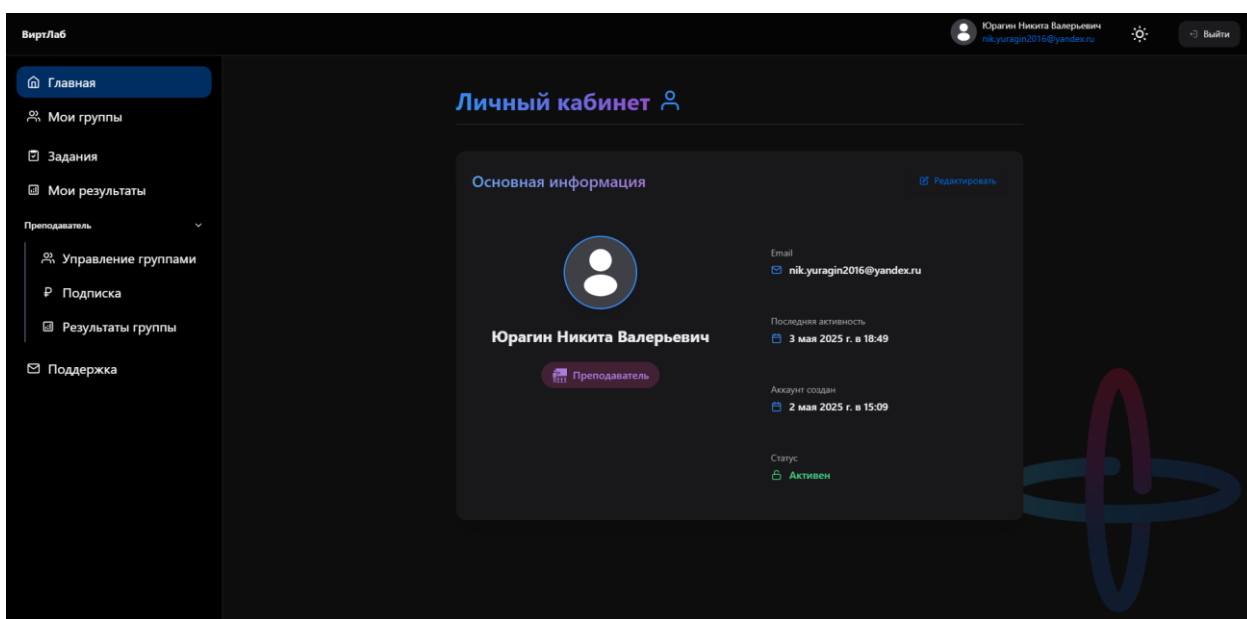


Рисунок 6 – Личный кабинет преподавателя (тёмная тема)

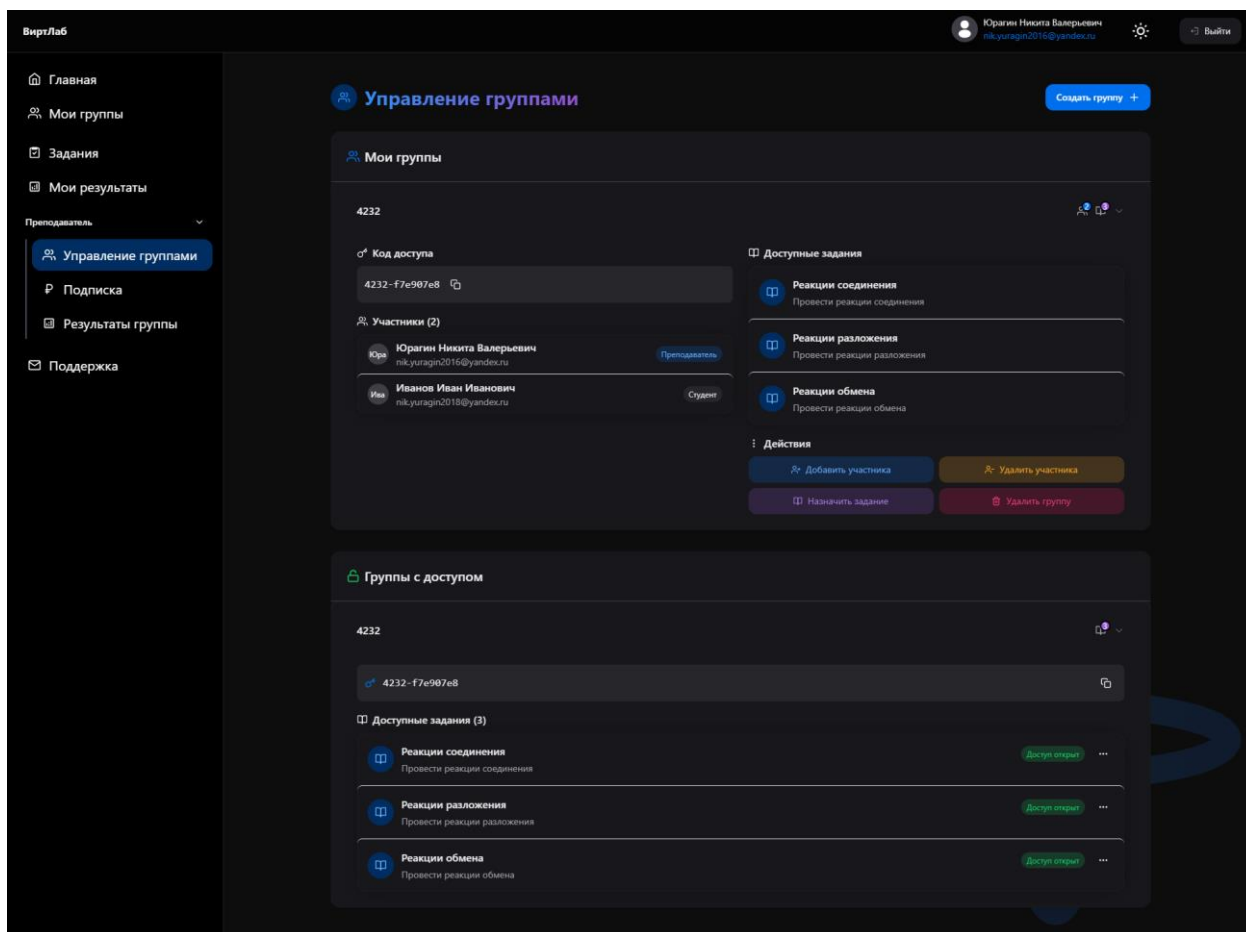


Рисунок 7 – Вкладка управления группами (функционал преподавателя)

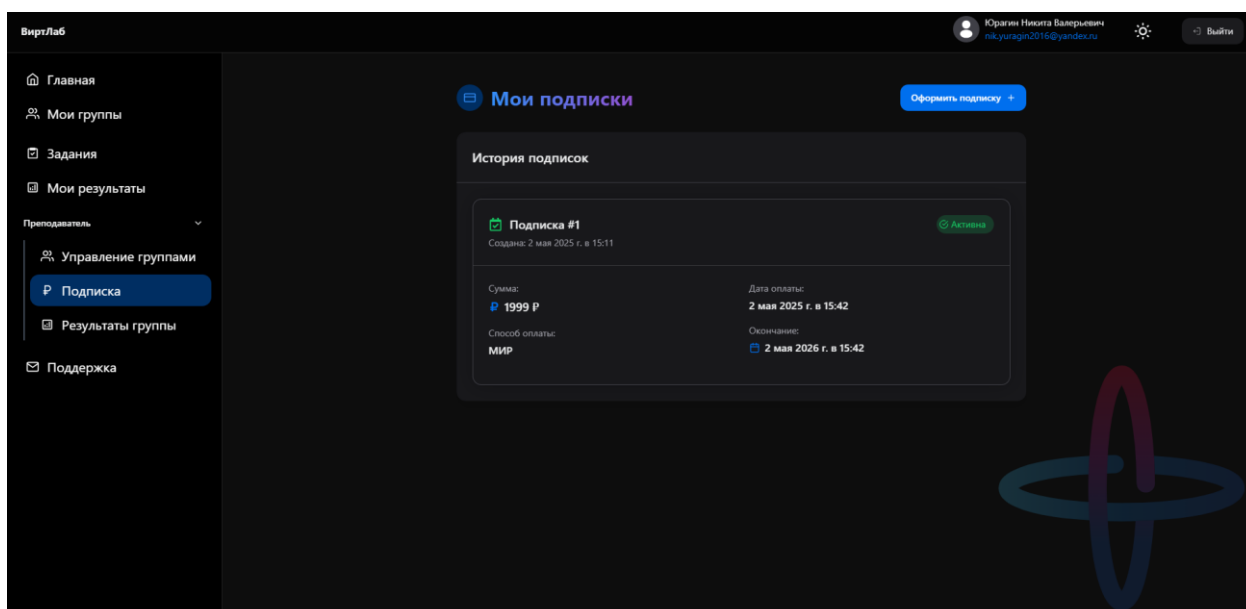


Рисунок 8 – Вкладка управления подпиской (функционал преподавателя)

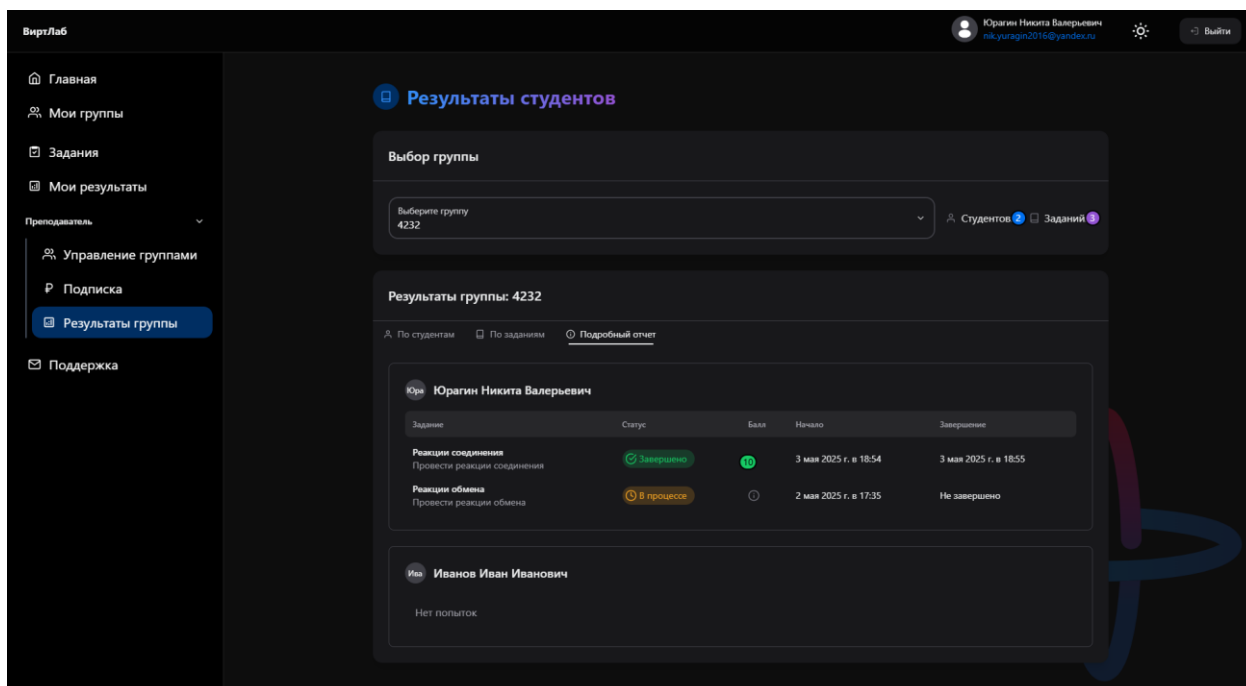


Рисунок 9 – Вкладка просмотра результатов выполнения заданий студентами (функционал преподавателя)

