

УДК 004.9

**ИНФОРМАЦИОННАЯ СИСТЕМА ПОДДЕРЖКИ ИНДИВИДУАЛЬНОГО ОБУЧЕНИЯ
«SCHOOLDOT»****Верещагин Елисей Сергеевич¹, Тимохин Никита Александрович²,
Бархатова Ирина Александровна³, Роенко Анна Олеговна⁴**¹Студент;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;

e-mail: ves.21@uni-dubna.ru.

²Студент;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;

e-mail: tna.21@uni-dubna.ru.

³Старший преподаватель;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;

e-mail: baria@uni-dubna.ru.

⁴Старший преподаватель;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;

e-mail: roenko.a.o@uni-dubna.ru.

В данной работе представлено проектирование и реализация информационной системы SchoolDot, предназначенной для автоматизации процессов деятельности частного преподавателя. Система охватывает ключевые аспекты образовательного взаимодействия: проведение онлайн-занятий, выдачу и проверку заданий, управление расписанием и оплатой, а также отслеживание прогресса учащихся. Разработка веб-приложения включает анализ требований, проектирование архитектуры, реализацию серверной и клиентской частей, проведение тестирования.

Ключевые слова: информационная система, управление, система обучения с репетитором.

Для цитирования:

Информационная система поддержки индивидуального обучения «SchoolDot» / Е. С. Верещагин, Н. А. Тимохин, И. А. Бархатова, А. О. Роенко // Системный анализ в науке и образовании. 2026. № 2. С. 93–104. EDN: QFICMQ. URL: <https://sanse.ru/index.php/sanse/article/view/719>.

INFORMATION SYSTEM FOR SUPPORT INDIVIDUAL LEARNING "SCHOOLDOT"**Vereshchagin Elisey S. ¹, Timokhin Nikita A. ²,
Barkhatova Irina A. ³, Roenko Anna O. ⁴**¹Student;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: ves.21@uni-dubna.ru.

² Student;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: tna.21@uni-dubna.ru.



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

³Senior teacher;
Dubna State University;
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;
e-mail: baria@uni-dubna.ru.

⁴Senior teacher;
Dubna State University;
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;
e-mail: roenko.a.o@uni-dubna.ru.

This paper presents the design and implementation of the SchoolDot information system intended to automate the processes of private teaching. The system encompasses the key aspects of educational interaction, namely conducting online classes, assigning and assessing tasks, schedule and payment management, and tracking student progress. The web application development process includes requirements analysis, architecture design, server-side and client-side implementation, and testing.

Keywords: information system, management, tutoring system.

For citation:

Vereshchagin E. S., Timokhin N. A., Barkhatova I. A., Roenko A. O. Information system for support individual learning “SchoolDot”. *System analysis in science and education*, 2026;(2):93–102 (in Russ). EDN: QFICMQ. Available from: <https://sanse.ru/index.php/sanse/article/view/719>.

Введение

В современном образовательном процессе деятельность репетитора сохраняет высокую актуальность, особенно в условиях растущего спроса на индивидуальные формы обучения и развитие дистанционного формата занятий. При этом цифровизация частного преподавания зачастую сводится к использованию разрозненных инструментов – мессенджеров, облачных хранилищ, таблиц, сервисов видеосвязи и календарей. Это создает дополнительную нагрузку на преподавателя, снижает управляемость процессами и усложняет контроль прогресса учащихся.

С целью решения обозначенной проблемы была разработана информационная система *SchoolDot* – веб-приложение, обеспечивающее комплексную поддержку репетиторской деятельности. Платформа объединяет в себе функциональность для проведения онлайн-занятий, выдачи и оценки заданий, мониторинга прогресса учеников, управления расписанием и приема оплаты.

Система ориентирована на индивидуальное взаимодействие «репетитор–ученик» и нацелена на сокращение временных затрат за счёт автоматизации ключевых рутинных процессов. Основными задачами проекта стали: формализация требований и проектирование системы (с точки зрения системного анализа), реализация серверной и клиентской части, подготовка и отладка функциональных модулей.

Широкий обзор аналогичных решений показал отсутствие отечественных систем, способных в полной мере закрыть потребности частных преподавателей. В отличие от корпоративных LMS или платформ для поиска репетиторов, разрабатываемая система ориентирована на практическое ведение занятий и контроль учебного процесса в рамках одного цифрового пространства. Применение системного анализа в науке и образовании непосредственно связано и с текущим проектом: подобное знание существенно повлияло на аналитику перед реализацией продукта.

Проект выполнен в рамках совместной выпускной квалификационной работы двух студентов направления 09.03.04 «Программная инженерия»: Верещагина Елисея (аналитик проекта) и Тимохина Никиты (разработчик).

