Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Самарский национальный исследовательский университет  
 имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

**Отчет по лабораторной работе**

Дисциплина: «Технологии сетевого программирования»

Выполнили:

Пироженкова И.С., группа 6303.

Нефёдова А.М., группа 6303.

Самара, 2025

# **Введение**

«TutorHub» – веб‐сервис для доступного и интересного взаимодействия между учениками и преподавателями.

Наш сайт объединит учеников и преподавателей в удобном и доступном формате, делая обучение более эффективным и интересным!

# 

Рисунок 1 - Логотип сайта

### **Целевая аудитория и её потребности**

### Ученики испытывают сложности с поиском мотивации и удобным форматом хранения всех материалов занятий, а преподаватели нуждаются в платформе, где можно легко взаимодействовать с учениками и организовывать процесс обучения.

### На данный момент репетиторы используют мессенджеры (WhatsApp, Telegram), но в них нет единой системы управления обучением. А ученики обращаются к сайтам онлайн-курсов, но там нет гибкости и личного взаимодействия с преподавателем.

#### **Цели проекта**

* Создать удобную платформу для взаимодействия учеников и преподавателей.
* Сделать образовательный процесс более доступным, увлекательным и интерактивным.
* Обеспечить преподавателям эффективные инструменты для организации занятий.

#### **Задачи проекта:**

* Разработать интуитивно понятный интерфейс для поиска и управления обучением.
* Внедрить систему рейтингов, расписания, домашнего задания и оплаты для удобства пользователей.
* Обеспечить гибкость и персонализацию обучения, учитывая потребности учеников.

### **Проблематика проекта**

* Проблемы учеников:
  + Отсутствие единой среды – приходится использовать разные платформы для занятий, общения и хранения материалов.
  + Низкая мотивация – традиционные методы обучения скучны, нет интерактивности.
  + Недостаток персонализированного подхода – программы не адаптируются под ученика.
* Проблемы преподавателей:
  + Трудности с организацией процесса – приходится вести учёт занятий и расписания вручную.

**Роли пользователей**

* Ученик
* Преподаватель

**Список используемых технологий с обоснованием выбора**

1. MySQL workbench – СУБД для хранения данных.

Причина выбора: Изучение новой технологии и возможность автоматического создания скрипта, открытый исходный код, бесплатна

1. HTML, CSS, JavaScript – языки, которые будут использоваться для написания фронтэнда.

Причина использования — наличие опыта работы с данным стеком технологий.

1. Java- бэкэнд

Причина выбора: Высокая производительность, поддержка многопоточности, платформонезависимость и развитая экосистема.

1. Git – система контроля версий.

Причина использования — повышение надежности разработки.

