

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ЗАПРОСОВ НА ГЕНЕРАЦИЮ ТЕКСТА С ПРИМЕНЕНИЕМ API YANDEXGPT И ТЕХНОЛОГИИ БЛОКЧЕЙН

Автор работы:
Студент группы КЭ-402
Никитин Н. П.

Научный руководитель:
доцент кафедры СП, к.ф.-м.н.
Латипова А.Т.

АКТУАЛЬНОСТЬ

- 1) Рост популярности
- 2) Постоянное развитие
- 3) Разнообразие возможностей и функций
- 4) Эффективность и доступность
- 5) Удобство и низкие затраты
- 6) Повышение качества и точности
- 7) Экономия времени и ресурсов

Цели и задачи

Цель: разработка веб-приложения для запросов на генерацию текста с применением API YandexGPT и технологии блокчейн

Задачи:

- 1) произвести обзор аналогичных решений
- 2) определить требования веб-приложения
- 3) спроектировать веб-приложение
- 4) реализовать веб-приложение
- 5) произвести тестирование разработанной системы

ChatGPT

The screenshot displays the ChatGPT web interface. On the left is a dark sidebar with a home icon, a 'ChatGPT' header, and a 'Обзор GPT' (GPT Overview) section. Below this, a 'Вчера' (Yesterday) section lists recent conversations: 'Clarify User's Request', 'Request for Summary.', and 'Фотосессия: Цвета и стиль.'. At the bottom of the sidebar is a 'Добавить рабочую область Team' (Add workspace) button with a note 'Совместная работа в плане Team'.

The main area features the 'ChatGPT 4o' header and the OpenAI logo. A settings menu is open in the top right, containing options: 'Мой план', 'Мои GPT', 'Настроить ChatGPT', 'Настройки', and 'Выйти'. Below the logo are four interactive cards with icons and text: 'Спроси меня о древних цивилизациях', 'Текст с приглашением соседей на...', 'Узнать Сеул как местный житель', and 'Создай игру по программированию'.

At the bottom, there is a text input field with a paperclip icon and the text 'Сообщить ChatGPT', and a send button with an upward arrow. A footer note states: 'ChatGPT может допускать ошибки. Рекомендуем проверять важную информацию.' and a help icon is visible in the bottom right corner.