1. Анализ существующих решений

На текущем этапе развития *EdTech*-индустрии на рынке присутствует ряд цифровых решений, предназначенных для поддержки образовательной деятельности. Однако анализ показал, что ни одно

из них не удовлетворяет в полной мере требованиям индивидуальной репетиторской практики, предполагающей персонализированный подход, высокую гибкость интерфейсов и минимизацию временных затрат на административные задачи.

Среди рассмотренных систем можно выделить следующие категории:

1. Системы дистанционного обучения университетского уровня (LMS):

- *Moodle, LmsDot* – обладают широким функционалом, ориентированным на массовое обучение студентов [1]. Включают модули заданий, расписания, групп и курсов, однако сложны в освоении, не рассчитаны на индивидуальный формат и не позволяют кастомизировать рабочую область под частные задачи,
- основные недостатки: перегруженность интерфейсов, слабая адаптация под частных преподавателей, отсутствие встроенных инструментов оплаты.

2. Платформы пошагового обучения:

- *Stepik, Foxford, Умскул, SkySmart* – предоставляют доступ к авторским курсам и структурам уроков [2], однако рассчитаны на широкий рынок и не позволяют гибко управлять учебным процессом или взаимодействием с конкретным учеником,
- основные недостатки: ограниченность в настройке контента, закрытые методики, невозможность интеграции с внешними инструментами.

3. Системы планирования занятий и взаимодействия с учениками:

- *Finch* – специализированная ИС для репетиторов, предоставляющая средства учета занятий, расписания и материалов [3]. Однако платформа не предусматривает проведение занятий внутри интерфейса, не поддерживает автоматизированную проверку заданий и не предоставляет модули оплаты,
- преимущества: интуитивно понятный интерфейс, простота планирования,
- недостатки: отсутствие инструментов взаимодействия, оценки и онлайн-уроков.

4. Платформы подбора репетиторов:

- *Профи.ру, Ваш репетитор, Ассоциация репетиторов, Avito, Preply, Italki* – ориентированы на поиск преподавателя и организацию первичного контакта, но не содержат инструментов для ведения образовательной деятельности и учета прогресса.

Результаты анализа представлены в таблице 1.

Табл. 1. Сравнительный анализ

Система	Основное назначение	Преимущества	Недостатки
<i>LmsDot, Moodle</i>	Обучение в вузах [1]	Модули заданий, расписание, группы	Нестабильность, перегруженность, отсутствие персонализации интерфейса
<i>Stepik, Foxford</i> и др.	Платформы пошагового обучения [2]	Готовые курсы, структурированность	Нет гибкости, закрытые методики, платная модель
<i>Finch</i>	Учёт и планирование занятий [3]	Удобный интерфейс, статистика	Нет уроков, заданий, системы оценивания и оплаты
<i>Профи.ру, Preply</i> и др.	Поиск преподавателей	Широкая база, рейтинги	Нет образовательной среды, ограниченный функционал

Как видно из анализа, существующие решения либо слишком обобщённые, либо функционально ограничены. Это обусловило необходимость создания новой системы, способной удовлетворить весь спектр задач частного преподавания — от расписания и заданий до аналитики и монетизации.

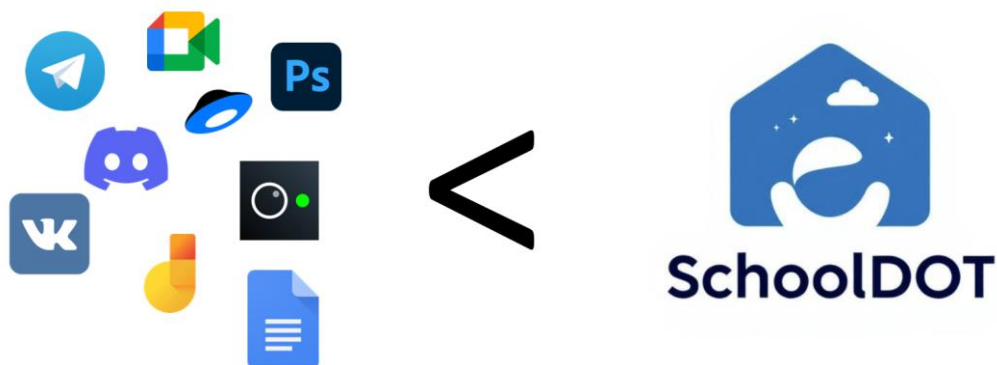


Рис. 1. Сравнение набора сервисов и платформы SchoolDOT

2. Описание информационной системы с точки зрения пользователей

2.1 Роли пользователей и их задачи

SchoolDot разработана для двух основных категорий пользователей: репетиторов и учеников. Каждая роль имеет свой набор функций и сценариев взаимодействия с системой.