# 

# 

# 

# 

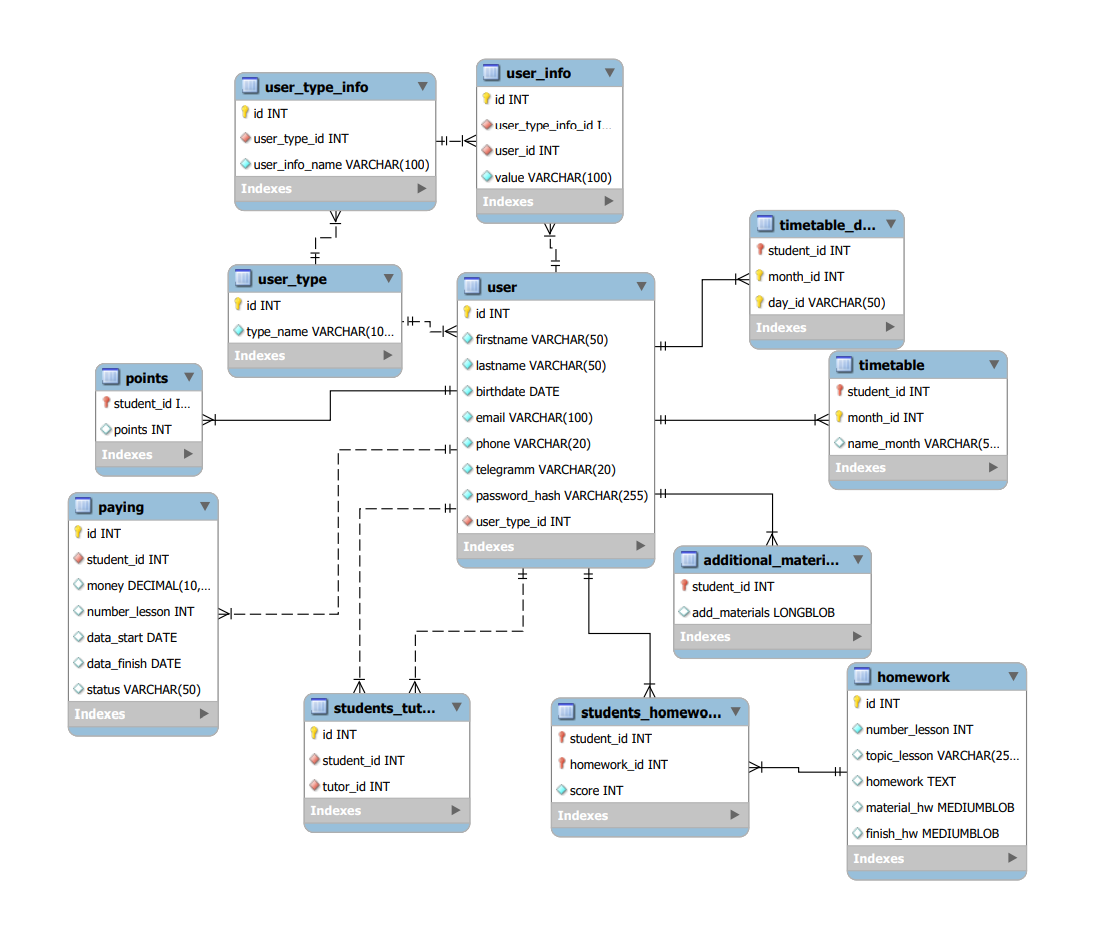
# 

# 

# 

# 

# **Схема базы данных**

****

# **Интерфейс пользователя**

Ниже представлены макеты страниц веб-сайта со стороны ученика и преподавателя.

Этот экран – стартовая точка взаимодействия пользователя с платформой. Пользователю будет предложено выбрать тип пользователя (преподаватель / студент).

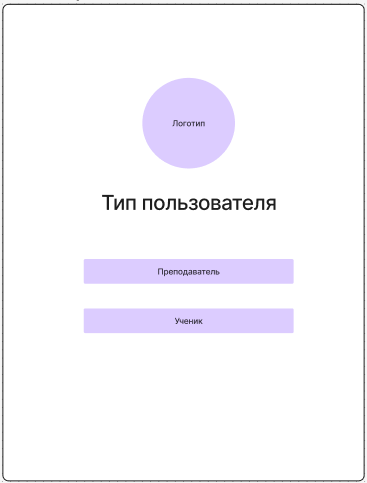


Рисунок 2 - Выбор типа пользователя

Далее будет предложено выбрать одно из двух действий: войти в существующий аккаунт (для авторизации зарегистрированных пользователей) или зарегистрироваться (для создания новой учетной записи).

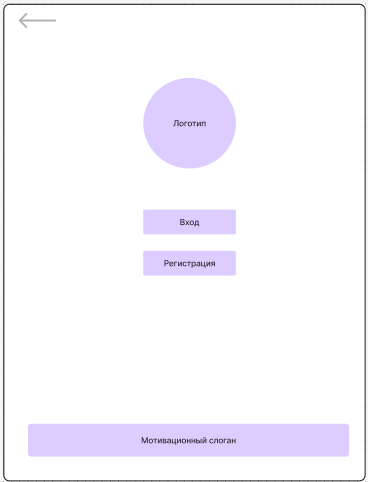


Рисунок 3 - Выбор входа или регистрации

Далее представлен макет страницы для регистрации ученика. Данные помещаются в базу данных. В дальнейшем после успешной регистрации ученик сможет войти в личный кабинет.



Рисунок 4 - Регистрация ученика

Также рассмотрим макет страницы для регистрации преподавателя. Данные аналогично помещаются в базу данных. В дальнейшем после успешной регистрации преподаватель сможет войти в личный кабинет.



Рисунок 5 - Регистрация преподавателя

Следующий экран предназначен для авторизации зарегистрированных пользователей. Данные хранятся в базе данных. Вводится email и пароль и пользователь попадает в свою учетную запись.

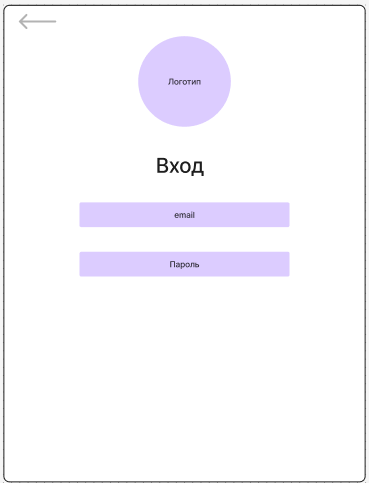


Рисунок 5 - Регистрация преподавателя

Для начала рассмотрим интерфейс для ученика. Главный экран, с которого пользователь может перейти в разные разделы (профиль, баллы, домашнее задание, правила и оплата, дополнительные материалы, расписание). Каждая кнопка ведёт к соответствующему разделу платформы.