GigaChat

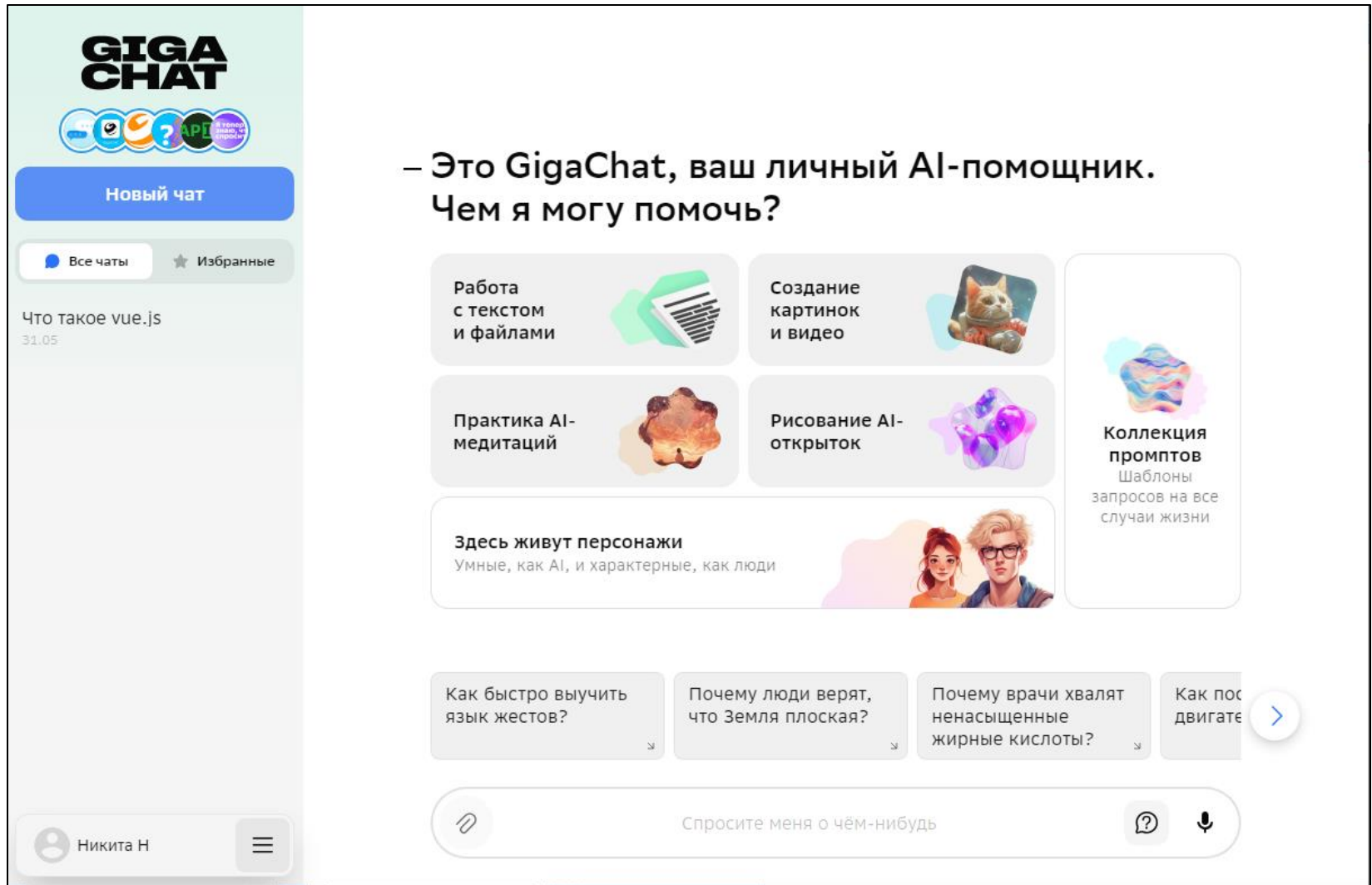


Диаграмма вариантов использования

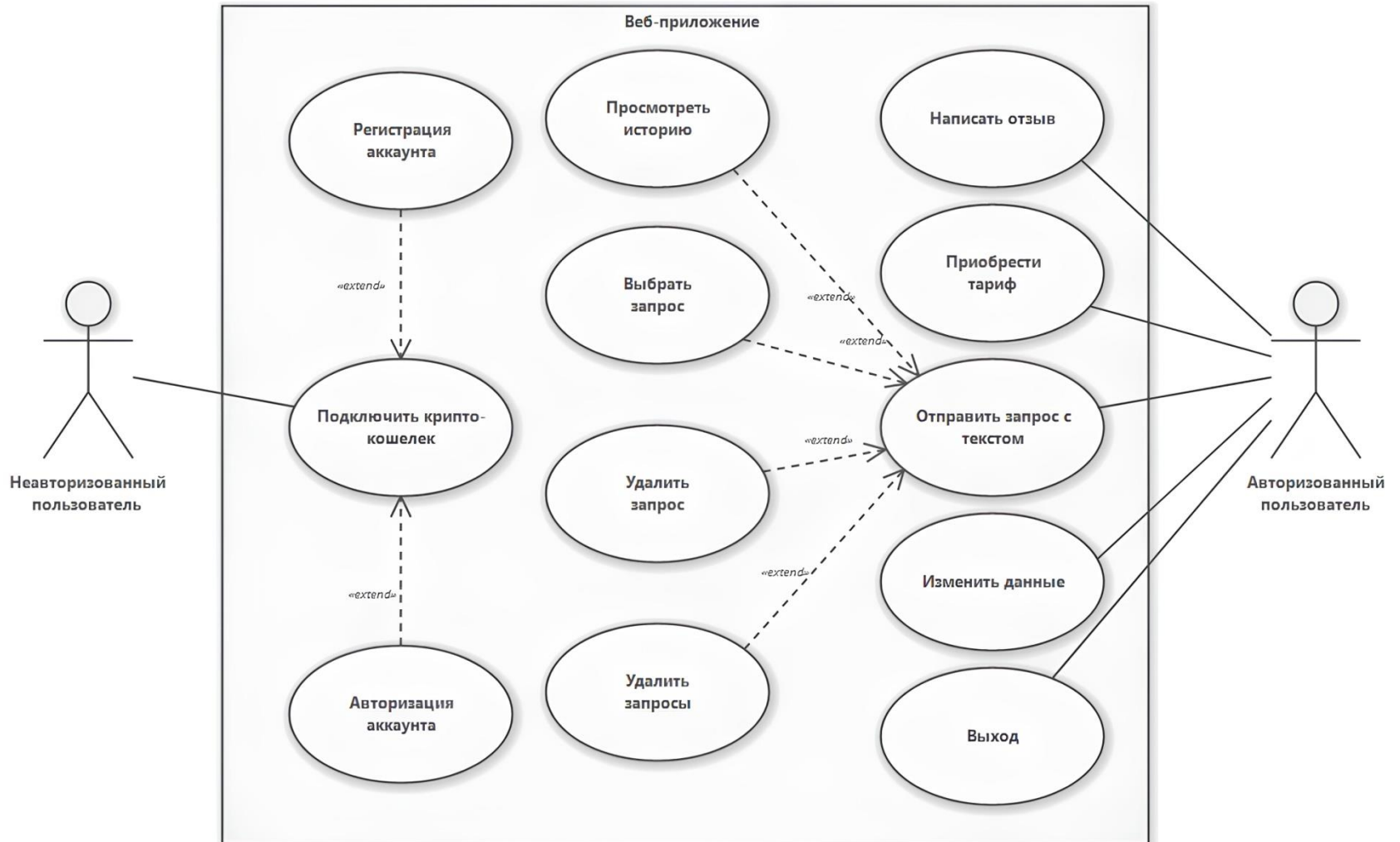
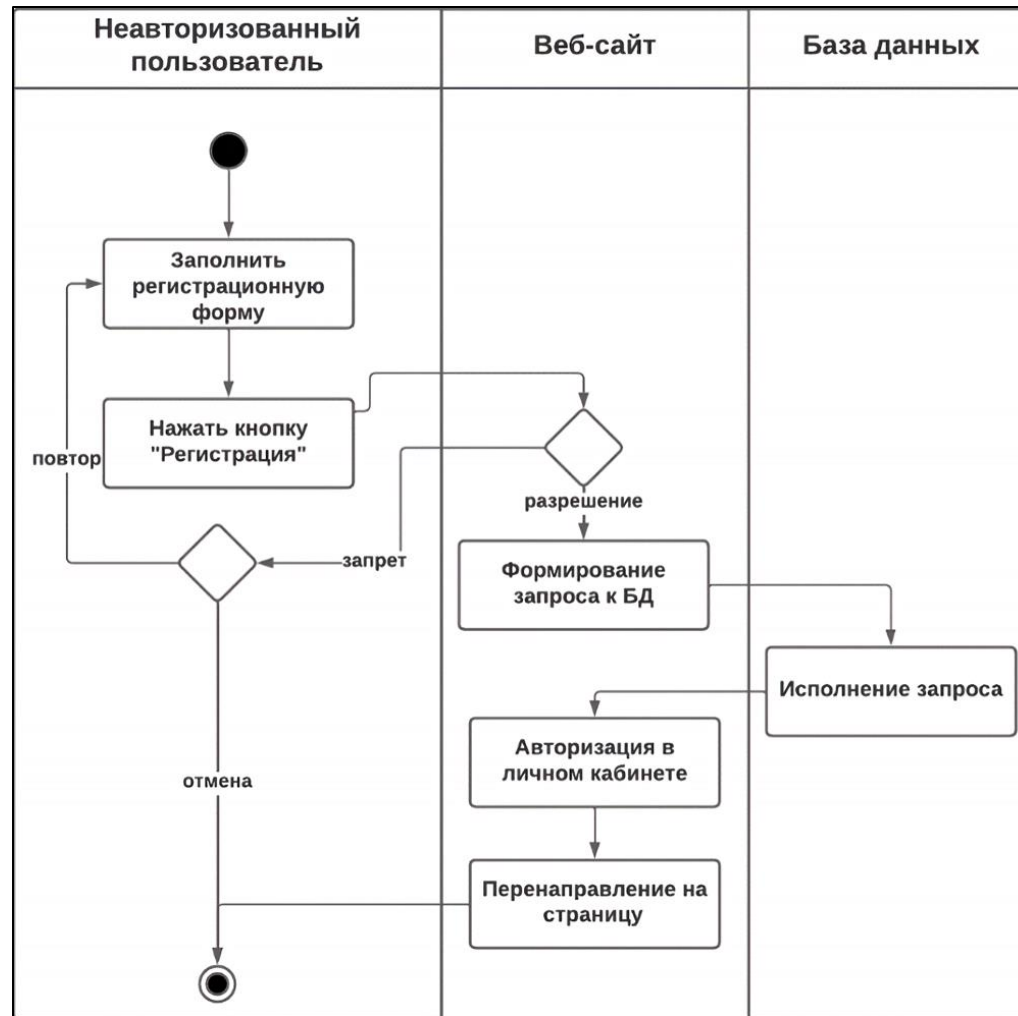