Репетиторы используют систему для:

- управления профилем с указанием предметов;
- настройки расписания для занятий;
- проведения онлайн-уроков через встроенный видеочат;
- создания и назначения учебных заданий ученикам;
- проверки выполненных работ и выставления оценок;
- отслеживания прогресса каждого ученика через аналитические дашборды;
- управления финансами: просмотр истории платежей и выручки.

Ученики используют систему для:

- просмотра расписания занятий;
- бронирования удобных временных слотов для занятий;
- участия в онлайн-уроках;
- получения заданий и их выполнения в системе;
- просмотра своей успеваемости и динамики прогресса;
- оплаты занятий через интегрированную платежную систему.

Гости используют систему для последующей регистрации и получения роли родителя или ученика.

Родители используют систему для:

- мониторинга успеваемости ученика;
- оплаты занятий через интегрированную платежную систему (функция доступа как со стороны ученика, так и со стороны родителя).

Администраторы используют систему для:

- добавления, удаления, настройки учетных записей пользователей
- мониторинга состояния системы.

2.2 Основные функциональные возможности

SchoolDot включает следующие ключевые модули, обеспечивающие полный цикл образовательного взаимодействия:

Модуль управления пользователями позволяет создавать и редактировать личную информацию. Репетиторы указывают свою специализацию, образование, опыт работы и стоимость занятий. Ученики заполняют данные о целях обучения и уровне знаний. Система поддерживает загрузку фотографий, документов об образовании и портфолио.

Модуль расписания — центральный элемент организации учебного процесса. Репетитор отмечает в календаре дату и время занятия, указывая длительность (30, 60, 90 минут). Ученик видит расписание в удобном календарном виде. Система автоматически отправляет уведомления о предстоящих занятиях обеим сторонам за 24 часа и за 1 час до начала урока.

Модуль проведения уроков обеспечивает возможность решать учебные задачи и коммуницировать в одном пространстве в рамках одного занятия. Во время урока доступны инструменты:

- виртуальная доска для рисования и записей;
- чат для текстового общения;
- совместный просмотр документов и презентаций;
- функционал отправки и решения задач;
- оценивание прикрепленных к занятию задач.

Модуль учебных материалов и заданий позволяет репетитору создавать задания различных типов: текстовые ответы, выбор варианта, загрузка файлов, тестирование. Для каждого задания устанавливается предмет и тема для их классификации. Ученик получает возможность просмотра банка задач в виде списка.

Модуль аналитики и прогресса предоставляет визуализацию учебных достижений. Репетитор видит:

- динамику оценок каждого ученика по времени;
- процент выполненных заданий в срок;
- темы, вызывающие наибольшие затруднения;
- общую статистику посещаемости занятий.

Ученик отслеживает:

- свою успеваемость в сравнении с предыдущими периодами;
- количество пройденных тем и выполненных заданий;
- рекомендации репетитора по улучшению результатов.

Модуль платежей и финансов интегрирован с платежными системами для автоматической обработки транзакций. Оплата может происходить по разным схемам:

- разовая оплата за одно занятие;
- покупка пакета занятий со скидкой;
- подписка на регулярные еженедельные уроки.

После завершения урока система автоматически списывает средства со счета ученика и переводит их репетитору (за вычетом комиссии платформы, если применимо). Пользователи имеют доступ к истории всех транзакций.

2.3 Сценарии взаимодействия пользователей с системой

Сценарий 1: Регистрация пользователя в системе и связь репетитор-ученик

1. Ученик регистрируется в системе, указывая свои данные.
2. Репетитор регистрируется в системе, указывая свои данные.
3. Репетитор находит ученика в списке пользователей и добавляет к своему профилю.
4. Ученик получает уведомление о добавлении к репетитору.
5. Репетитор настраивает расписание занятий в системе.
6. У ученика отображается расписание занятий и доступные уроки.

Сценарий 2: Проведение типичного урока

1. За час до занятия репетитор и ученик получают напоминание с кнопкой «Войти в урок».
2. В назначенное время обе стороны входят в виртуальную комнату урока.
3. Репетитор объясняет новую тему, используя виртуальную доску для записей и графиков.
4. Ученик задает вопросы через аудио или текстовый чат.
5. Репетитор демонстрирует презентацию с примерами задач через функцию «Поделиться экраном».
6. В конце урока репетитор назначает домашнее задание прямо в системе, устанавливая срок выполнения.

