

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Расчётно-графическое задание
По дисциплине: Технологии Web-программирования
Тема: «Сервис для облачного машинного обучения»

Выполнил:
студент группы ПВ-192
Бочков Андрей
Проверил:
Картамышев С.В.

Белгород 2022

Оглавление

HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS	3
<i>Главная страница</i>	3
<i>Контентная страница</i>	3
<i>Страница авторизации</i>	4
Клиентское программирование	4
<i>Главная страница</i>	4
<i>Контентная страница</i>	5
<i>Страница авторизации</i>	7
Разработка и проектирование базы данных	9
<i>Схема базы данных в нотации IDEF1X</i>	9
<i>Код модели Операции</i>	9
REST API.....	10
<i>Схема API (в Postman)</i>	10
Работа с HTTP запросами.....	13
<i>Код запроса</i>	13
<i>Результат работы запроса</i>	14

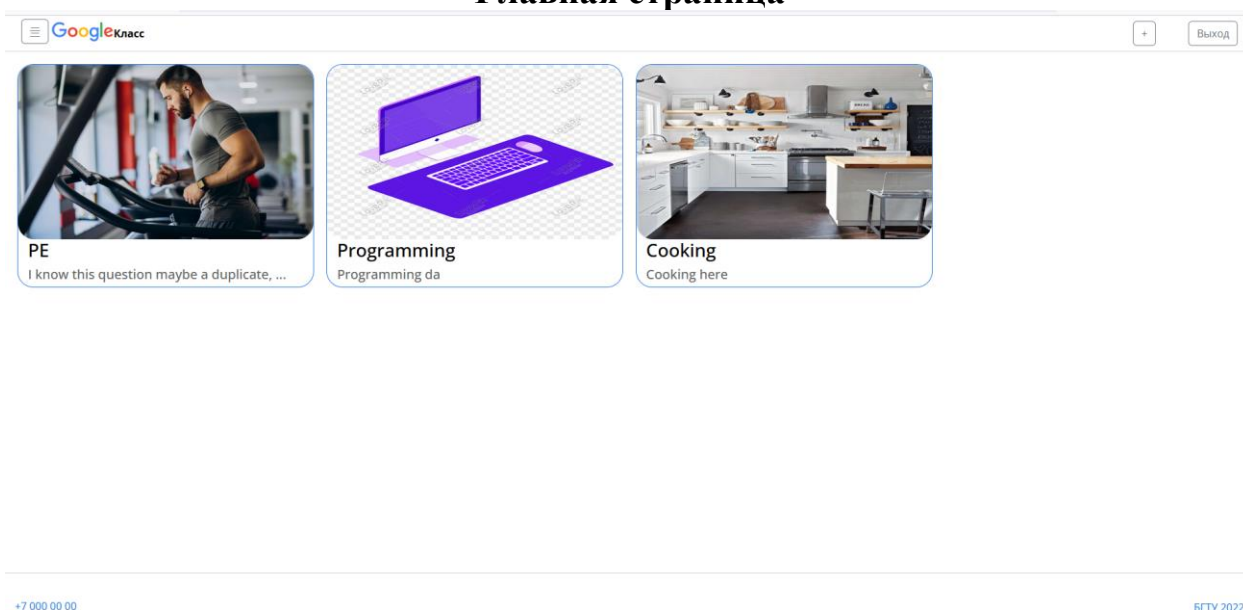
Цель работы: создать web приложение с сайтом и сервером для сервиса с курсами на различную тематику, с возможностью подписываться и отписываться от этих курсов.