Диаграмма деятельности



Процесс регистрации нового пользователя

Средства разработки

- **Язык программирования:** JavaScript, TypeScript
- **Препроцессоры:** CSS-Modules, SCSS
- **Веб-сервер:** Node.js
- **Фреймворк:** Nest.js, Next.js
- **Библиотеки:** Axios, React, TypeORM, UUID, Bcrypt, Cookie-Parser, Web3, Hardhat, Ethers
- **СУБД:** PostgreSQL

Архитектура системы

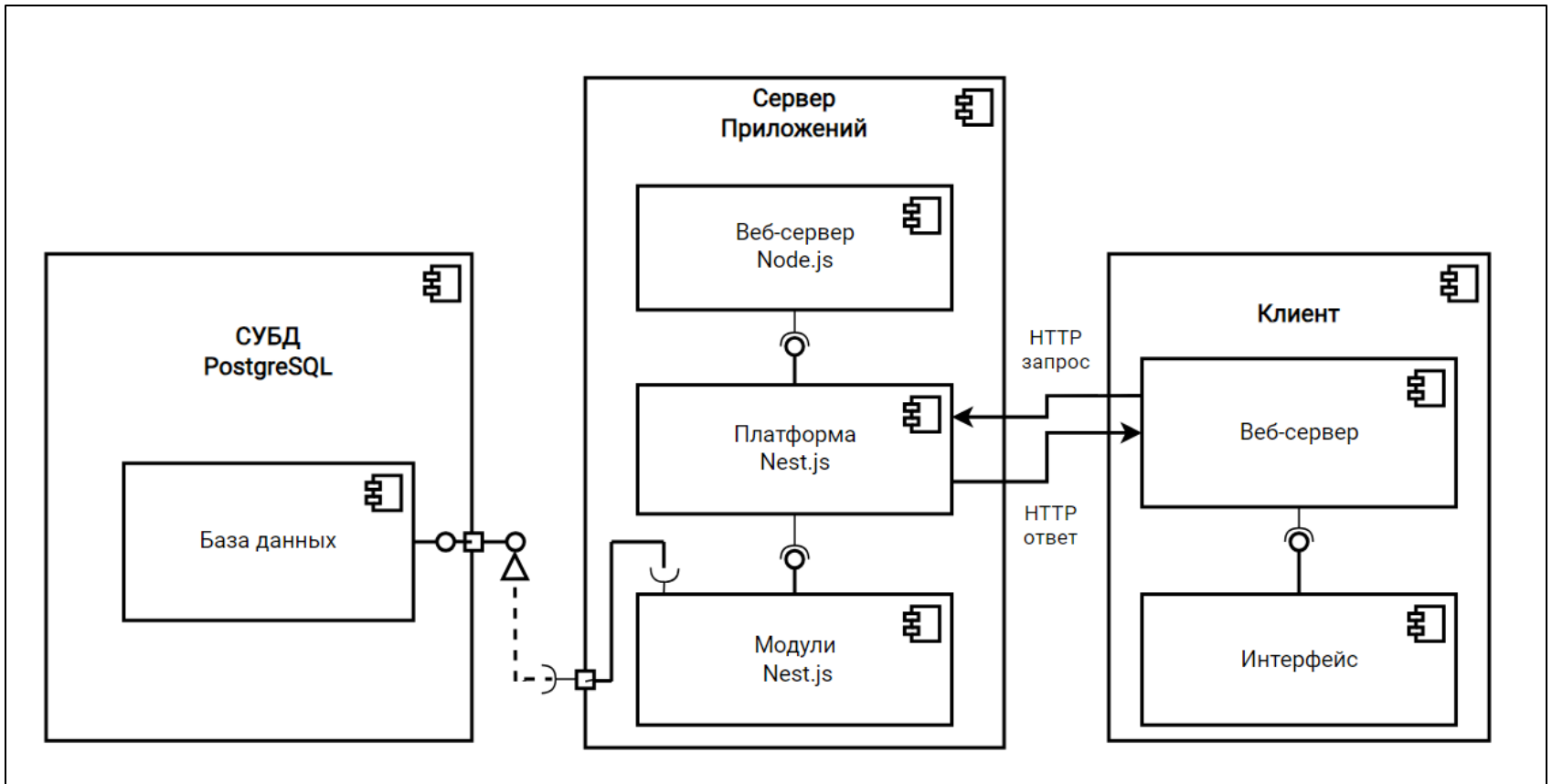
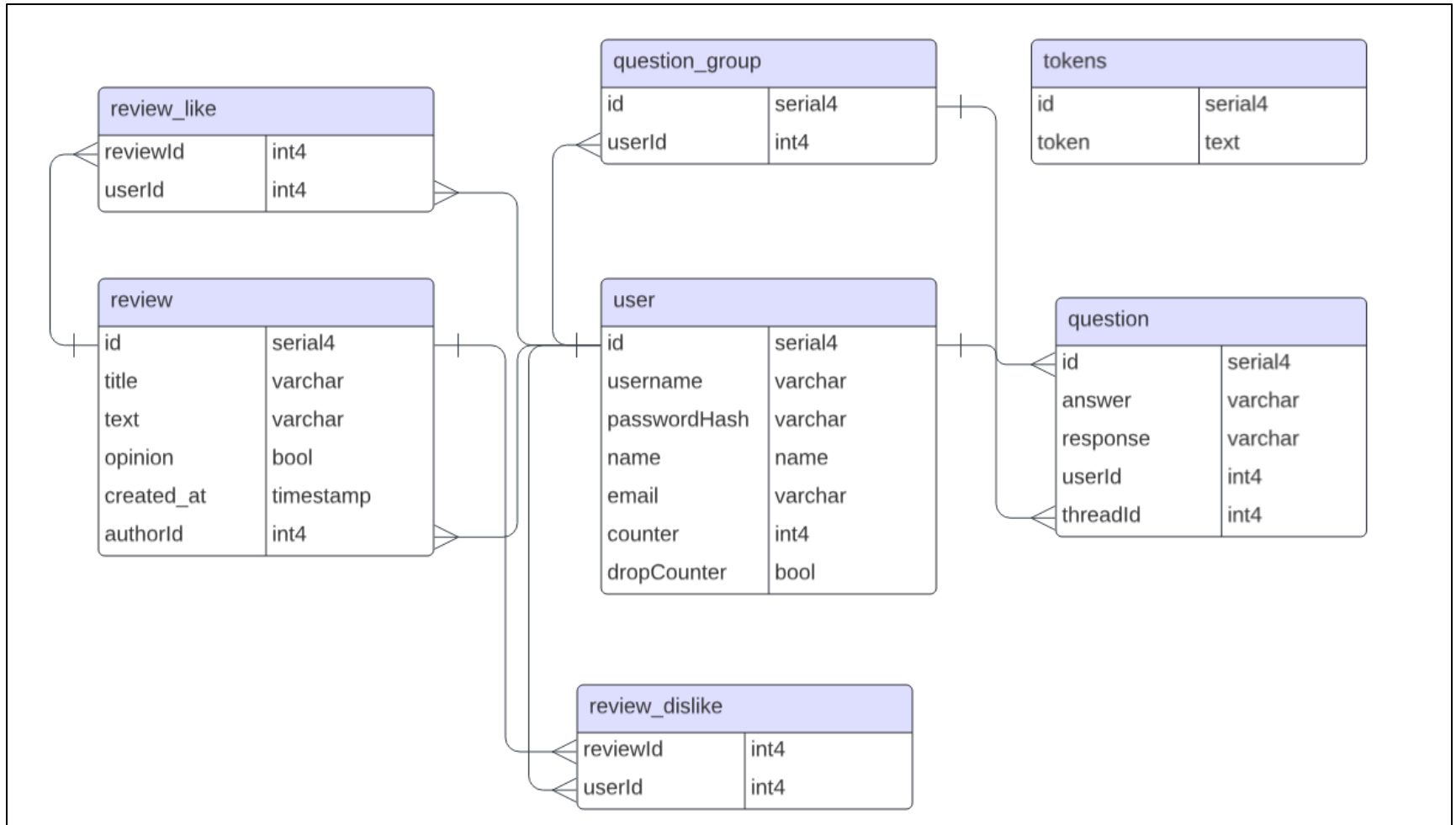