Сценарий 3: Выполнение и проверка домашнего задания

1. Ученик получает уведомление о новом задании: «Решить 10 задач по теме "Квадратные уравнения"».
2. Открывает задание в личном кабинете, видит условия задач и форму для ответов.
3. Решает задачи, вводит ответы в соответствующие поля и загружает фото с подробным решением.
4. Нажимает «Отправить на проверку». Статус задания меняется на «На проверке».
5. Репетитор получает уведомление о выполненном задании.
6. Проверяет ответы, оставляет комментарии к ошибкам: «В задаче 3 забыл учесть второй корень».
7. Выставляет оценку «4/5» и публикует результат.
8. Ученик получает уведомление с оценкой и комментариями, может задать уточняющие вопросы.

Сценарий 4: Анализ прогресса

1. Репетитор открывает раздел «Аналитика» и выбирает конкретного ученика.
2. Система отображает график изменения оценок за последние три месяца: замечен рост с 3.2 до 4.5 баллов.
3. В разделе «Проблемные темы» видит, что ученик допускает ошибки в задачах на иррациональные уравнения.
4. Принимает решение посвятить следующий урок повторению этой темы.
5. Создает дополнительное задание с акцентом на слабые места ученика.

Эти сценарии демонстрируют, как *SchoolDot* упрощает все этапы репетиторской деятельности: от первого контакта до системной работы над улучшением знаний.



Рис. 2. Контекстная диаграмма ролей SchoolDOT

3. Архитектура системы

Разрабатываемая информационная система [4] *SchoolDot* реализована в виде веб-приложения с клиент-серверной архитектурой [5].

Система включает три основных уровня:

1. клиентский уровень: реализован на *JavaScript* с использованием библиотеки *React* [6]. Отвечает за отображение интерфейса, обработку пользовательского ввода, маршрутизацию и взаимодействие с сервером через *REST API*,
2. серверный уровень: реализован на языке *Python* с использованием фреймворка *FastAPI*. Обеспечивает маршрутизацию запросов, бизнес-логику, аутентификацию, взаимодействие с базой данных и генерацию спецификации *API*,
3. уровень хранения данных: реализован с использованием реляционной СУБД *PostgreSQL* [7]. Структура базы данных спроектирована в соответствии с нормальными формами и включает связи между сущностями (пользователи, курсы, задания, занятия, оплаты и др.).

Коммуникация между слоями осуществляется по протоколу *HTTPS* с передачей данных в формате *JSON*. Авторизация реализована через *JWT*-токены, что обеспечивает безопасность сессий пользователей и разграничение доступа к функционалу [8].

4. Проектирование интерфейса

Проектирование пользовательского интерфейса (*UI*) системы *SchoolDot* осуществлялось с учётом задач, выполняемых представителями разных ролей: учеником, репетитором и администратором. Основная цель заключалась в обеспечении простоты, интуитивной понятности и согласованности визуальных компонентов при одновременном сохранении функциональной полноты. Разработка интерфейсов велась в графическом редакторе *Figma*.

Для каждого экрана в *Figma* были разработаны интерактивные прототипы, отражающие как состояние интерфейса до выполнения действия, так и после (например, до и после отправки ответа на задание). Это позволило заранее выявить потенциальные точки затруднения пользователя и устранить их на этапе проектирования.

В качестве общего принципа проектирования использовались:

- консистентность расположения элементов;
- единая палитра и стиль иконок;
- контекстная доступность функций (например, кнопки «Оценить» доступны только для репетитора при наличии ответа);
- минимизация количества кликов до ключевого действия.

Таким образом, проектирование интерфейса обеспечило соответствие функциональности системы ожиданиям пользователей, снизило когнитивную нагрузку и повысило удобство в работе.