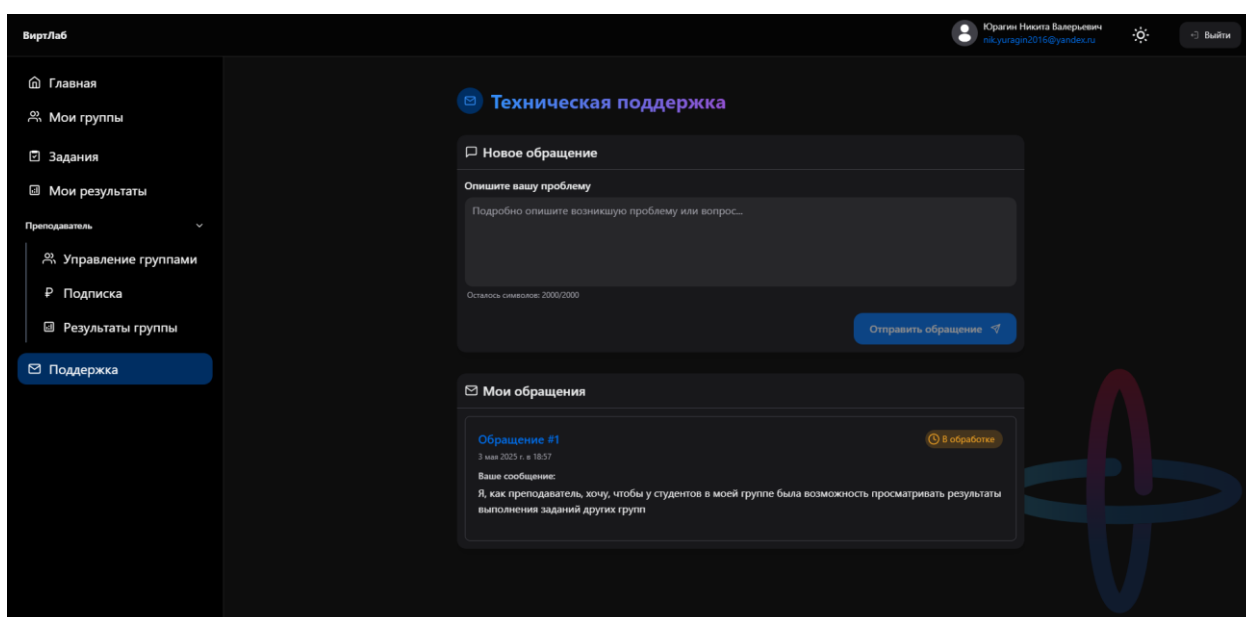


Рисунок 10 – Вкладка обращения в техническую поддержку (общий функционал)

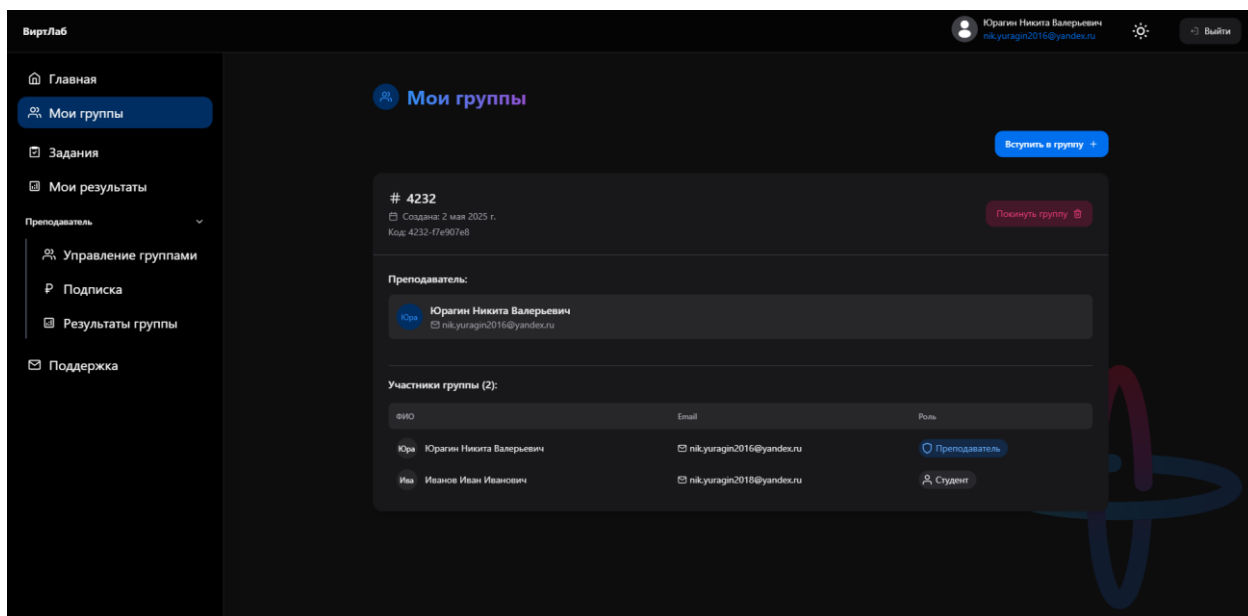


Рисунок 11 – Вкладка просмотра моих групп (функционал студента)

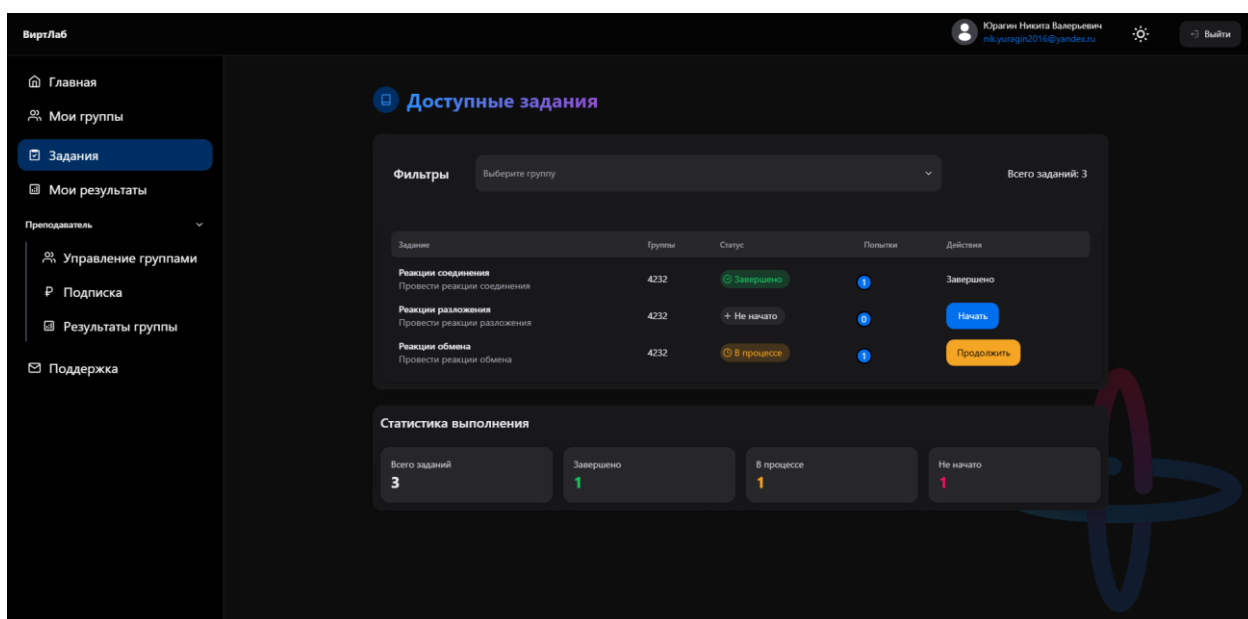


Рисунок 12 – Вкладка просмотра доступных заданий (функционал студента)

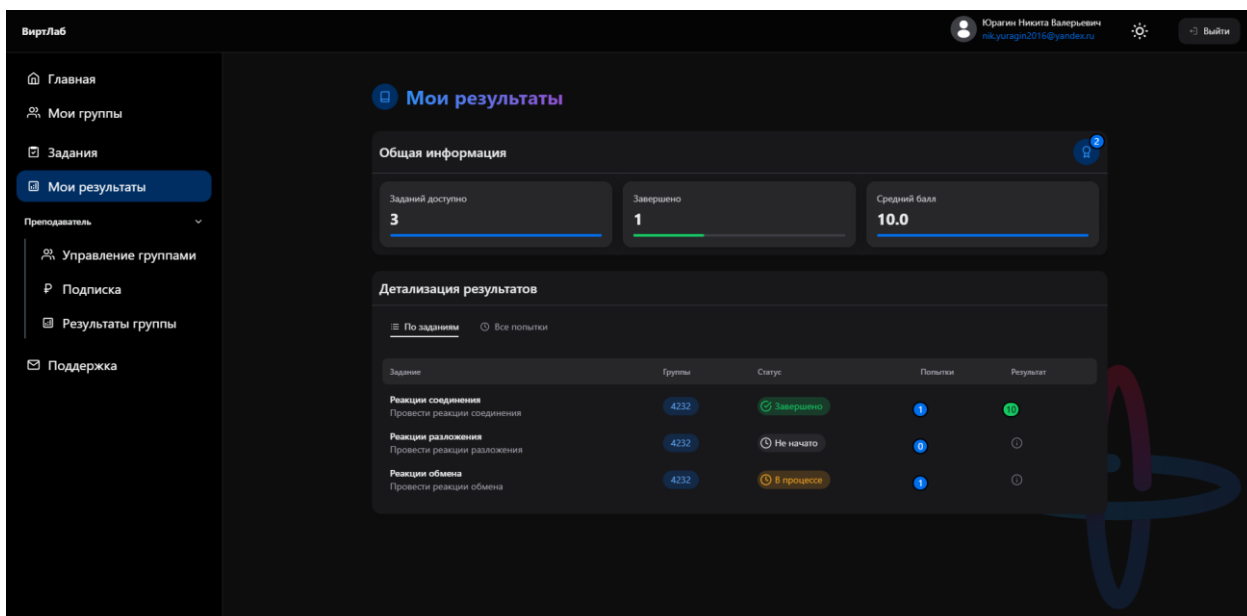


Рисунок 13 – Вкладка просмотра результатов выполненных заданий
(функционал студента)