Схема базы данных



Реализация подключения API

```
async function reqBearer(){
  await axios({
    url: 'https://llm.api.cloud.yandex.net/foundationModels/v1/completion',
    method: 'POST',
    headers: {
      'Content-Type' : 'application/x-www-form-urlencoded',
      'RqUID' : uuidv4(),
    },
    auth: {
      username: 'a604ad8a-0262-45bc-99ad-81515cabec5e',
      password: '2455dfa0-5db3-436b-8b46-88c31c6cc549',
    },
    data: stringify({
      scope: 'YandexGPT_API'
    }),
    httpsAgent: new https.Agent({
      rejectUnauthorized: false,
    })
  }).then(async (res) => {
    await MainDatasource.getRepository(tokens).delete({
      token: Not(NotNull())
    })
    await MainDatasource.getRepository(tokens).save({
      token: res.data.access_token
    })
  }).catch((e)=>{
    console.log('Ошибка получения токена', e.response.data)
  })
}

bootstrap().then(res=>{
  console.log('Server starts.')
  MainDatasource.initialize().then(async ()=>{
    console.log('Postgres SQL DB init')
    await reqBearer()
    setInterval(async ()=>{
      await reqBearer()
    }, 1000 * 60 * 15)
  })
})
```

Реализация отправки запроса

```
return this.httpService.post(  
    'https://llm.api.cloud.yandex.net/foundationModels/v1/completion',  
    {  
        model: "gpt://*****/yandexgpt-lite",  
        //@ts-ignore  
        messages: history,  
        temperature: 1,  
        top_p: 0.1,  
        n: 1,  
        stream: false,  
        max_tokens: 512,  
        repetition_penalty: 1,  
        update_interval: 0,  
    },  
    {  
        method: "post",  
        headers: {  
            'Authorization': `Bearer ${String(BearerToken.token)}`,  
            'Content-Type': `application/json`,  
        },  
        httpsAgent: new https.Agent({  
            rejectUnauthorized: false,  
        })  
    }  
).pipe(  
    map((response: any) => { //@ts-ignore  
        this.checkAndSave(  
            req.body?.prompt,  
            response.data.choices[0].message.content,  
            req,  
            req.body?.threadId  
        )  
        return {bot: response.data.choices[0].message.content}  
    })  
);
```