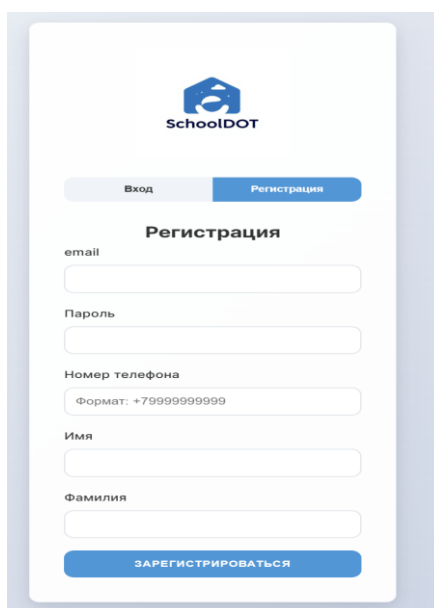


Рис. 3. Интерфейс регистрации

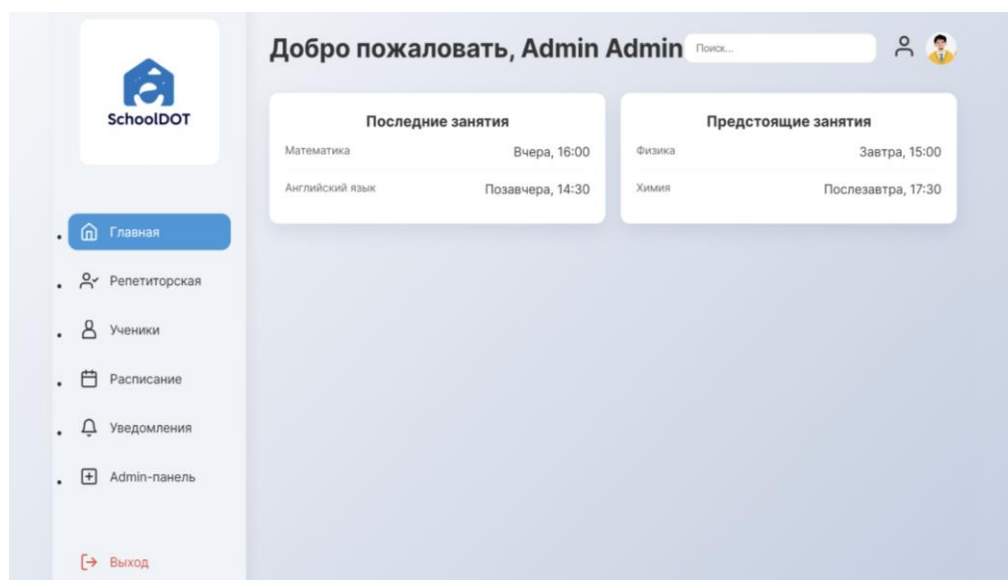


Рис. 4. Интерфейс главной страницы

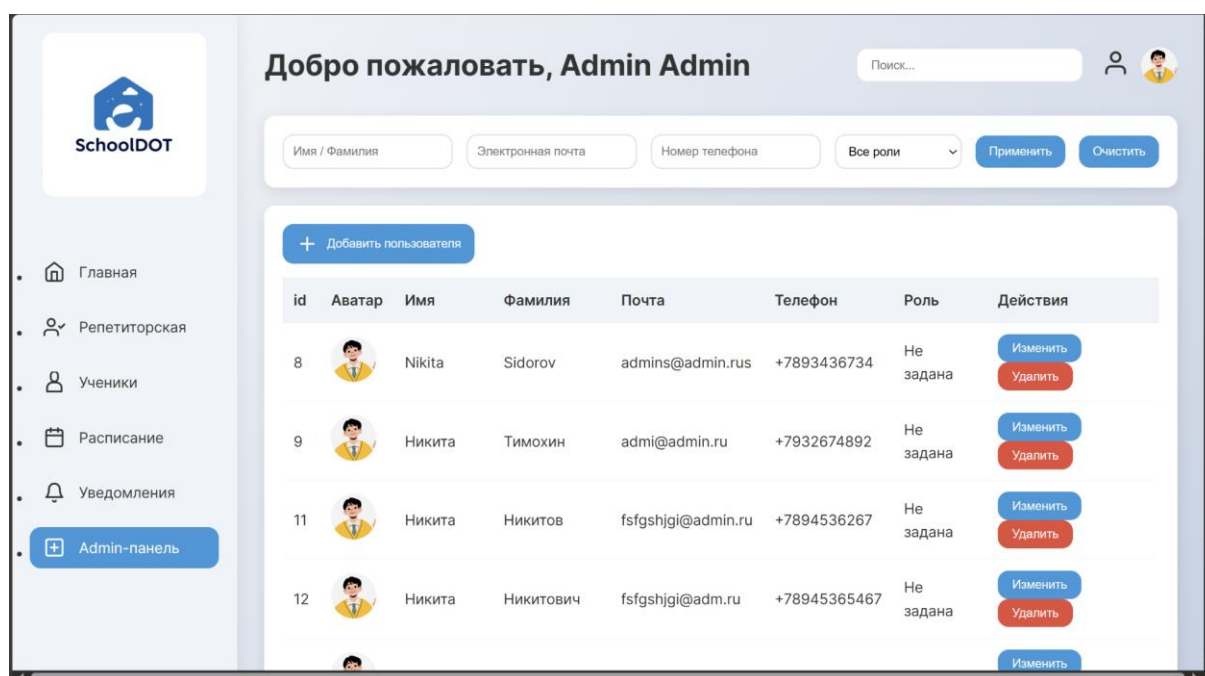


Рис. 5. Интерфейс панели администратора

5. Технологии реализации

Для реализации описанного решения использовались следующие программные средства: *Python* и *FastAPI* [9] для серверной разработки *API*, *PostgreSQL* как основная СУБД с поддержкой *SQLAlchemy* для *ORM* и *Alembic* для миграций, *React* с *React Query*, *JavaScript*, *HTML* и *CSS* для динамического клиентского интерфейса, *Swagger/OpenAPI* для документации *API*, *Figma* для *UI*-прототипирования, а также *Visual Studio Code* и *PyCharm* в качестве *IDE*.