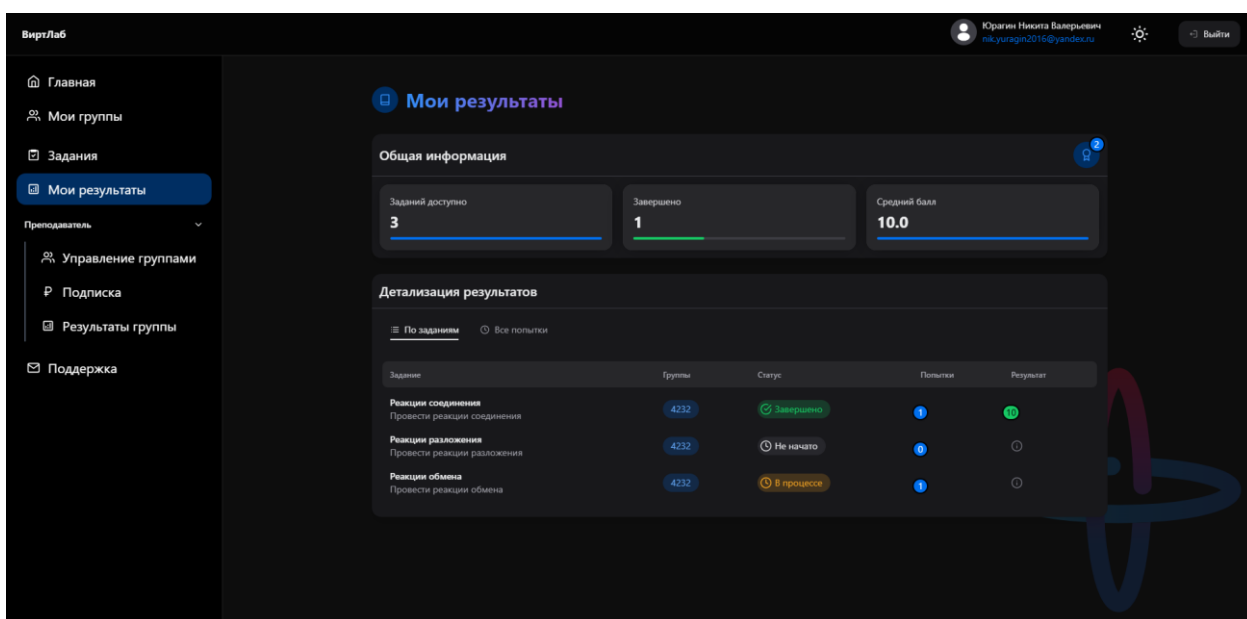


Рисунок 14 – Вкладка просмотра результатов выполненных заданий
(функционал студента)

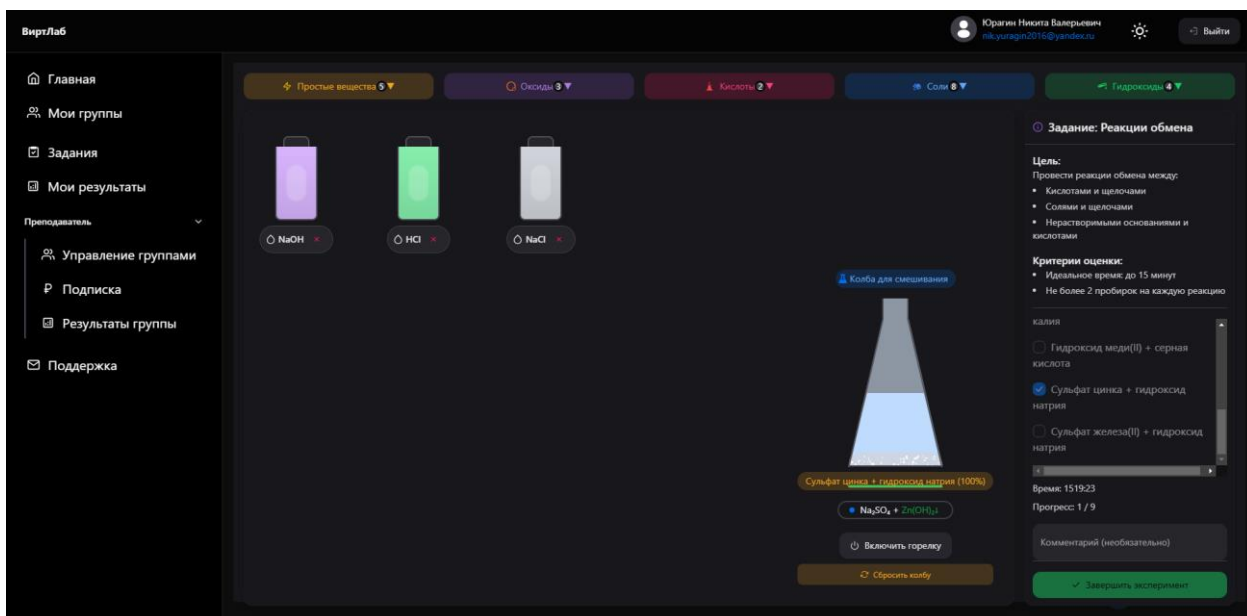


Рисунок 15 – Вкладка выполнения задания (функционал студента)



Рисунок 16 – Вкладка выполнения задания с завершённым заданием
(функционал студента)



Админ-панель

Email

Обязательное поле

Пароль

Обязательное поле

Войти

© 2025 Виртуальная лаборатория. Все права защищены.