Реализация модуля аутентификации

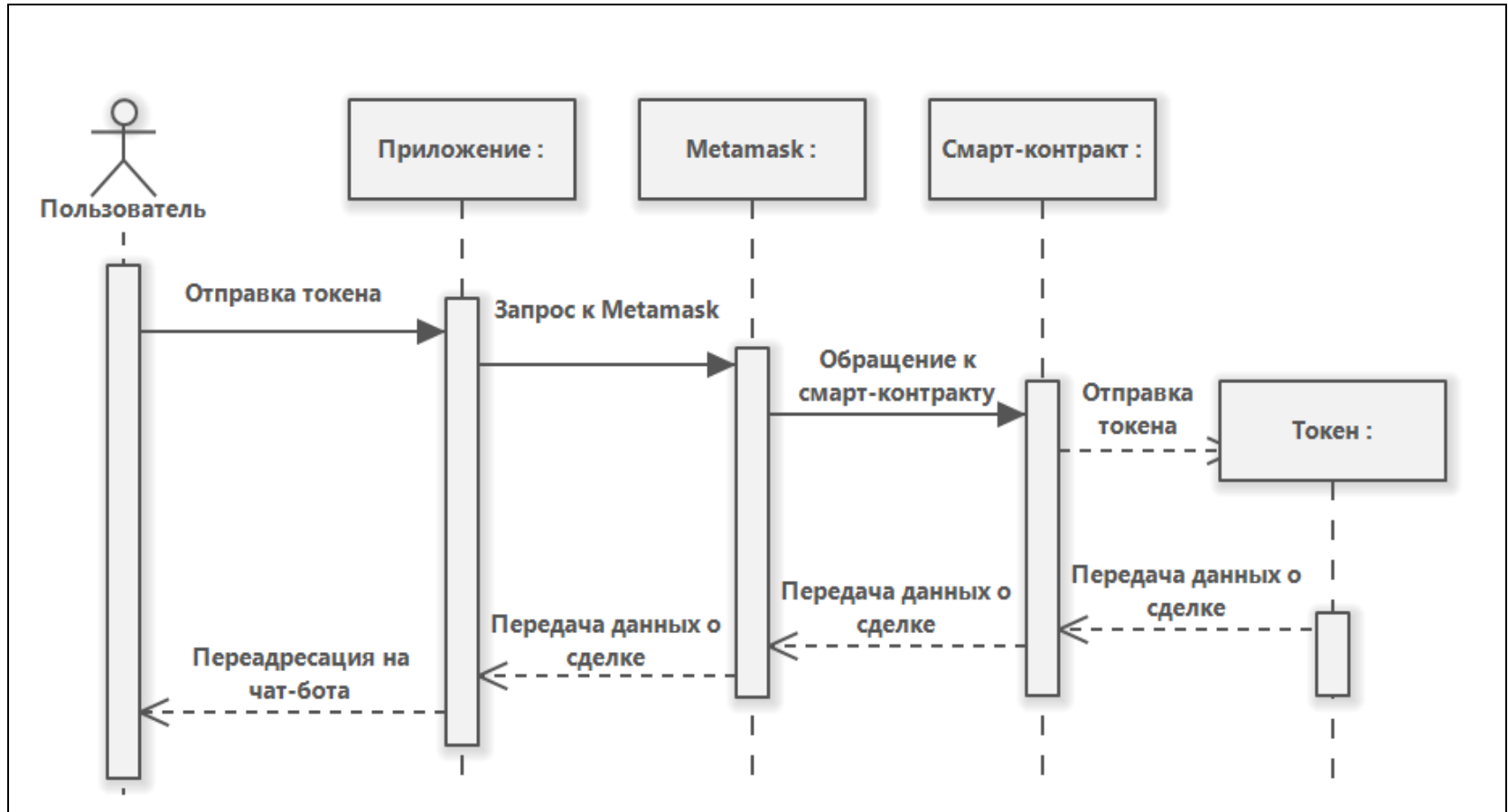
```
@Injectable()
export class AuthGuard implements CanActivate {
  constructor(private jwtService: JwtService, private reflector: Reflector) {}

  async canActivate(context: ExecutionContext): Promise<boolean> {
    const isPublic = this.reflector.getAllAndOverride<boolean>(IS_PUBLIC_KEY, [
      context.getHandler(),
      context.getClass(),
    ]);
    if (isPublic) {
      return true;
    }

    const request = context.switchToHttp().getRequest();
    const token = request.cookies['Bearer'];
    if (!token) {
      throw new UnauthorizedException();
    }
    try {
      request['user'] = await this.jwtService.verifyAsync(token, {
        secret: jwtConstants,
      });
    } catch {
      throw new UnauthorizedException();
    }
    return true;
  }

  private extractTokenFromHeader(request: Request): string | undefined {
    const [type, token] = request.headers.authorization?.split(' ') ?? [];
    return type === 'Bearer' ? token : undefined;
  }
}
```

Диаграмма последовательности



Главная страница

[Подключить кошелек](#)


AI BOT ДЛЯ УПРОЩЕНИЯ ВАШЕЙ ЖИЗНИ

AI BOT - это надежный инструмент искусственного интеллекта, который использует алгоритмы искусственного интеллекта для создания различного контента. Этот инструмент был разработан для того, чтобы помочь пользователям без особых усилий создавать высококачественный контент, экономя энергию и время, сохраняя при этом постоянный уровень креативности и точности.


- Отвечает на вопросы
- Ведёт диалог
- Пишет код
- Предлагает идеи
- Решает задачи


★★★★★

Техника, позволяющая писать весь ваш материал в 10 раз быстрее.


 | Генерация кода


AI BOT отладит, сгенерирует, отформатирует код.




 | SEO эксперт


Способен повысить значимость сайта в глазах поисковых систем.




 | Фото капчи

Способен придумывать качественные подписи для фото




 | Копирайтинг

Можно использовать для написания статей, маркетинговых текстов,...



Страница регистрации



Добро пожаловать в AI Bot !