6. Реализация

Приложение построено по модульной архитектуре, где каждый компонент системы изолирован в отдельный каталог с типичными файлами: *dao.py* для доступа к данным (*CRUD*), *models.py* с моделями *SQLAlchemy*, *router.py* с *API*-маршрутами и *schemas.py* с *Pydantic*-схемами [6]. Конфигурационные параметры, такие как настройки базы данных и секреты, хранятся в *.env* и загружаются через класс

Settings в *config.py*, что повышает безопасность и гибкость. Подключение к базе данных реализовано в *database.py*, где определён базовый класс модели, а вспомогательные операции, например преобразование *JSON*, вынесены в *utils.py*. Все зависимости проекта описаны в *requirements.txt* (Рис. 6).

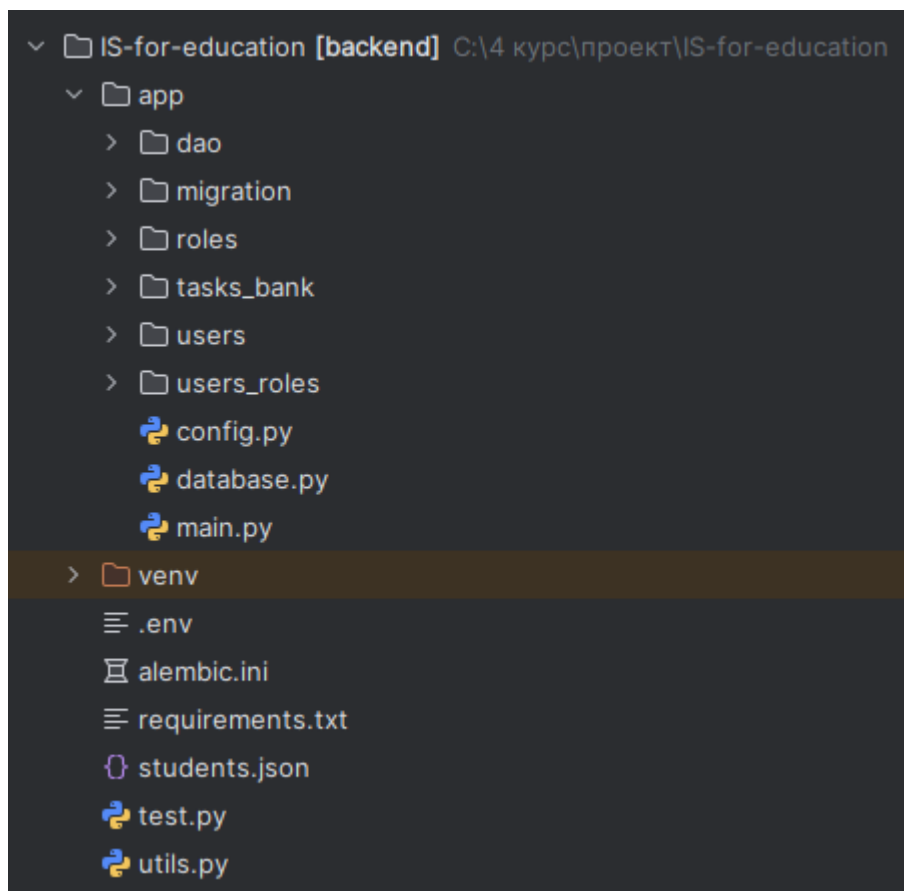


Рис. 6. Структура проекта

Для идентификации пользователя используется механизм *JWT*: после авторизации создаётся токен доступа с данными пользователя, который шифруется, подписывается и используется клиентом в заголовках *HTTP*-запросов к защищённым методам (см. Рис7) [7]. Реализованы регистрация с хешированием пароля, авторизация с верификацией, проверка токена, смена пароля, изменение аватара, обновление данных и валидация форматов телефона, электронной почты и пароля.



Рис. 7. Использование *JWT* для *HTTP*-запросов

Пользователи связаны с ролями, а роли – с правами доступа (*permissions*) по принципу многие-ко-многим. Для этого созданы отдельные таблицы и *API*. Если роль пользователя не содержит необходимых прав для вызова метода, система возвращает ошибку 403 *Forbidden*, а проверка доступа реализована через зависимость *verify_permissions* [8]. Поддерживаются создание и удаление ролей и прав, назначение ролей пользователям и проверка прав при вызове защищённых методов.