Рисунок 17 – Окно авторизации администратора

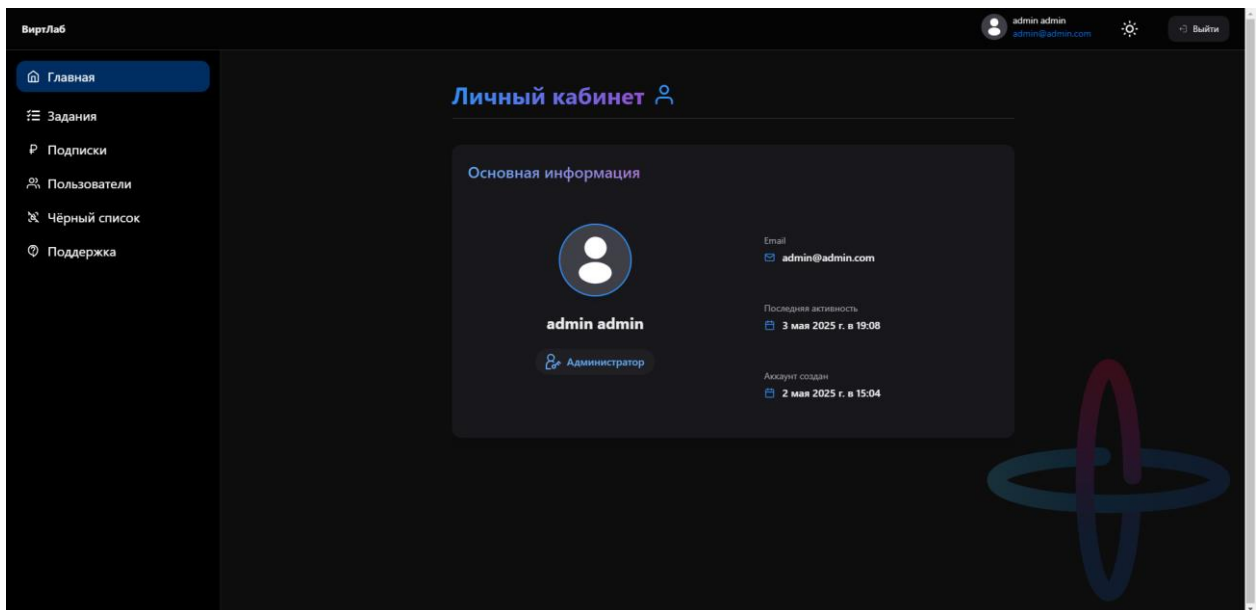


Рисунок 18 – Личный кабинет администратора

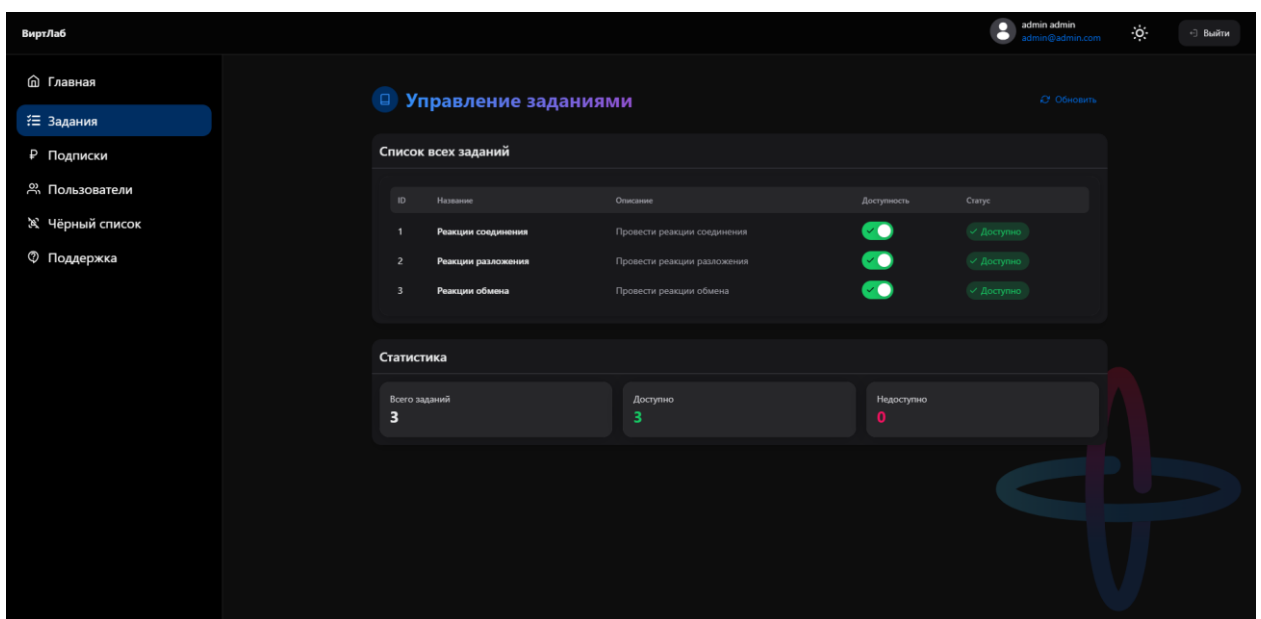


Рисунок 19 – Вкладка управления заданиями

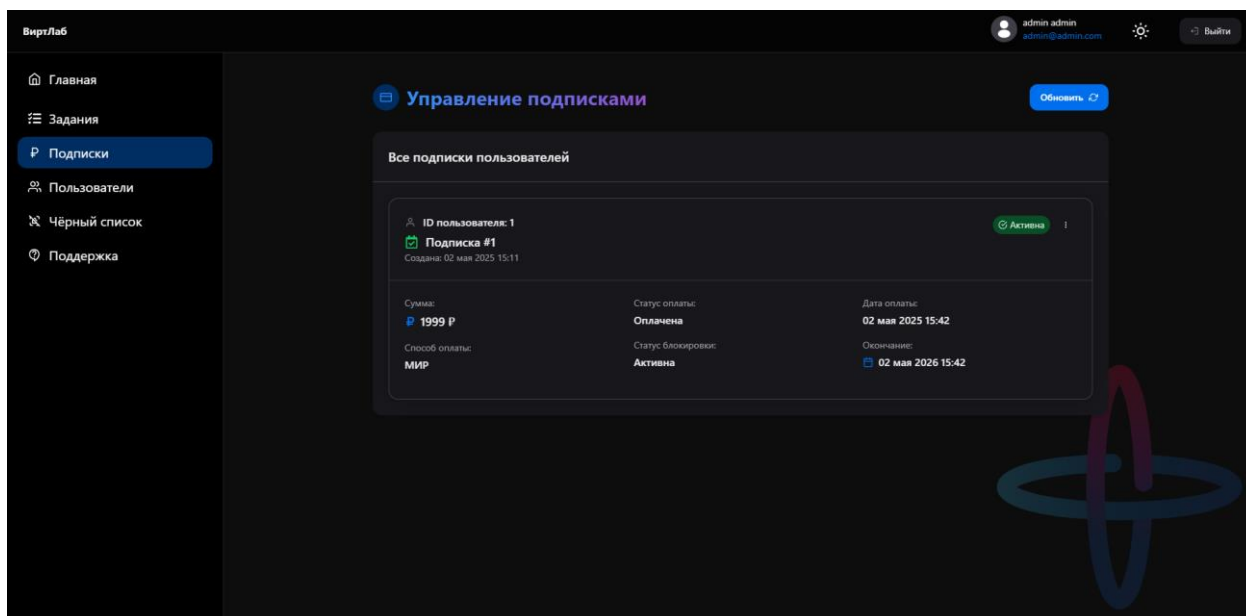


Рисунок 20 – Вкладка управления подписками

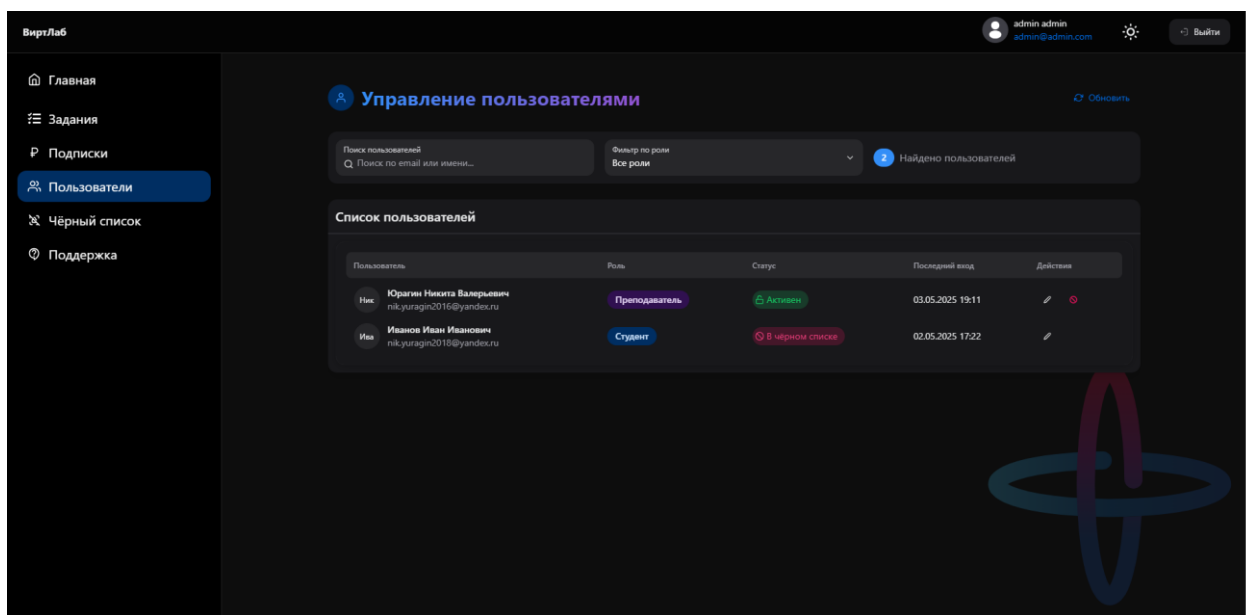


Рисунок 21 – Вкладка управления пользователями

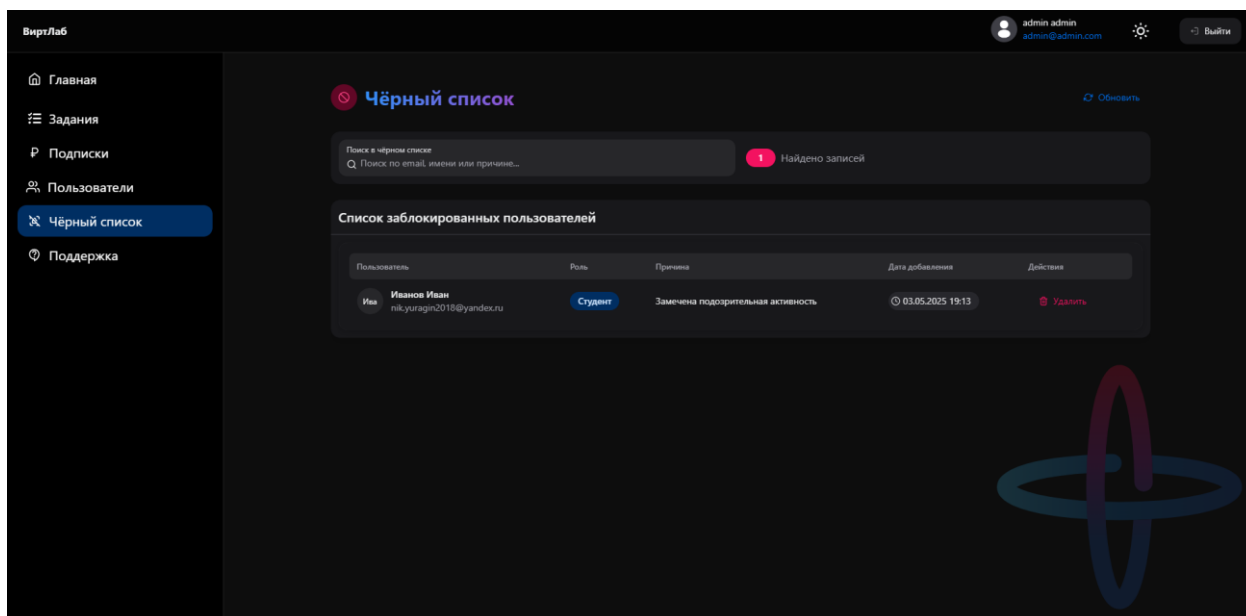


Рисунок 22 – Вкладка управления чёрным списком

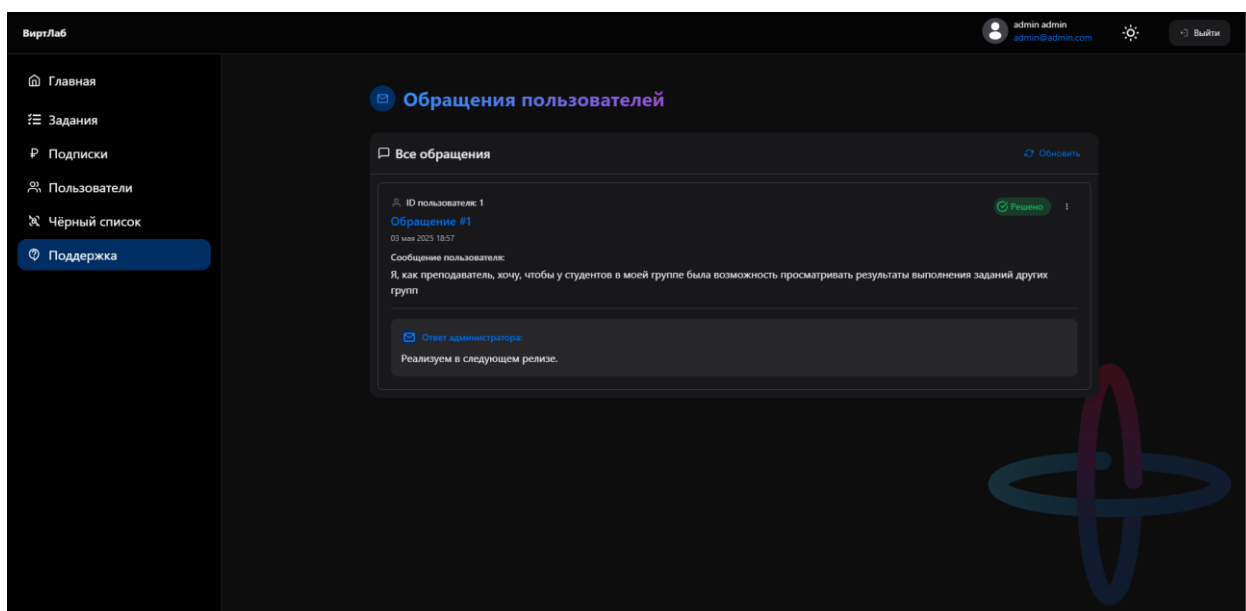


Рисунок 23 – Вкладка управления технической поддержкой

АПРИЛОЖЕНИЕ В (РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ)

Назначение системы

"ИС ВиртЛаб" — это инновационная платформа виртуальных химических лабораторий, предназначенная для безопасного и эффективного изучения химических процессов через браузер. Система позволяет проводить реалистичные эксперименты, получать мгновенную оценку результатов и анализировать прогресс в обучении.

Основные возможности и целевой функционал

- Проведение химических экспериментов в виртуальной среде:
 - Смешивание реагентов
 - Нагревание реагентов
 - Наблюдение за реакциями
- Автоматическая оценка работ по критериям:
 - Правильность выполнения
 - Время реакции
- Учебные инструменты:
 - Визуализация молекул
 - Интерактивный справочник свойств веществ
- Система мониторинга прогресса:
 - История выполненных работ
 - Сравнение с группой
 - Рекомендации по улучшению

Поддерживаемые устройства и браузеры

Платформа доступна на:

- Компьютерах (Windows 10+, Linux Ubuntu 20.04+)
- Браузерах:
 - Google Chrome
 - Mozilla Firefox
 - Yandex Browser

- Microsoft Edge (Chromium)

Технические требования к ПО и характеристике аппаратуры

- **Обязательно:**
 - Включенный JavaScript
 - Поддержка WebGL 2.0
 - Интернет-соединение (5 Мбит/с)
- **Рекомендуется:**
 - Разрешение экрана 1920*1024
 - Мышь/тачпад с колесом прокрутки

1. НАЧАЛО РАБОТЫ С ИС

1.1. Типовой алгоритм работы

Типовой алгоритм работы:

- Преподаватель зарегистрировался
- Преподаватель оформил подписку
- Преподаватель создал группу
- Преподаватель разослал студентам код подключения к группе
- Преподаватель выдал группе права на выполнение доступных заданий
- Студент присоединился к группе по выданному ему коду подключения
- Студент приступил к выполнению задания
- Студент закончил выполнение задания
- Студент просмотрел результаты выполнения задания
- Преподаватель просмотрел результаты выполнения заданий студентами