Ход работы

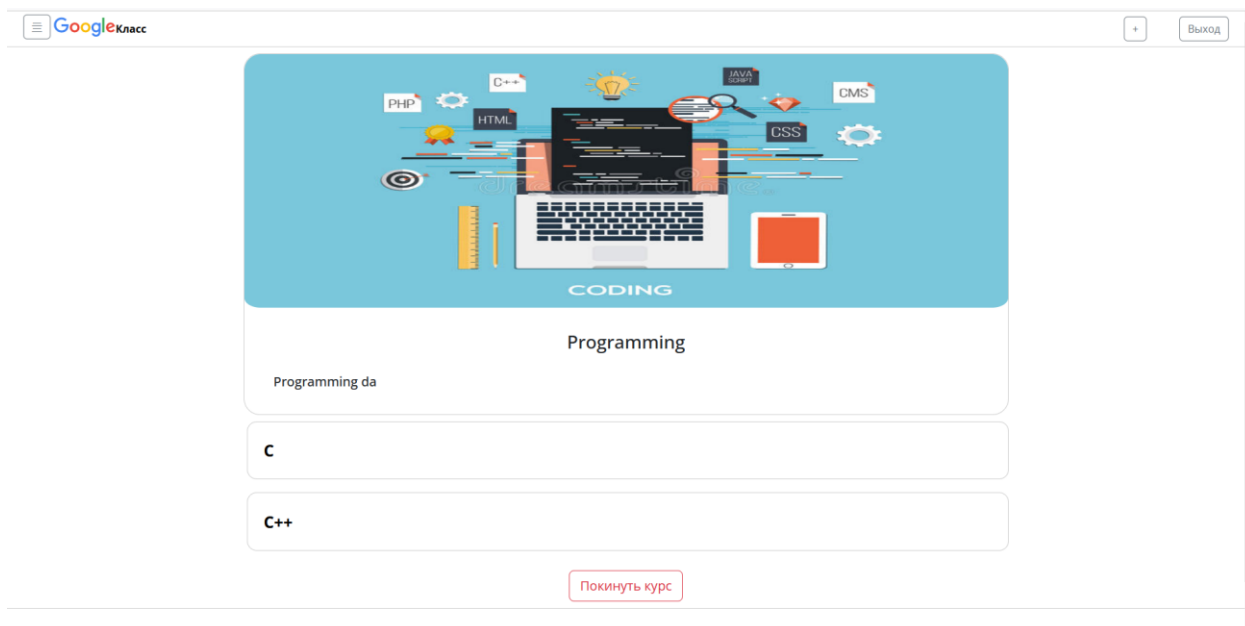
HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS

1. Выбрать шаблон или разработать свой для web-приложения. Шаблон должен включать минимум страницы (главная, контентная страница, страница авторизации/регистрации).
2. Сделать макеты страниц с помощью языка разметки HTML.
3. Стилизовать страницы с помощью языка CSS.
4. Продемонстрировать внешний вид разработанных страниц.

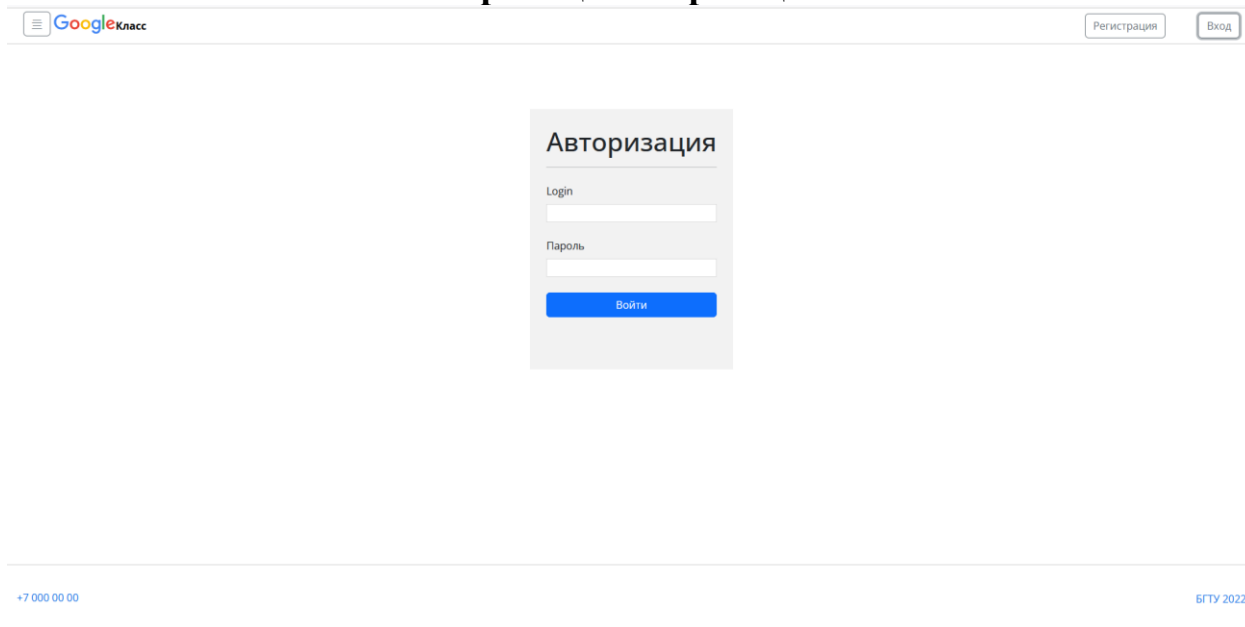
Главная страница



Контентная страница



Страница авторизации



1. Изучить основы разработки на языке JavaScript.
2. Изучить основы разработки frontend-приложения.
3. Развернуть базовое приложение на фреймворке.
4. Добавить необходимые компоненты и перенести в них вёрстку, сделанную в прошлой лабораторной работе.
5. Продемонстрировать работу web-приложения.