Авторизация Регистрация

Логин

Email

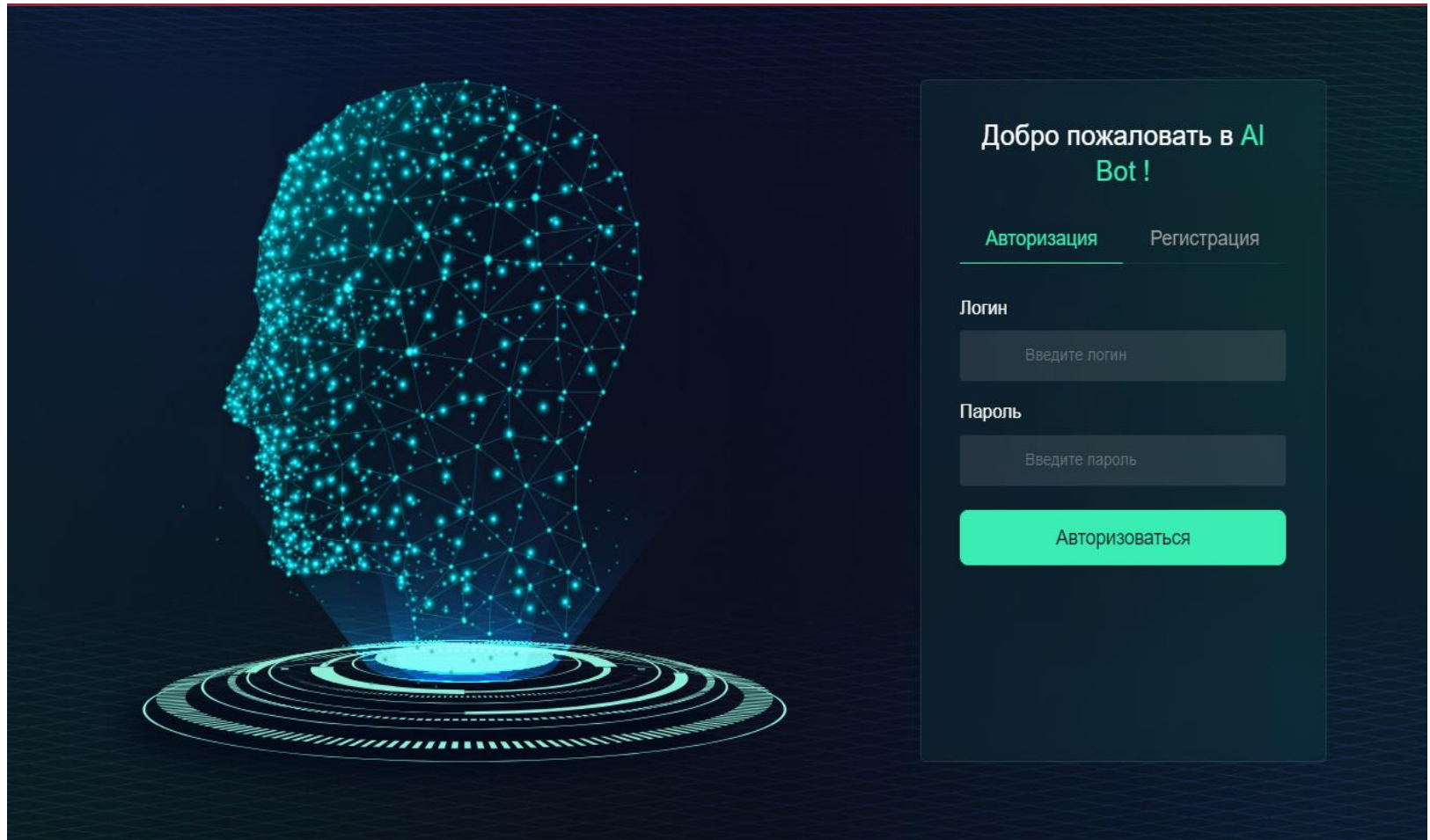
Имя

Пароль

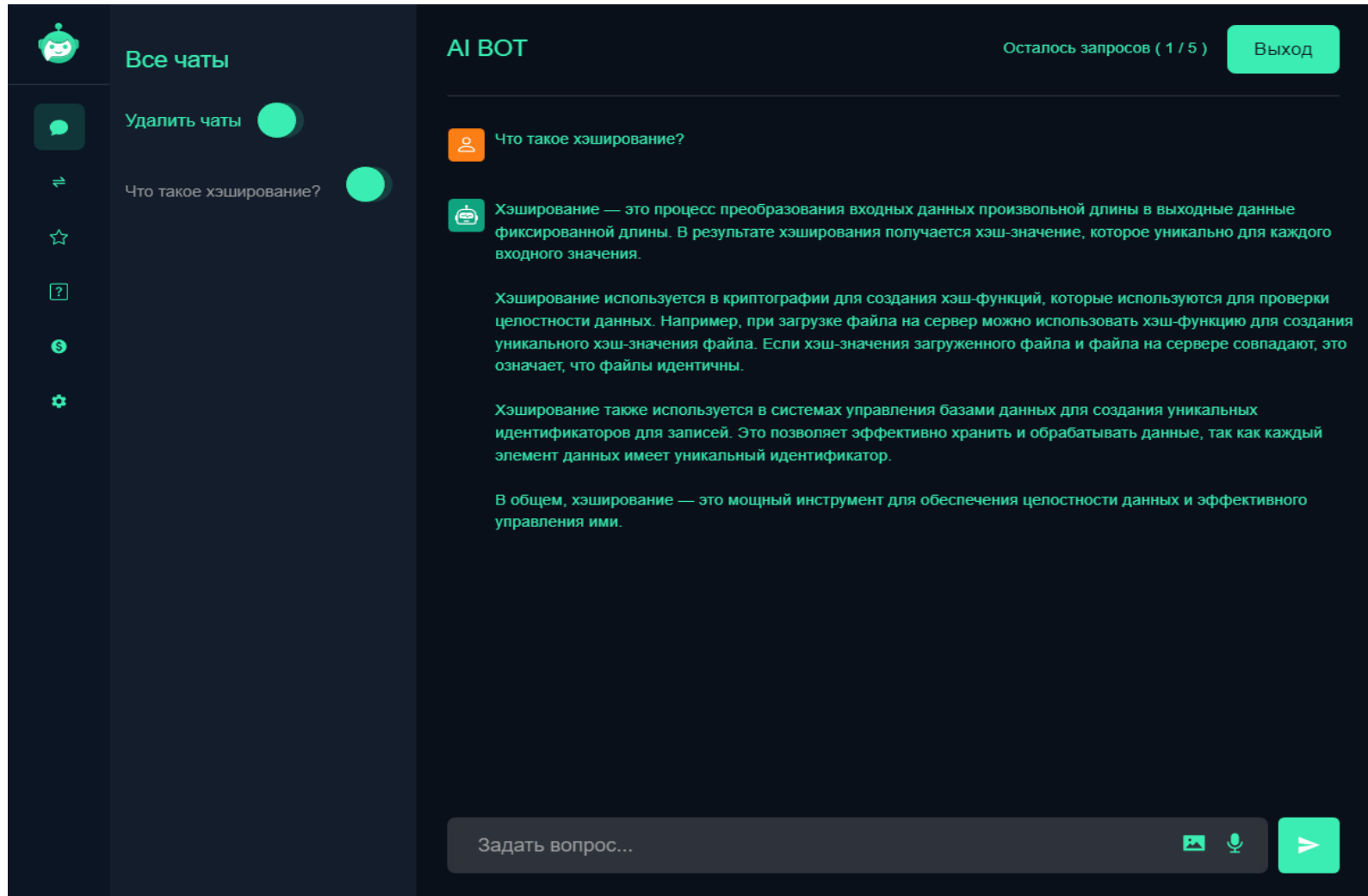
Подтвердить пароль

Зарегистрироваться

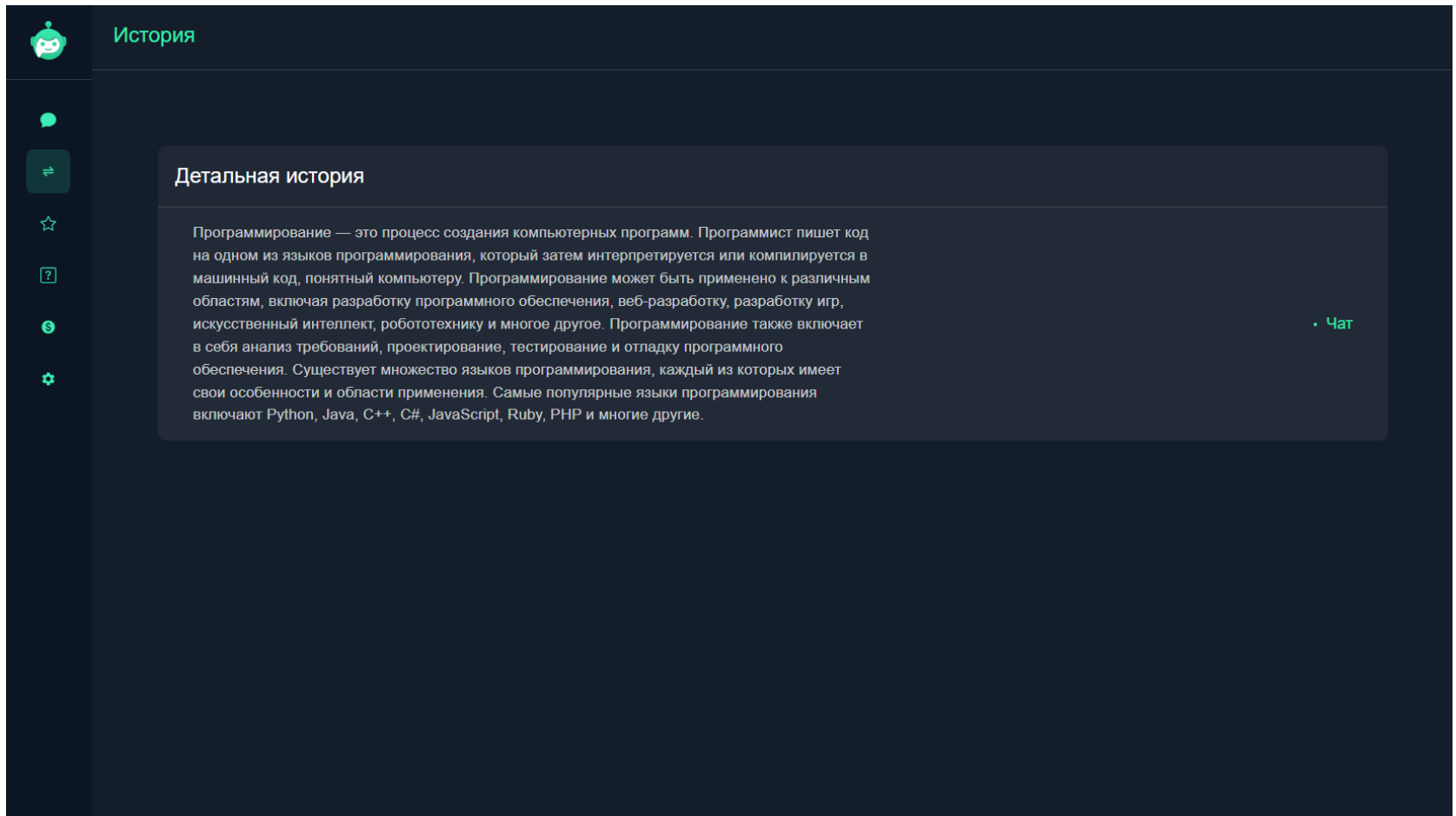
Страница авторизации



Интерфейс веб-приложения



История ответов



The screenshot shows a dark-themed chat application interface. On the left is a vertical sidebar with several icons: a green chat bubble with a white face, a blue speech bubble, a blue square with a white right-pointing arrow, a blue star, a blue square with a white question mark, a blue speech bubble with a white dot, and a blue gear. The main area is titled "История" (History) in green text. Below the title is a dark grey box with the heading "Детальная история" (Detailed history) in white. The main content is a paragraph of text in white, describing programming. To the right of the text is a small green chat bubble icon with the word "Чат" (Chat) next to it.

История

Детальная история

Программирование — это процесс создания компьютерных программ. Программист пишет код на одном из языков программирования, который затем интерпретируется или компилируется в машинный код, понятный компьютеру. Программирование может быть применено к различным областям, включая разработку программного обеспечения, веб-разработку, разработку игр, искусственный интеллект, робототехнику и многое другое. Программирование также включает в себя анализ требований, проектирование, тестирование и отладку программного обеспечения. Существует множество языков программирования, каждый из которых имеет свои особенности и области применения. Самые популярные языки программирования включают Python, Java, C++, C#, JavaScript, Ruby, PHP и многие другие.

• Чат

Приобретение тарифа

The image shows a web application interface for selecting a tariff plan. The main screen is titled "Тарифы" (Tariffs) and displays a "Выбрать план:" (Select plan:) section. A "Премиум" (Premium) plan is highlighted, featuring a diamond icon and a list of benefits:

- Интеллектуальный чат-бот с искусственным интеллектом
- Неограниченное количество ответов на вопросы
- Нет необходимости продлевать тариф

A "Loading..." button is visible at the bottom of the premium plan details. Overlaid on the right is a MetaMask transaction confirmation window. The window title is "MetaMask" and the URL is "localhost 8545". It shows a transaction from "Account 5" to "Account 1" with a value of "0 ETH". A warning message states: "You are sending 0 ETH." The estimated fee is 0.00002258 ETH, and the total amount to be sent is 0.00002258 ETH. The window includes "Отклонить" (Decline) and "Подтвердить" (Confirm) buttons.