Ключевым модулем выступает банк задач. В нём поддерживаются три типа заданий: текстовые (описание и поле ответа), с медиа (описание и файл или ссылка) и тестовые (вопросы с вариантами ответов). В тестах предусмотрен ввод ответа, а также вопросы с одиночным и множественным выбором. Процесс создания тестов включает добавление вопросов и их типов, создание ответов и объединение их в тест. *API* позволяет создавать, редактировать, удалять и получать вопросы, их типы и ответы, а также получать полный состав теста по его *ID*. Текстовые задачи управляются отдельным модулем, содержат название и описание и поддерживают стандартные *CRUD*-операции (см. Рис. 8).

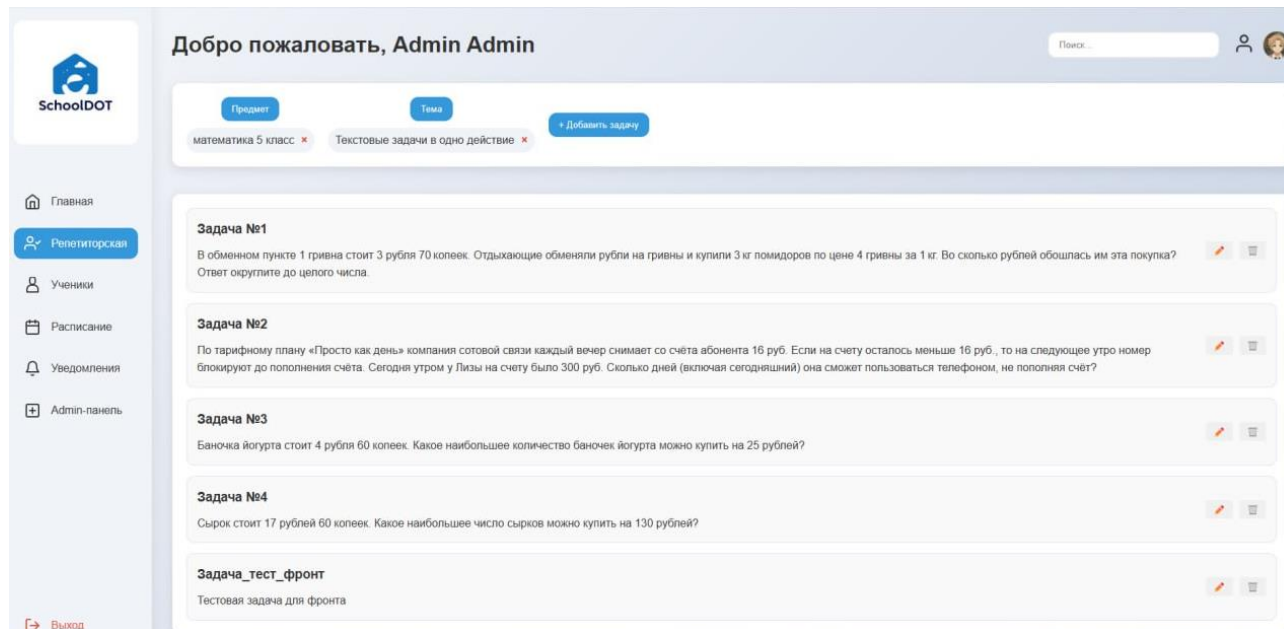


Рис. 8. Страница с текстовыми задачами

Кроме того, реализованы блоки расписания и уроков. Уроки имеют название, описание, оценку и возможность прикрепления заданий. Для них предусмотрены методы управления и связывания с задачами с указанием *ID* урока и задачи, ответа и оценки. Расписание представлено элементами с конкретной датой и временем занятия и содержит время начала и конца, ФИО ученика и репетитора, предмет, статус, цену и комментарий, а также связь с уроком. Для работы с расписанием реализованы *CRUD*-операции. Модуль учебных дисциплин позволяет управлять перечнем предметов (название и описание), которые используются в расписании и уроках.

База данных системы включает 17 основных таблиц, а на текущий момент реализовано 80 *API*-методов, а также страницы авторизации, личный кабинет пользователя, банк задач, админ-панель для управления системой и страницу расписания.

7. Внедрение и тестирование

После завершения основных этапов разработки информационной системы *SchoolDot* была проведена комплексная проверка её работоспособности и соответствия функциональным требованиям. Тестирование осуществлялось в два этапа: модульное тестирование *backend*-компонентов и интеграционное тестирование всего приложения в процессе опытной эксплуатации.

Серверная часть [10], реализованная с использованием *FastAPI*, проходила тестирование на уровне отдельных *API*-методов. Проверке подвергались ключевые эндпоинты: авторизация и регистрация пользователей, создание и изменение заданий, отправка и оценка ответов, формирование расписания и работа с платёжными данными. Для обеспечения обратной совместимости и корректной обработки исключений использовались встроенные механизмы обработки ошибок *FastAPI*.