Клиентское программирование

Главная страница

```
import React, { useEffect } from 'react';
import '../css/home.css';
import useContext from '../components/authContext';
import CursBaner from '../components/cursBaner';
```

```

import { Link } from "react-router-dom";
import CursData from '../models/CursData';
import {getCurs} from '../apis/cursApi';

function Home() {

  const [cursData, setCursData] = React.useState<CursData[]>([]);

  const isLogin = React.useContext(accContext);

  useEffect( () =>{
    if (isLogin?.auth === true){
      getCurs(setCursData);
    }

  }, [isLogin?.auth])

  return (
    <div>
      <main className="cont">
        {isLogin?.auth === false &&
          <div className='aa text-center'>
            
            <Link className="home-link" to="/login"><button type="submit"
className="btn btn-primary w-20 ">Войти в аккаунт</button></Link>
          </div>
        }
        {isLogin?.auth === true &&
          <section className="curs-list">
            {cursData.map(curs => <CursBaner {...curs} key={curs.id} />)}
          </section>
        }
      </main>
    </div>
  )
}

export default Home;

```

Контентная страница

```

import React, { useEffect } from 'react';
import '../css/curs.css';
import { Link } from "react-router-dom";
import { useParams } from 'react-router-dom'
import CursData from '../models/CursData'

```

```

import LectionsData from '../models/LectionsData'
import {getCurs} from '../apis/cursApi';
import {getLections} from '../apis/lectionsApi';
import CursLentaPart from '../components/cursLentaPart';
import { cursSet } from "../apis/cursSetApi";
import { useNavigate, NavigateFunction } from "react-router-dom";
import accContext from "../components/authContext";

const Curs = () => {

  const navigate: NavigateFunction = useNavigate();
  const isLogin = React.useContext(accContext);
  const {cursId} = useParams()
  const [cursData, setCursData] = React.useState<CursData>({
    id: 0,code:'',name:'',descr:'', preview_img:'', main_img:'', author:'',
date:''});
  const [lectionData, setLectionsData] = React.useState<LectionsData[]>([]);
  const [cursCode, setCursCode] = React.useState<string>('');
  const [flagCurs, setFlagCurs] = React.useState<number>(-1);

  useEffect( () =>{
    getCurs(setCursData, cursId)
  }, [cursId])

  useEffect( () =>{
    getLections(setLectionsData,'', cursId)
  }, [cursId])

  useEffect( () =>{
    if (flagCurs===1){
      const setCurs = async () => {
        const temp = await cursSet(cursCode, 0, isLogin, navigate)
        setFlagCurs(-1)
        navigate("/buff")
      }
      setCurs()
    }
  }, [cursCode, isLogin, navigate, flagCurs])

  function handleSubmit(event: any) {
    event.preventDefault()
    setCursCode(cursData.code)
    setFlagCurs(1)
  };

  return (
    <div className="container">
      <div className="title-all">

```

```

        <img className="curs-logo" src={cursData.main_img} alt="curs-main-
img" />
        <div className="title">
            <h3> {cursData.name} </h3>

        </div>
        <div className='descr'><h5>{cursData.descr}</h5></div>
    </div>

    <section className="lenta">
        {lectionData.map(lections => <CursLentaPart {...lections}
key={lections.id} />)}

        {isLogin?.auth === true &&
        <div className="leave-button"> <button type="button" onClick={(event) =>
handleSubmit(event)} className="btn btn-outline-danger btn-lg">Покинуть
курс</button></div>
        }
    </section>
</div>
)
}

export default Curs;

```

Страница авторизации

```

import React from 'react';
import '../css/login.css';
import authContext from "../components/authContext";
import { useNavigate, NavigateFunction } from "react-router-dom";
//import { User } from "../models/UsersData";
import { postLogin } from "../apis/loginApi";

export interface LoginValues {
    username: string,
    password: string
}

function Login() {

    const navigate: NavigateFunction = useNavigate();
    const [login, setLogin] = React.useState<string>('');
    const [password, setPassword] = React.useState<string>('');
    const [isPressed, setPressed] = React.useState<boolean>(false);
    const acc_context = React.useContext(authContext);