Окно отзыва

Оцените работу приложения и напишите
ОТЗЫВ ✕

Тема отзыва

Отзыв

Поставить оценку

Нравится

Не нравится

Отправить отзыв

Настройки профиля

Настройки

Мой профиль

Mike
Mike@gmail.com

Имя: Mike

Email адрес: Mike@gmail.com

Обновить

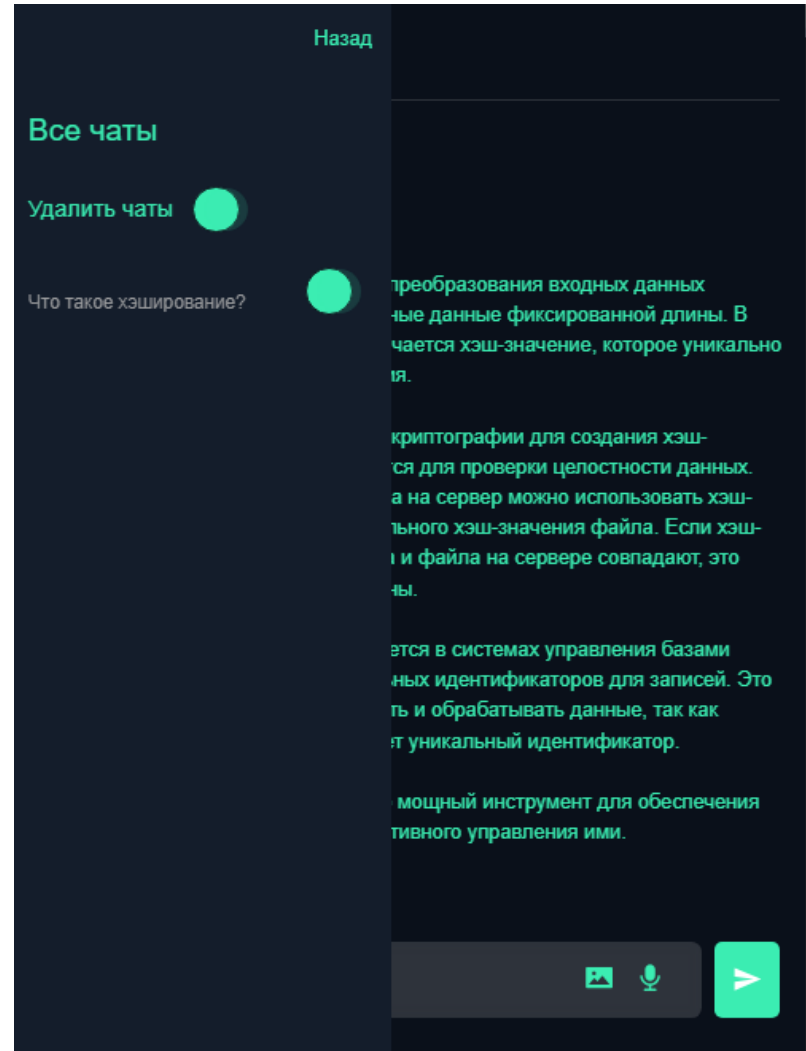
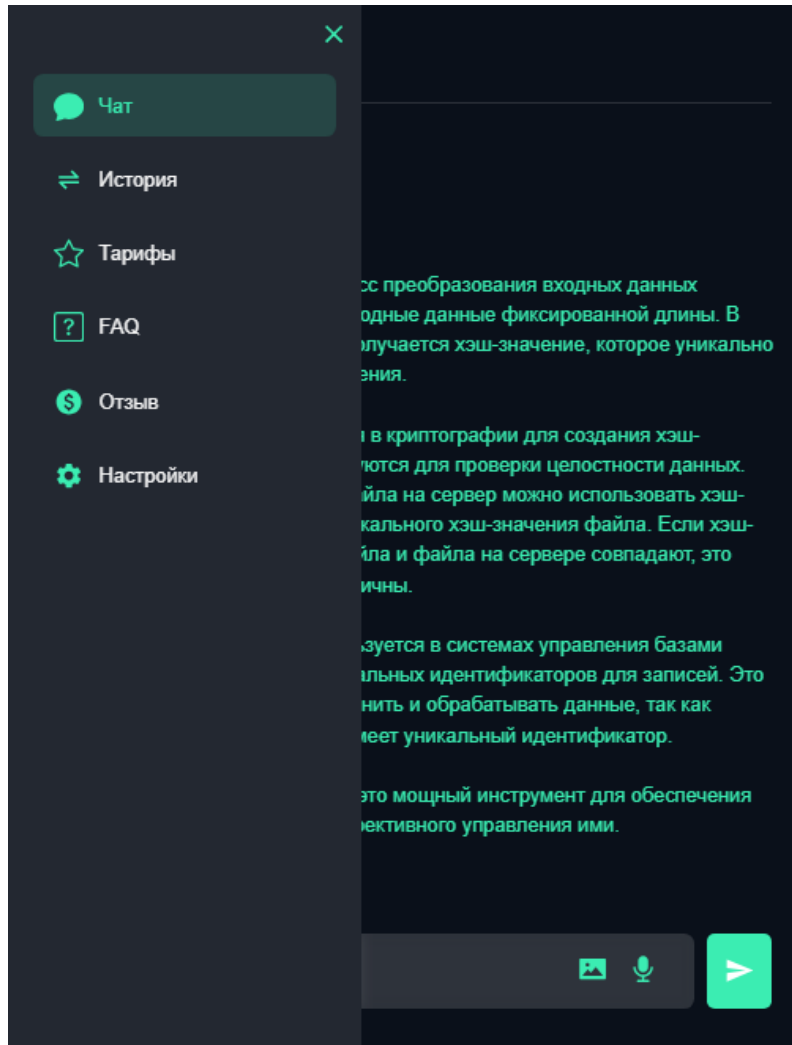
Изменить пароль

Новый пароль: Новый пароль

Подтвердить пароль: Подтвердите пароль

Обновить

Адаптивная версия



Функциональное тестирование

- 1) Подключение крипто-кошелька
- 2) Регистрация пользователя
- 3) Авторизация пользователя в системе
- 4) Авторизация с некорректными данными
- 5) Отправка запроса
- 6) Добавление записей
- 7) Выбор записи
- 8) Удаление записей
- 9) Приобретение тарифа
- 10) Отправка отзыва
- 11) Редактирование учетных данных
- 12) Выход пользователя из системы

Основные результаты

1. Произведен обзор существующих приложений по предметной области
2. Проведен анализ предметной области
3. Выполнено проектирование веб-приложения
4. Реализованы функциональные возможности веб-приложения
5. Проведено тестирование разработанного веб-приложения на тестовой сети Metamask