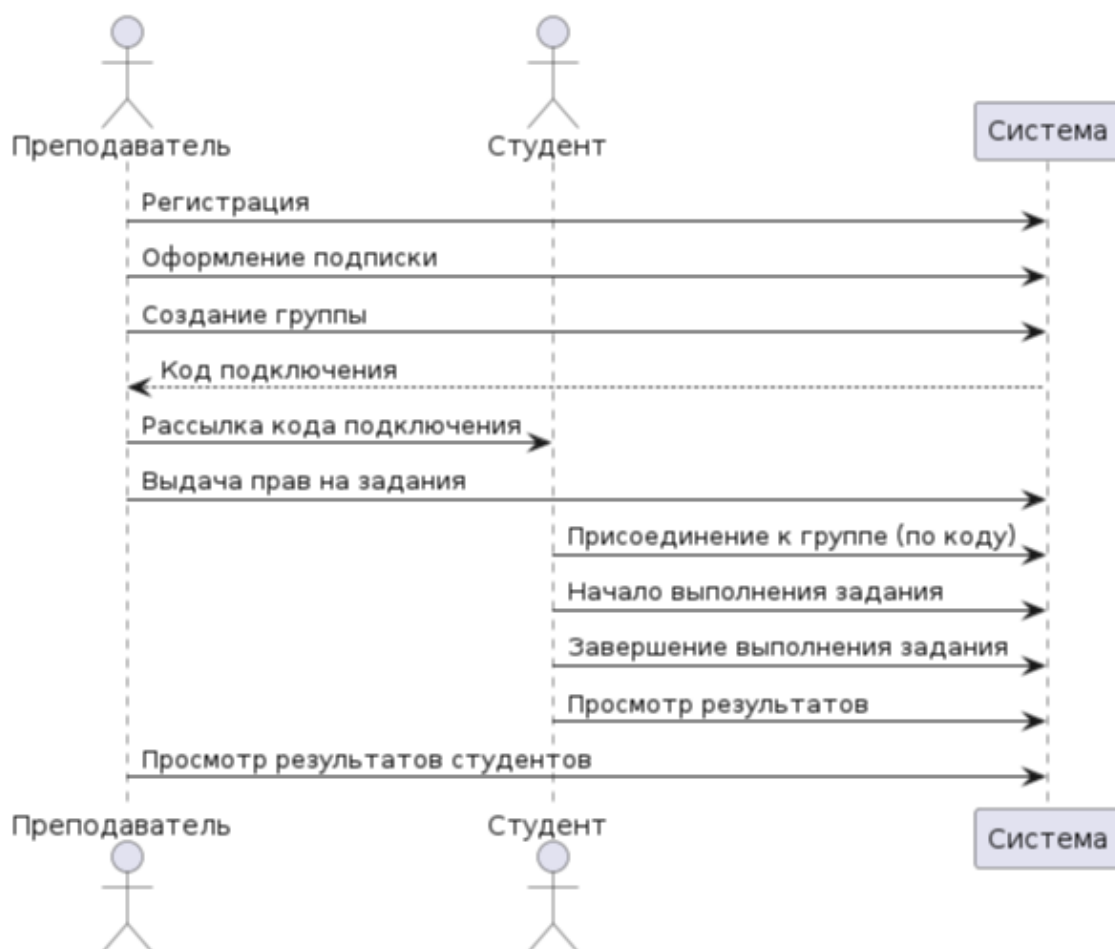


Рисунок 1 – Типовой сценарий использования

1.2.Регистрация и вход в систему

Для начала работы с ИС ВиртЛаб необходимо:

- 1.Нажать кнопку "Зарегистрироваться" на главной странице
- 2.Заполнить форму регистрации и выполнить регистрацию.
- 3.Для входа использовать кнопку "Войти" и ввести свои данные

Обзор интерфейса

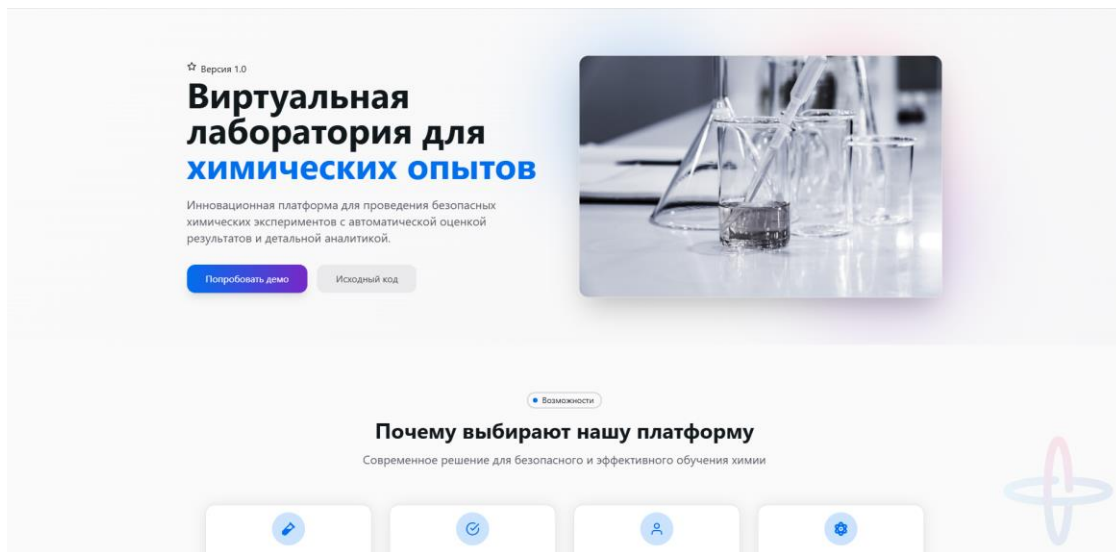


Рисунок 1 – Заголовок главной страницы

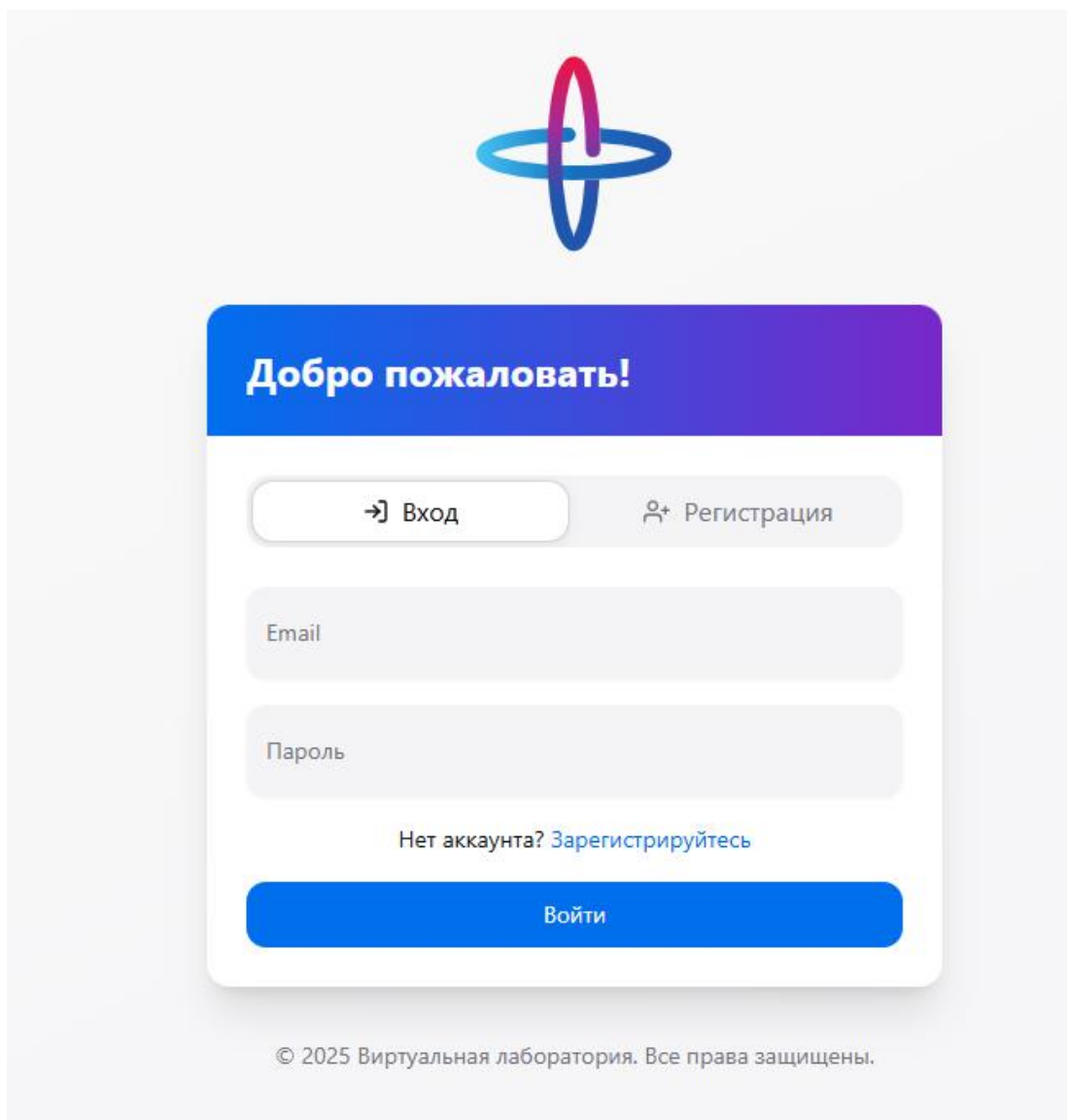


Рисунок 2 – Форма регистрации

Личный кабинет преподавателя предоставляет инструменты для управления учебным процессом, группами студентов и отслеживания активности. Интерфейс разделен на несколько функциональных блоков.

1.3.Описание основных разделов

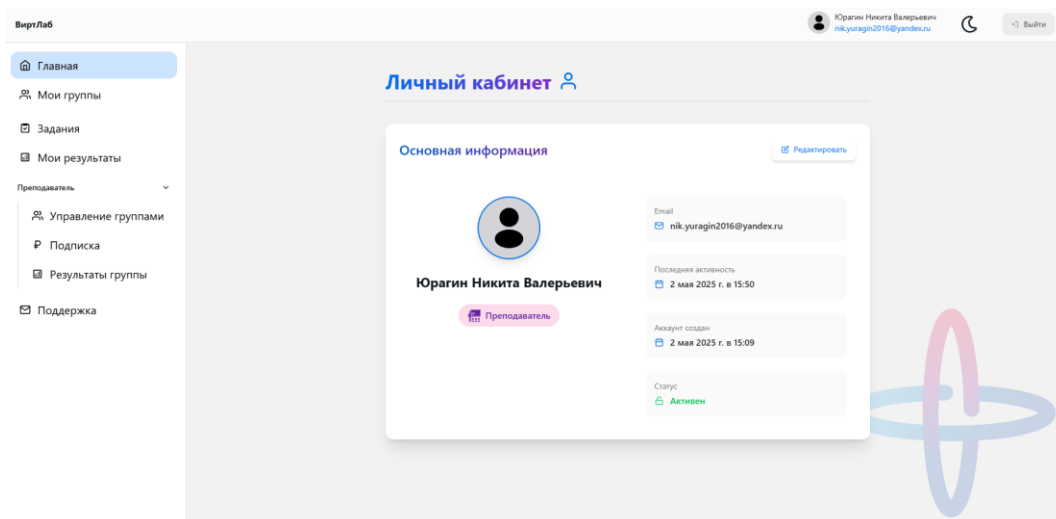


Рисунок 3 – Личный кабинет преподавателя

1. Основная информация о пользователе

- Отображает ФИО преподавателя: Юрагин Никита Валерьевич
- Указывает роль пользователя: Преподаватель
- Содержит кнопку [Редактировать] для изменения личных данных
- Показывает контактный email
- Отображает данные об активности:
 - Последняя активность: 2 мая 2025 г. в 15:50
 - Дата создания аккаунта: 2 мая 2025 г. в 15:09
 - Статус аккаунта: Активен

2. Навигационное меню

Разделено на две основные категории:

Функционал студента

- Главная - переход на основную страницу
- Мои группы - просмотр списка учебных групп студента
- Задания - управление учебными заданиями
- Мои результаты - просмотр персональных результатов

Функционал преподавателя

- Управление группами - инструменты для работы с учебными группами
- Подписка - управление подписками
- Результаты группы - анализ успеваемости студентов
- Поддержка - обращение в службу поддержки

Функциональные возможности ИС

1. Управление учебными группами

- Создание и редактирование групп
- Назначение студентов в группы
- Просмотр состава групп

2. Работа с заданиями

- Создание и публикация учебных заданий
- Проверка выполненных работ
- Выставление оценок

3. Мониторинг успеваемости

- Просмотр результатов студентов
- Анализ успеваемости по группам
- Генерация отчетов

4. Персональные настройки

- Изменение контактных данных
- Просмотр истории активности

1.4.Пример выполнения задания в web-интерфейсе лаборатории

Подробнее стоит остановиться на разделе выполнения задания.