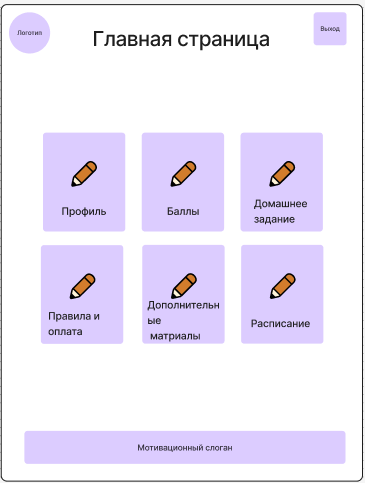


Рисунок 6 - Главная страница личного кабинета ученика

Один из разделов личного кабинета - профиль. Все данные хранятся в базе данных.

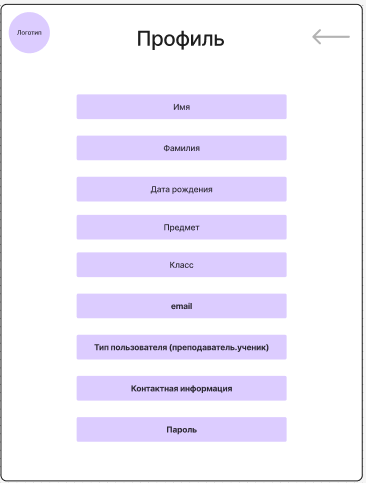


Рисунок 7 - Профиль

Следующая страница отображает личные баллы пользователя, а также позволяет сравнить свои результаты с другими учениками. Рейтинговая система используется для мотивации учеников. Баллы можно как получать (за выполненное дз, хорошую работу на уроке), так и терять (за опоздание, плохую работу). Все данные хранятся в базе данных.

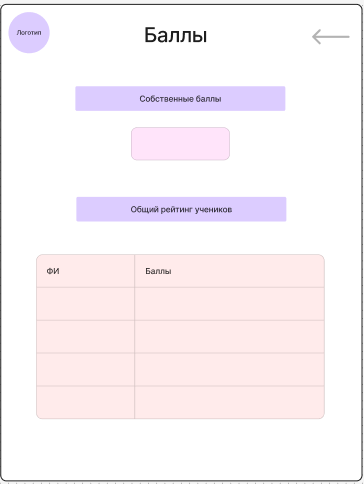


Рисунок 8 - Балльная система

Страница с домашним заданием позволяет ученикам видеть список заданий и прикреплённые материалы к каждому уроку. А также есть возможность прикладывать выполненное дз после каждого урока. Все данные хранятся в базе данных.



Рисунок 9 - Система домашнего задания

Дополнительные материалы - раздел, в котором преподаватели могут выкладывать полезные материалы. Ученики могут загружать файлы или открывать ссылки. Все данные хранятся в базе данных.

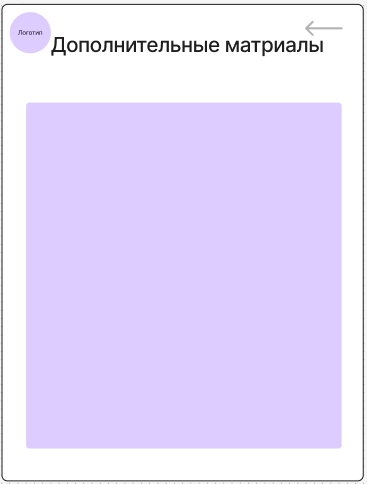


Рисунок 10 - Система дополнительных материалов

Страница с расписанием позволяет просматривать расписание занятий, есть возможность переключаться между месяцами. Все данные хранятся в базе данных.

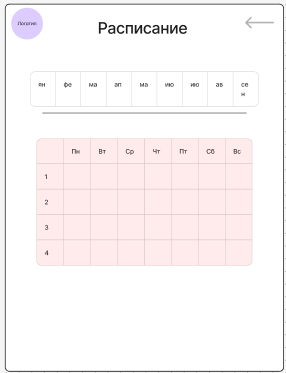


Рисунок 11 - Расписание

Последний раздел - правила и оплата. Раздел для ознакомления с правилами и условиями работы с учеником. В таблице оплата представлена информация о сумме абонементов, сроке действия абонемента, а также статусе оплаты. Если оплаты не было, ученику приходит напоминание.