Тестирование осуществлялось вручную с использованием *Swagger*-интерфейса, а также *Postman*. В качестве критериев корректности рассматривались:

- соответствие *HTTP*-кодов ожидаемому поведению;
- валидность структуры *JSON*-ответов;

- корректная работа с параметрами запроса;
- соблюдение логики разграничения прав доступа по ролям.

Фронтенд, реализованный на *React*, проверялся вручную путём прохождения типовых пользовательских сценариев (*use-cases*), таких как:

- вход под учётной записью репетитора/ученика;
- создание и просмотр заданий;
- выполнение задания и отправка ответа;
- отображение расписания и статистики;
- выставление счетов и отметка оплаты.

После внутреннего тестирования система была развёрнута в тестовой среде с использованием сервера, поддерживающего *PostgreSQL* и *Uvicorn*. Доступ к приложению был открыт группе студентов и преподавателей, которые провели пробное использование платформы для реальных учебных целей.

По результатам опытной эксплуатации были зафиксированы и устранены следующие замечания:

- некорректная валидация даты при создании занятия;
- отсутствие фильтра по ученикам в разделе «Статистика»;
- неточная работа уведомлений о неоплаченных занятиях.

Все ошибки были устранены в рамках финальных итераций разработки. Пользователи положительно оценили простоту интерфейса, логичную структуру и удобство в использовании.

Проект подразумевает коммерческую составляющую, вследствие чего планируется внедрение системы подписки на использование сервиса со стороны репетиторов. В текущей реализации проект полностью бесплатный, и требует опытной эксплуатации для последующего внедрения системы оплаты.

Заключение

В ходе выполнения проекта была разработана и внедрена информационная система *SchoolDot*, предназначенная для сопровождения частной репетиторской деятельности. Система предоставляет комплексный функционал для организации образовательного процесса, включая создание и проверку заданий, ведение расписания, отслеживание прогресса учащихся и учёт оплат.

Проект охватывал полный цикл создания информационной системы: от формализации требований и анализа аналогов до проектирования архитектуры, реализации клиентской и серверной частей, тестирования и опытной эксплуатации. В результате удалось построить стабильную и масштабируемую платформу, ориентированную на индивидуальное взаимодействие «преподаватель–ученик».

Практическая значимость работы заключается в устранении организационной фрагментированности, характерной для частной репетиторской практики, за счёт интеграции ключевых функций в единую цифровую среду. Система может быть адаптирована под другие образовательные форматы – малые онлайн-школы, корпоративное обучение, наставничество и др.

В дальнейшем планируется развитие системы в следующих направлениях:

- интеграция видеосвязи и синхронных занятий в реальном времени;
- расширение аналитических модулей (в том числе построение индивидуальных траекторий);
- добавление поддержки внешних *API* (оплаты, календарей, уведомлений);
- запуск *SaaS*-модели для коммерческого использования.

Результаты, полученные в процессе разработки, подтверждают актуальность автоматизации репетиторской деятельности и демонстрируют потенциал системного подхода к созданию *EdTech*-продуктов.

Список источников

1. Moodle : [open-source learning platform]. – URL: <https://moodle.org/> (дата обращения: 01.08.2025).
2. Онлайн-курсы - Stepik | Stepik: online education. – Stepik, 2013 – 2025. – URL: <https://stepik.org/> (дата обращения: 01.08.2025).
3. Finch — платформа для онлайн-репетиторов / Smile Tech. – Finch, 2025. – URL: <https://www.finch.study/> (дата обращения: 01.08.2025).
4. Семакин И. Г. Информационные технологии. Базовый курс: учебник / И. Г. Семакин, А. А. Шестаков. – 8-е изд., стер. – Москва: БИНОМ; Лаборатория знаний, 2021. – 400 с.
5. Васильев В. А. Основы проектирования информационных систем: учеб. Пособие / В. А. Васильев, А. И. Лопатин. – Москва: КноРус, 2022. – 238 с.
6. React: [веб-сайт]. – Meta Platforms, Inc.¹, 2025. – URL: <https://react.dev> (дата обращения: 01.08.2025).
7. PostgreSQL: Documentation. – The PostgreSQL Global Development Group, 1996-2025. – URL: <https://www.postgresql.org/docs/> (дата обращения: 01.08.2025).
8. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.
9. FastAPI framework, high performance, easy to learn, fast to... : [официальная документация]. – URL: <https://fastapi.tiangolo.com/> (дата обращения: 01.08.2025).
10. Ермаков С. А. Основы построения клиент-серверных приложений / С. А. Ермаков, И. В. Тихонов. Санкт-Петербург: Питер, 2020. – 288 с.

¹ Деятельность Meta Platform Inc. на территории РФ запрещена.