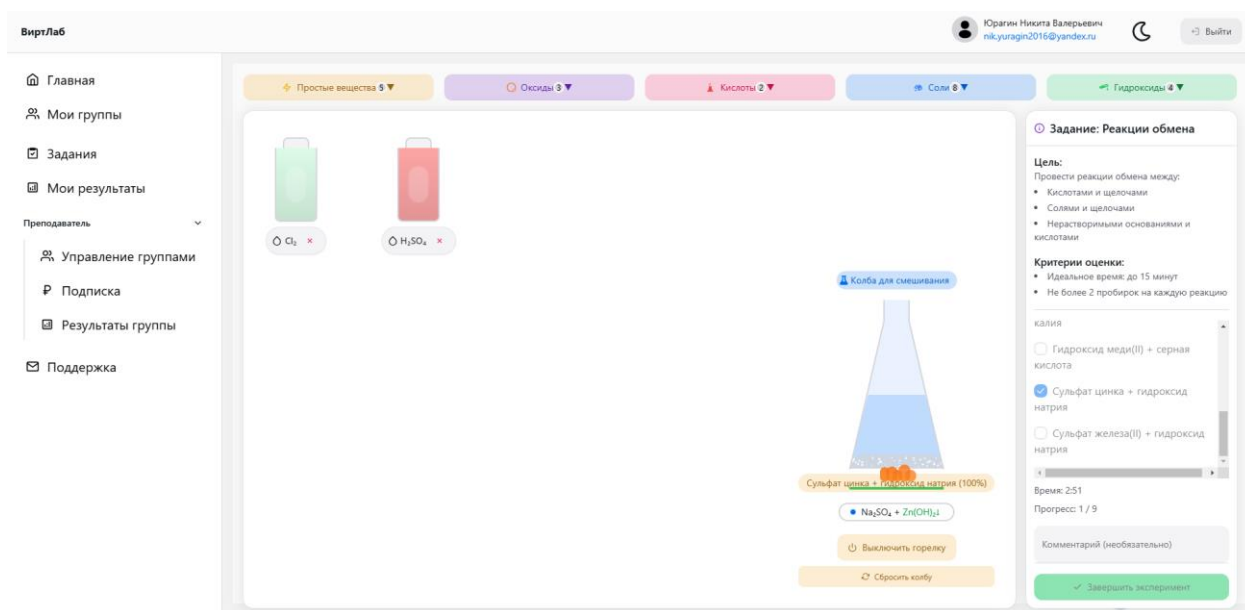


Рисунок 4 – Окно выполнения задания

UX-интерфейс:

- Пять кнопок с реагентами, отсортированными по типу веществ
- Область с описанием задания и кнопкой его завершения
- Область для смешивания веществ

Перемещая реагенты в колбу, пользователь может проводить реакции, в случае успешно проведённой нужной реакции, пункт задания засчитывается выполненным.

2. ФУНКЦИОНАЛ ПОЛЬЗОВАТЕЛЯ С РОЛЬЮ “АДМИНИСТРАТОР”

Точкой входа в функционал администратора является форма авторизации:

Рисунок 5 – Форма входа в админ-панель

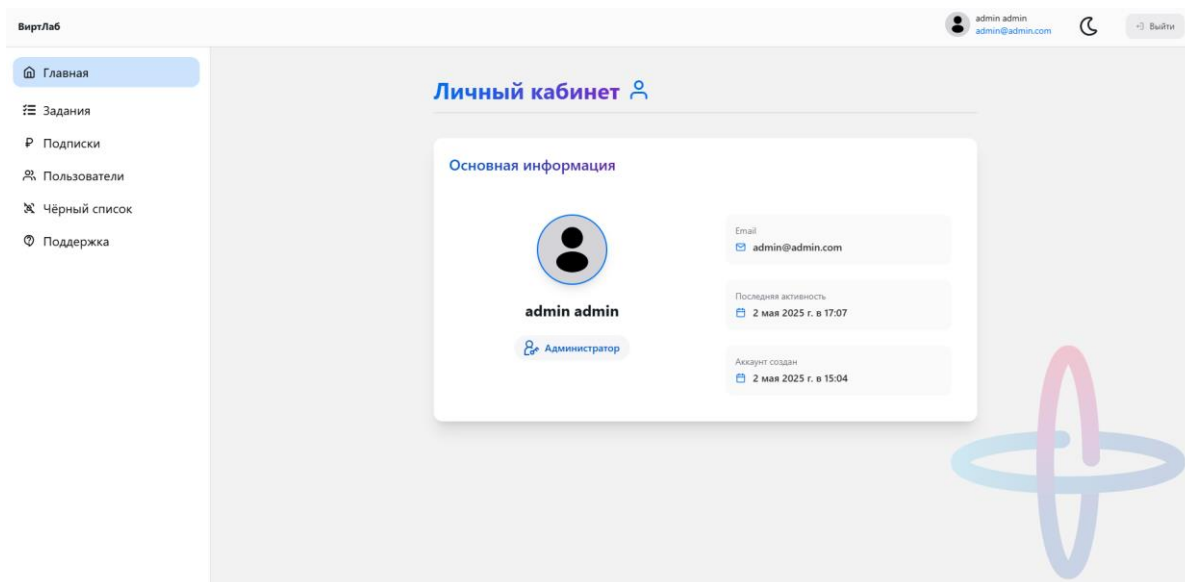


Рисунок 6 – Личный кабинет администратора

Вкладки:

- Главная (Информация об аккаунте администратора)
- Задания (Управление доступности заданий)
- Подписки (Управление подписками пользователей)
- Пользователи (Просмотр, управление и редактирование пользователей)
- Черный список (Просмотр пользователей в чёрном списке и манипулирование ими)
- Поддержка (Просмотр обращений пользователей в поддержку. Отправка ответов на обращения пользователей)

3. ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИИ

Обращение в поддержку

Нажав на кнопку “Поддержка” в боковом меню пользователь может обратиться в поддержку и описать возникшую проблему или пожелания для улучшения работы сервиса.

4. БЕЗОПАСНОСТЬ И АДМИНИСТРИРОВАНИЕ

Управление доступом

- **Роли пользователей:**

- *Преподаватель*: полный доступ к созданию групп, заданий и аналитике.
- *Студент*: доступ только к выполнению заданий и просмотру своих результатов.
- *Администратор*: доступ только к выполнению заданий и просмотру своих результатов.

Рекомендации по безопасности

- Не передавать код подключения к группе третьим лицам.
- Использовать сложные пароли (сочетание букв, цифр и символов).

5. ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ (FAQ)

Вопрос: как добавить студента в группу, если он не получил код?

Ответ: Преподаватель может самостоятельно добавить студента в группу по его email

Вопрос: почему реакция не засчитывается как выполненная?

Ответ: Проверьте

1. Соответствие реагентов условиям задания.
2. Правильность последовательности действий.
3. Наличие ошибок в интерфейсе (красные подсказки).

6. ПРИЛОЖЕНИЕ

Контакты поддержки

- Email: support@virtlab.ru
- Телефон: +7 (938) 504-10-41 (работает с 10:00 до 18:00 по МСК).

ПРИЛОЖЕНИЕ Г (РУКОВОДСТВО РАЗРАБОТЧИКА)

ВВЕДЕНИЕ

Настоящая документация предназначена для разработчиков, занимающихся сопровождением и развитием системы "ИС ВиртЛаб" — платформы виртуальных химических лабораторий для образовательных учреждений. В документе содержится полное описание архитектуры, API и ключевых механизмов работы системы, необходимое для понимания её устройства и внесения изменений в код.

Система реализована как современное веб-приложение с клиент-серверной архитектурой. Клиентская часть разработана на React с использованием TypeScript и компонентной библиотеки HeroUI, что обеспечивает надежность типизации и единообразие интерфейса. Серверная часть построена на Node.js с применением Express.js, а для хранения данных используется PostgreSQL с оптимизированными схемами для химических экспериментов. Выбор технологического стека обусловлен требованиями к производительности при работе с сложными химическими расчетами и необходимостью обеспечения высокой доступности образовательного контента.

Документация соответствует отраслевым стандартам разработки образовательного ПО и включает детальное описание всех компонентов системы. Особое внимание уделено взаимодействию между клиентом и сервером при выполнении экспериментов, работе с базой данных для хранения результатов, а также вопросам безопасности при обработке персональных данных учащихся. Для удобства работы ключевые фрагменты кода, включая примеры химических реакций и алгоритмы автоматической оценки, вынесены в приложения с соответствующей нумерацией.

Материал рассчитан на разработчиков с опытом работы с современными веб-технологиями и базовыми знаниями в области химии. Изучение данной документации позволит быстро освоить логику работы системы, вносить

изменения в функционал лабораторных работ и устранять возможные неисправности. Дополнительные сведения о требованиях к окружению, процессе развертывания и интеграции с системами дистанционного обучения приведены в соответствующих разделах документа

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Назначение и архитектура системы

"ИС ВиртЛаб" представляет собой веб-платформу для проведения виртуальных химических экспериментов, построенную по клиент-серверной модели. Система предназначена для образовательных учреждений и обеспечивает безопасное, доступное и интерактивное изучение химических процессов через браузер.