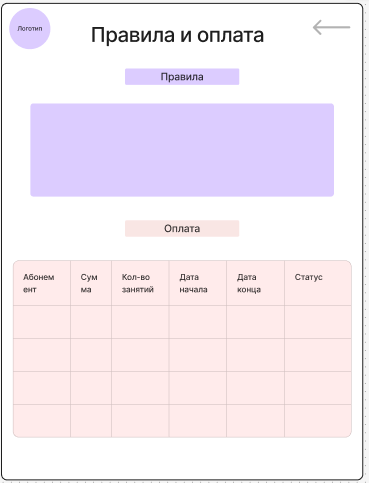


Рисунок 12 - Правила и оплата

Далее рассмотрим интерфейс для преподавателя. Главный экран, с которого пользователь может перейти в разные разделы (профиль, ученики, расписание учеников). Каждая кнопка ведёт к соответствующему разделу платформы.

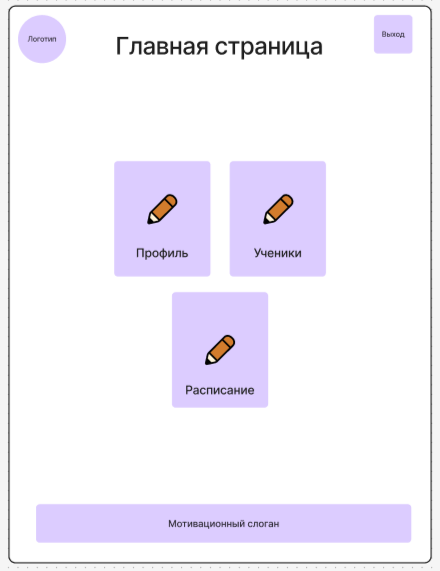


Рисунок 13 - Главная страница личного кабинета преподавателя

Все данные хранятся в базе данных.

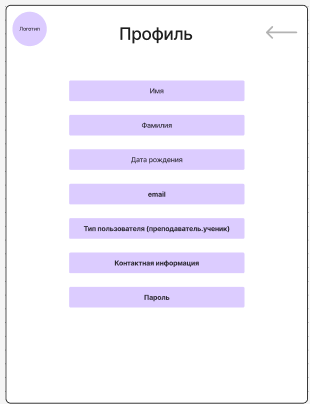


Рисунок 14 - Профиль преподавателя

Второй раздел преподавателей - таблица с данными об учениках, хранящая все необходимые данные для работы с учеником. Также можно просмотреть статус оплаты абонемента, и если оплаты не было - отправить напоминание.

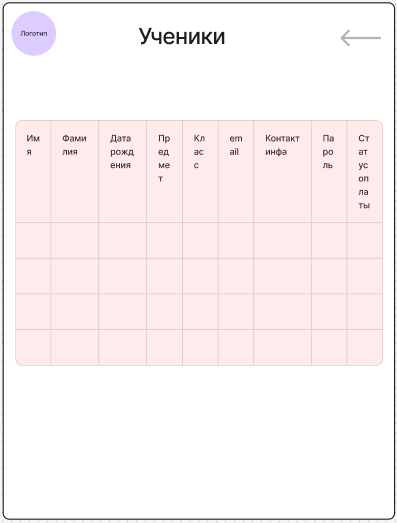


Рисунок 15 - Информация об учениках

При нажатии на строку ученика, преподаватель переходит на главную страницу профиля ученика (рисунок 6). Макет соответственно всех разделов был представлен выше.

Страница с расписанием преподавателя позволяет просматривать расписание занятий одновременно всех учеников, есть возможность переключаться между месяцами. Все данные хранятся в базе данных.

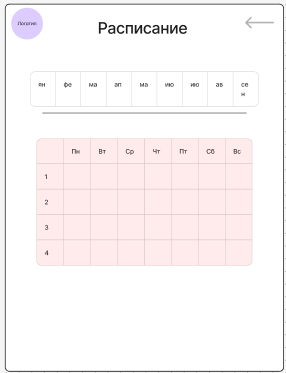


Рисунок 11 - Расписание

Преподаватель имеет права редактировать все данные в таблицах на страницах, а ученик может только просматривать все страницы, без права их изменения.

# **Приложение**

Ссылка на github:<https://github.com/newpinksnow/TutorHub>

Ссылка на API: [https://editor-next.swagger.io/](https://vk.com/away.php?to=https%3A%2F%2Feditor-next.swagger.io%2F&utf=1)