Архитектурные особенности:

1. Клиентская часть:

- Реализована на React с TypeScript и HeroUI
- Адаптивный интерфейс с поддержкой различных устройств
- Локальное кэширование данных для повторного просмотра экспериментов

2. Серверная часть:

- Построена на Node.js с Express.js
- Микросервисная архитектура с выделенными сервисами:
 - User Service – обработка работы пользователей
 - Admin Service – обработка работы администратора
- RESTful API с JWT-аутентификацией

3. Хранение данных:

- PostgreSQL для структурированных данных (пользователи, группы, задания)

Ключевые характеристики:

- Поддержка 3 базовых химических экспериментов
- Система автоматической оценки с 3 критериями проверки
- Механизмы контроля академической честности

Архитектура системы оптимизирована для:

- Высокой отзывчивости при работе с графикой
- Стабильной работы при одновременном выполнении экспериментов группой студентов

- Быстрой обработки химических расчетов
 - Легкого масштабирования под нагрузку
- Система соответствует требованиям:
- Образовательных стандартов (ФГОС)
 - Защиты персональных данных (152-ФЗ)
 - Доступности (WCAG 2.1 AA)

1.2.Используемые технологии

Серверная часть:

- Node.js (v18 LTS) - основа серверной платформы
- Express.js - фреймворк для REST API с расширенной системой middleware
- PostgreSQL (v15+) - основное хранилище данных с оптимизированными схемами

Клиентская часть:

- React (v18) с TypeScript - компонентный подход
- HeroUI - библиотека UI-компонентов
- Drag-and-drop - анимации манипуляций с объектами реакций
- Axios - HTTP-клиент с интегрированной авторизацией

Безопасность:

- JWT (HS512) + Refresh Tokens - система аутентификации
- bcrypt (cost=12) - хеширование паролей
- RBAC - ролевая модель доступа (студент/преподаватель/админ)

Дополнительно:

- Docker - контейнеризация сервисов

1.3.Структура проекта

Проект организован в соответствии с практиками веб-разработки:

/client – клиентские web-сервисы преподавателя и администратора

Общие файлы (для обоих проектов)

Корневая директория:

- .dockerignore - Игнорируемые файлы при Docker-сборке
- .env - Переменные окружения (API endpoints, настройки)

- .eslintrc.json - Конфигурация линтера JavaScript/TypeScript
- .prettierrc.json - Настройки форматирования кода
- code.code-workspace - Конфиг VSCode для workspace
- Dockerfile - Инструкции для сборки Docker-образа
- index.html - Главный HTML-файл приложения
- package.json / package-lock.json - Зависимости и версии пакетов
- postcss.config.js - Конфигурация PostCSS (обработка стилей)
- tailwind.config.js - Настройки Tailwind CSS
- tsconfig.json / tsconfig.node.json - Конфигурация TypeScript
- vite.config.ts - Настройки сборщика Vite

Директория src/:

- constants.ts - Общие константы приложения
- index.css - Глобальные стили
- main.tsx - Точка входа React-приложения
- vite-env.d.ts - Типы для переменных окружения Vite

Admin-панель (специфичные части)

admin/:

- hooks.ts - Кастомные хуки состояния
- store.ts - Redux/Zustand хранилище
- types.ts - Типы TypeScript
- services/ - API-клиенты:
 - blacklistApi.ts - Работа с черным списком
 - purchasesApi.ts - Управление покупками
 - tasksApi.ts - API задач
 - usersApi.ts - Управление пользователями

components/: (UI-компоненты)

- error-modal/ - Модальное окно ошибок
- header/ - Шапка администратора
- layout/ - Основной лейаут страниц

features/user/:

- authGuard.tsx - Защита авторизованных роутов
- userSlice.tsx - Redux-слайс пользователя
- pages/: (Страницы)
- blacklist/ - Управление черным списком
- users/ - CRUD пользователей
- User-client-app (специфичные части)
- app/services/:
- groupApi.ts - API групп пользователей
- resultsApi.ts - Работа с результатами
- taskApi.ts - API задач (специфичное для пользователей)
- components/: (Уникальные компоненты)
- chemical-dropdown/ - Выпадающий список химикатов
- flask/ / mixing-flask/ - Компоненты виртуальной лаборатории
- grant-rights-modal/ - Выдача прав пользователям
- enum/reactions.ts - Перечисление химических реакций
- features/user/:
- roleGuard.tsx - Проверка ролей пользователя
- register.tsx - Логика регистрации
- logic/: (Бизнес-логика)
- chemicalEngine.ts - Ядро симуляции химических процессов
- chemicalReactions.ts - Формулы реакций
- pages/: (Уникальные страницы)
- makeTask/ - Создание заданий
- virtual-lab-1/2/3/ - Страницы виртуальных лабораторий
- modufy-group/ (опечатка?) - Управление группами
- utils/:
- register-validation.ts - Валидация формы регистрации

/server – серверное приложение, осуществляющее обработку запросов пользователей и работу с БД.

Корневая директория

- .env - Хранит конфиденциальные переменные окружения (базы данных, секретные ключи, настройки)
- db.js - Конфигурация подключения к базе данных (MongoDB/PostgreSQL/MySQL)
- Dockerfile - Инструкции для сборки Docker-образа приложения
- index.js - Главный файл приложения, инициализирует сервер, подключает middleware и маршруты
- logger.js - Настройки логгера (формат записей, ротация логов)
- package.json - Метаданные проекта и зависимости
- package-lock.json - Точные версии зависимостей
- certificate/
- certificate.crt - SSL-сертификат для HTTPS
- private.key - Приватный ключ для SSL
- controllers/

Бизнес-логика приложения (обработчики запросов):

- adminController.js - Управление админ-панелью
- adminUserController.js - CRUD для администраторов
- authController.js - Аутентификация/регистрация
- blacklistController.js - Работа с черным списком
- groupController.js - Управление группами пользователей
- purchaseController.js - Обработка платежей/покупок
- studentController.js - Логика для студентов
- supportController.js - Обработка обращений в поддержку
- taskController.js - Управление заданиями/задачами
- logs/

Журналы работы приложения:

- access.log - Лог HTTP-запросов
- app.log - Основной лог приложения
- error.log - Лог ошибок

middleware/

Промежуточное ПО:

- adminAuthMiddleware.js - Проверка прав администратора
- authMiddleware.js - Проверка аутентификации
- checkRoleMiddleware.js - Проверка ролей пользователя

models/

- models.js - Схемы и модели данных (Mongoose/Sequelize)

routes/

Маршрутизация:

admin/

- adminRouter.js - Роуты админ-панели
- blacklistRouter.js - API для работы с черным списком

user/

- authRouter.js - Роуты аутентификации
- studentRouter.js - API для студентов
- supportRouter.js - Роуты поддержки
- teacherRouter.js - API для преподавателей

utils/

Вспомогательные функции:

- jwtGenerate.js - Генерация и верификация JWT-токенов

Такая организация обеспечивает:

1. Четкое разделение клиентской и серверной логики
2. Простоту навигации по кодовой базе
3. Легкость добавления новых функциональных модулей
4. Удобство совместной разработки

2. УСТАНОВКА И НАСТРОЙКА

2.1. Требования к серверу

Минимальные требования:

- Процессор: 3+ ядра (рекомендуется Intel Xeon или AMD EPYC)
- Оперативная память: 6 ГБ (12 ГБ для production-среды)
- Дисковое пространство: 60 ГБ SSD (с учетом хранения результатов экспериментов)
- ОС: Ubuntu 22.04 LTS / CentOS 8+ (с поддержкой Docker)
- Сетевые порты:
 - HTTP/HTTPS (80/443)
 - PostgreSQL (5432)

2.2. Запуск серверной части

Установите Git, Docker и docker-compose, клонируйте репозиторий и запустите сборку докер-контейнеров из корневой директории с файлом docker-compose.

Сервер по умолчанию запускается на порту 5000 – для пользователя и на 5001 – для администратора.

2.3. Настройка переменных окружения

Создайте файл .env в корне проекта с обязательными параметрами:

Backend

HTTPS_PORT_USER=5000

HTTPS_PORT_ADMIN=5001

HTTP_PORT_USER=5000

HTTP_PORT_ADMIN=5001

Database

DB_NAME=lab_db

DB_PORT=5432

DB_HOST=db

DB_PASSWORD=qwerty

DB_USER=postgres

App

SECRET_KEY=SECRET_KEY

ADMIN_EMAIL=admin@admin.com

ADMIN_PASSWORD=\$2a\$05\$7OA1ASUAV/URxrKNq3ofxeJlcJl7OWgx/
m.ZVdkU9ANjG3C37KZ36

NODE_ENV=production

После настройки проверьте работоспособность API, выполнив тестовые запросы к эндпоинтам. Убедитесь, что миграции базы данных применены корректно и все таблицы созданы.

3. ОПИСАНИЕ API И ОСНОВНЫХ ФУНКЦИЙ СИСТЕМЫ

3.1. Базовая структура API

Все API-эндпоинты используют префиксы маршрутизации, определенные в server.js. Сервер ожидает данные в формате JSON для POST/PUT запросов и возвращает ответы в том же формате. Для авторизованных запросов требуется установка cookie сессии (кроме /auth).

API-эндпоинты администратора:

1. Авторизация и аутентификация

POST https://localhost:5001/auth/login

- Аутентификация администратора

- Принимает: {email, password}
- Возвращает: JWT-токен и базовую информацию о пользователе

GET http://localhost:5001/api/auth/getInfoAboutSelf

- Получение информации о текущем авторизованном администраторе
- Требуется авторизация

GET https://localhost:5001/auth/check

- Проверка валидности токена
- Используется для проверки сессии

2. Управление черным списком

POST https://localhost:5001/blacklist/addToBlacklist

- Добавление пользователя в черный список
- Параметры: {id_user, reason}

POST https://localhost:5001/blacklist/removeFromBlacklist

- Удаление пользователя из черного списка
- Параметр: {id_user}

GET https://localhost:5001/blacklist/getAll

- Получение полного списка заблокированных пользователей
- Возвращает массив записей с причинами блокировки

3. Работа с обращениями в поддержку

GET <https://localhost:5001/support/getAllAppeal>

- Получение всех обращений в техподдержку
- Возвращает список с статусами и текстами обращений

POST <https://localhost:5001/support/sendResponseToSupport>

- Отправка ответа на обращение
- Параметры: {id_support, admin_response, status}

4. Управление пользователями

GET <https://localhost:5001/api/user/getAllStudents>

- Получение списка всех студентов
- Возвращает массив с ФИО, email и статусом аккаунта

GET <https://localhost:5001/user/getAllTeachers>

- Получение списка всех преподавателей
- Включает информацию о привязанных группах

POST <https://localhost:5001/user/editUserByID>

- Редактирование данных пользователя
- Параметры: {id_user, email, lastname, firstname, middlename, role_name}

5. Управление заданиями

GET <https://localhost:5001/task/getAll>

- Получение всех лабораторных заданий
- Возвращает список с названиями и описаниями

POST <https://localhost:5001/task/add>

- Добавление нового задания
- Параметры: {title, description, is_available}

POST <https://localhost:5001/task/changeAvailableById>

- Изменение доступности задания

- Параметры: {id_task, is_available}

6. Управление платежами

GET https://localhost:5001/purchase/getAll

- Получение всех платежных транзакций
- Возвращает историю оплат с детализацией

POST https://localhost:5001/purchase/changelsPaidById

- Изменение статуса оплаты
- Параметры: {id_purchases, is_paid}

POST https://localhost:5001/purchase/changelsBlockedById

- Блокировка/разблокировка платежа
- Параметры: {id_purchases, id_blocked}

API-эндпоинты для пользователей (кроме администратора):

Аутентификация и профиль:

- POST /auth/registration - Регистрация нового пользователя
 - Параметры: {email, password, lastname, firstname, middlename, role}
 - Возвращает: JWT токен
- POST /auth/login - Вход в систему
 - Параметры: {email, password}
 - Возвращает: JWT токен
- POST /auth/editSelfFromToken - Редактирование своего профиля
 - Параметры: {email, lastname, firstname, middlename}
- GET /api/auth/getInfoAboutSelf - Получение информации о текущем пользователе
- GET /auth/check - Проверка валидности токена

Техническая поддержка:

- POST /support/sendToSupport - Отправка обращения в поддержку
 - Параметры: {message_text}
- GET /support/getListMyAppeal - Получение списка своих обращений

Эндпоинты преподавателя

Управление группами:

- POST /teacher/group/create - Создание новой учебной группы
 - Параметры: {group_name, subject}
- GET /teacher/group/getAllMyGroups - Получение списка своих групп
- POST /teacher/group/addUserToGroup - Добавление студента в группу
 - Параметры: {group_id, user_email}
- POST /teacher/group/removeFromGroupByEmail - Удаление студента из группы
 - Параметры: {group_id, user_email}
- POST /teacher/group/deleteById - Удаление группы
 - Параметры: {group_id}
- POST /teacher/group/grantRightsToGroup - Предоставление прав на группу другому преподавателю
 - Параметры: {group_id, teacher_email}
- POST /teacher/group/changeIsOpenById - Изменение статуса доступности группы
 - Параметры: {group_id, is_open}
- GET /teacher/group/getAllMyAccess - Получение списка групп с доступом
- GET /api/teacher/group/getGroupAttempts - Получение попыток выполнения заданий группой
 - Параметры: {group_id}

Управление заданиями:

- GET /teacher/tasks/getAllAvailable - Получение списка доступных заданий

Платежи и подписки:

- POST /teacher/purchase/add - Создание платежа за доступ к заданиям
 - Параметры: {task_id, group_id, price}
- POST /teacher/purchase/confirm - Подтверждение оплаты

- Параметры: {purchase_id}
- GET /teacher/purchase/getAllMy - Получение своих платежей

Эндпоинты студента

Работа с группами:

- POST /student/group/addSelfToGroup - Вступление в группу
 - Параметры: {group_hash}
- POST /student/group/removeSelfFromGroup - Выход из группы
 - Параметры: {group_id}
- GET /api/student/group/getGroupsWhereiAmMember - Получение списка своих групп

Работа с заданиями:

- GET /student/task/getUserGroupsWithTasks - Получение групп с доступными заданиями
- GET /student/task/getTasksWithGroups - Получение заданий по группам
- GET /student/task/getUserTaskAttempts - Получение своих попыток выполнения заданий
- POST /student/task/createUserTaskAttempt - Начало выполнения задания
 - Параметры: {task_id}
- POST /student/task/finishUserTaskAttempt - Завершение выполнения задания
 - Параметры: {attempt_id, results_data}
- GET /api/student/task/getSelfAttempts - Получение истории своих попыток

3.2. Описание основных функций системы

Система ИС ВиртЛаб реализует комплексный подход к проведению виртуальных химических экспериментов с элементами автоматизированного контроля и оценки. Ключевые функциональные возможности включают:

1. Управление лабораторными работами:

- Преподаватели могут создавать, редактировать и настраивать лабораторные задания, устанавливая параметры:
 - Цель эксперимента
 - Перечень необходимых реактивов и оборудования
 - Последовательность выполнения этапов
 - Критерии оценки (точность, время выполнения, безопасность)

2. Проведение виртуальных экспериментов:

- Студенты получают доступ к интерактивной лаборатории с возможностью:
 - Выбора химических веществ из виртуальной базы (кислоты, щелочи, соли, оксиды)
 - Проведения реакций в интерактивных колбах и пробирках
 - Наблюдения за ходом реакции (изменение цвета, выделение газа, образование осадка)

3. Система автоматической оценки:

- Анализ выполнения работы по параметрам:
 - Правильность последовательности действий
 - Время выполнения задания
- Формирование итоговой оценки с комментариями по ошибкам

4. Игровые элементы для повышения мотивации:

- Рейтинги:
 - Личные показатели прогресса
 - Сравнение результатов с другими студентами группы

5. Администрирование системы:

- Управление пользователями (студенты, преподаватели)
- Настройка прав доступа к лабораторным работам

4. РАБОТА С БАЗОЙ ДАННЫХ

4.1. Схема данных и связи между таблицами

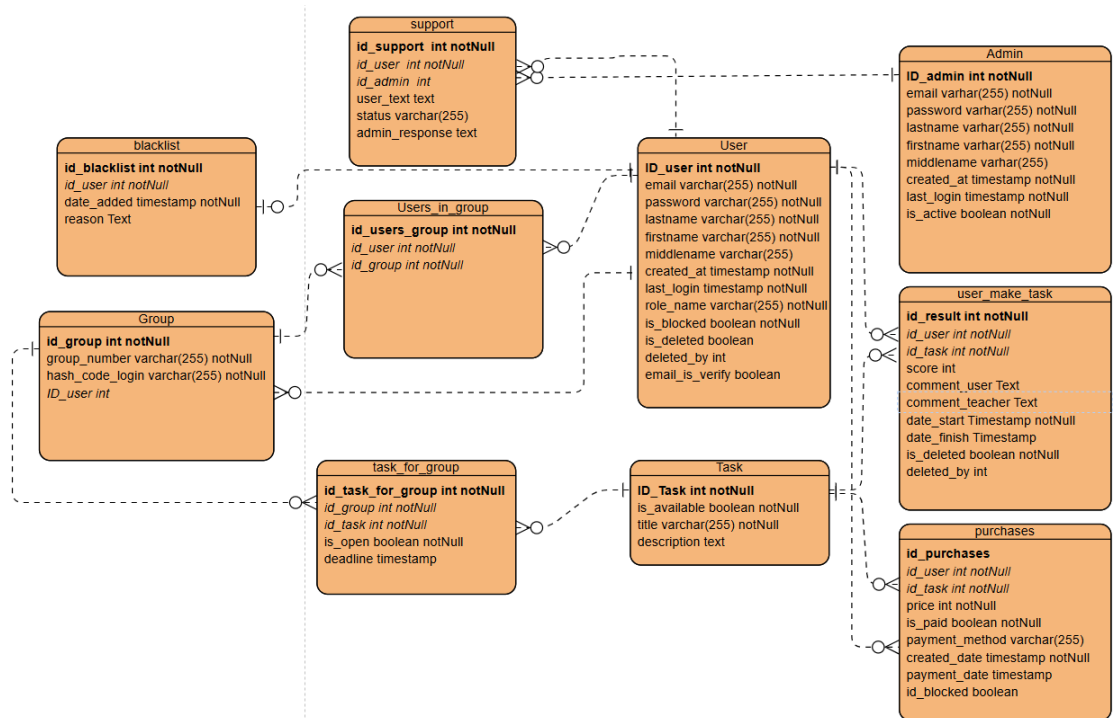


Рисунок 1 – Схема таблиц и связей в БД

Основные таблицы базы данных включают:

Пользователи (User)

- Основные учетные данные (email, пароль, ФИО)
- Ролевая модель (role_name: student/teacher/admin)
- Статус аккаунта (is_blocked, email_is_verify)
- Активность (created_at, last_login)

Группы (Group)

- Учебные группы (group_number)
- Уникальный хэш для вступления (hash_code_login)
- Привязка к преподавателю (ID_user)
- Пользователи в группах (Users_in_group)

Задания (Task)

- Базовые параметры (title, description)
- Доступность (is_available)

Задания для групп (Task_for_group)

- Параметры доступа (is_open, deadline)

Результаты (User_make_task)

- Оценка (score)
- Комментарии (comment_user, comment_teacher)
- Временные метки (date_start, date_finish)
- Контроль удаления (is_deleted)

Поддержка (Support)

- Обращения пользователей (user_text)
- Ответы администраторов (admin_response)
- Статус обработки (status)

Черный список (Blacklist)

- Заблокированные пользователи (id_user)
- Причина блокировки (reason)
- Дата добавления (date_added)

Платежи (Purchases)

- Доступ к платным заданиям (id_task)
- Статус оплаты (is_paid)
- Метод оплаты (payment_method)
- Финансовые метрики (price)

Администраторы (Admin)

- Отдельная таблица для админсостава

Контроль активности (is_active)

5. РАЗРАБОТКА И РАСШИРЕНИЕ ФУНКЦИОНАЛА

5.1.Добавление новых заданий

Для расширения системы новыми заданиями и экспериментами требуется вносить изменения в файл, в котором описываются все возможные реакции между реагентами, добавление новых факторов, влияющих на проведение реакции (катализаторы, ингибиторы, электролиз и т.д.)

5.2.Интеграция с внешними сервисами

Подключение дополнительных сервисов начинается с разработки модуля аутентификации, который будет обрабатывать запросы авторизации со сторонних сервисов и предоставлять в ответ информацию о результатах выполнения различных заданий студентами.

6. ТЕСТИРОВАНИЕ И ОТЛАДКА

6.1.Проверка API

Интеграционное тестирование API выполняется с помощью Postman для проверки взаимодействия между компонентами системы. Создана коллекция тестовых запросов, охватывающая все ключевые сценарии работы: от регистрации пользователя до сложных манипуляций со структурами заданий и групп.

6.2..Логирование ошибок

Система оснащена системой логирования ошибок. Логируется подключение к БД в начале работы, доступные api-эндпоинты, а также все обрабатываемые и поступающие на сервер http запросы.

7. БЕЗОПАСНОСТЬ

7.1. Аутентификация и авторизация

Система использует комбинированный подход к проверке подлинности пользователей. Для веб-интерфейса применяются сессии с защищенными LocalStorage, где идентификатор сессии хранится в зашифрованном виде. API-запросы аутентифицируются через JWT-токены и подписанные с помощью HMAC-SHA256. Авторизация реализована на основе ролей (пользователь, администратор) с проверкой прав доступа на уровне middleware.

7.2. Защита от SQL-инъекций

Взаимодействие с базой данных построено на принципах параметризованных запросов, где все пользовательские данные передаются отдельно от SQL-кода. Для работы с PostgreSQL используется библиотека sequelize, которая автоматически генерирует запросы по введенным параметрам.

Хранение паролей

Пароли пользователей никогда не хранятся в открытом виде - вместо этого система использует адаптивное хеширование через алгоритм bcrypt с cost factor 8. При каждой операции аутентификации происходит сравнение хешей с добавлением уникальной "соли" для каждого пользователя. В случае компрометации базы данных этот подход делает невозможным восстановление исходных паролей. Для дополнительной защиты реализована политика сложности паролей, требующая минимальной длины 8 символов с обязательным сочетанием разных типов символов. При смене пароля система автоматически инвалидирует все активные сессии пользователя, вынуждая повторную аутентификацию на всех устройствах.

8. РАЗВЕРТЫВАНИЕ В PRODUCTION

8.1.Настройка Nginx/Apache для Node.js

Установите Git, Docker и docker-compose, клонируйте репозиторий и запустите сборку докер-контейнеров из корневой директории с файлом docker-compose.

Сервер по умолчанию запускается на порту 5000 – для пользователя и на 5001 – для администратора.

Клиентская часть запускается на портах 5173 (пользовательская) и 5174 (администраторская).

Nginx прослушивает порты: 80, 443, 8080 и проксирует запросы на серверные порты.