```

```

function handleSubmit(event: any) {
  event.preventDefault()
  setPressed(true)
  if (login !== '' && password !== '') {
    const userInfo = {} as LoginValues
    userInfo.username = login
    userInfo.password = password
    postLogin(userInfo, acc_context)
    navigate('/')
  }
};

function handleLogin(event: any) {
  event.preventDefault()
  setLogin(event.target.value.trim())
}

function handlePassword(event: any) {
  event.preventDefault()
  setPassword(event.target.value.trim())
}

return (
  <div className="login-form">
    <h1>Авторизация</h1>
    <hr />
    <form className="mt-4" >
      <div className="mb-4">
        <div><label>Login</label></div>
        <div><input type="text" className="w-100" name="email" required
onChange={evt => handleLogin(evt)} /></div>
        {(isPressed && login === '') && <div className="errors">Заполните
"Логин"</div>}
      </div>
      <div className="mb-4">
        <div><label>Пароль</label></div>
        <div><input type="text" className="w-100" name="password" required
onChange={evt => handlePassword(evt)} /></div>
        {(isPressed && password === '') && <div className="errors">Заполните
"Пароль"</div>}
      </div>
      <div className="mb-4">
        <button type="submit" className="btn btn-primary w-
100" onClick={(event) => handleSubmit(event)} >Войти</button>
      </div>
    </form>
  </div>
)
}

```



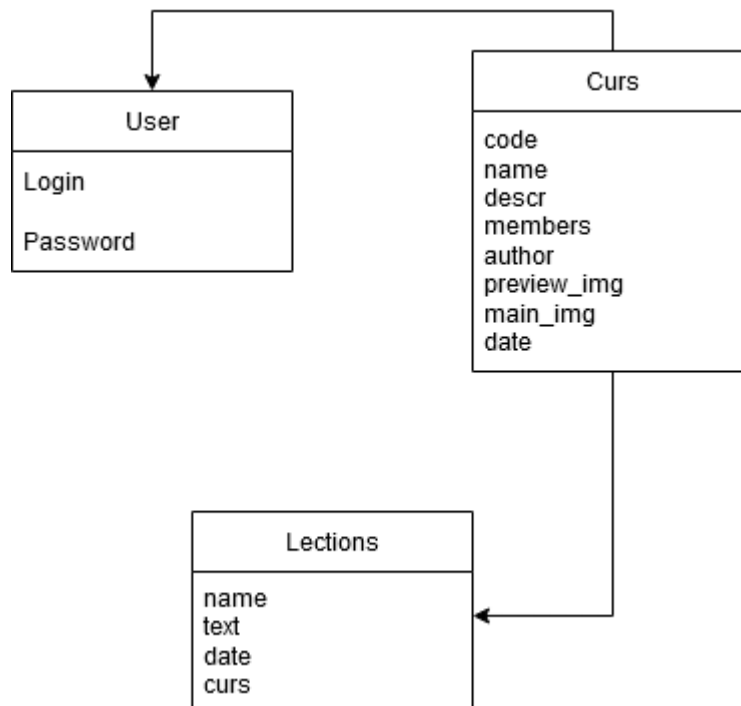
```
export default Login;
```

Разработка и проектирование базы данных

1. Выбрать подходящую СУБД.
2. Изучить методы взаимодействия web-приложения с базой данных.
3. Разработать структуру базы данных.
4. Разработать соответствующие модели в приложении.
5. В отчёт приложить схему базы данных, а так же код одной из моделей (на своё усмотрение).

СУБД - PostgreSQL

Схема базы данных в нотации IDEF1X



Код модели Операции

```
class Curs(models.Model):
    code = models.CharField(max_length=8, unique=True, null=False)
    name = models.CharField(max_length=100, default="name")
    descr = models.TextField()
    members = models.ManyToManyField(User, related_name='curs_members')
    author = models.ForeignKey(User, on_delete=models.CASCADE,
to_field='username', default='admin', related_name='curs_author')
    preview_img = models.URLField(max_length=500,
default='https://img.lovepik.com/original_origin_pic/19/01/08/5016ef557fa043da013
71d7a86300fa3.png_wh860.png')
```

```

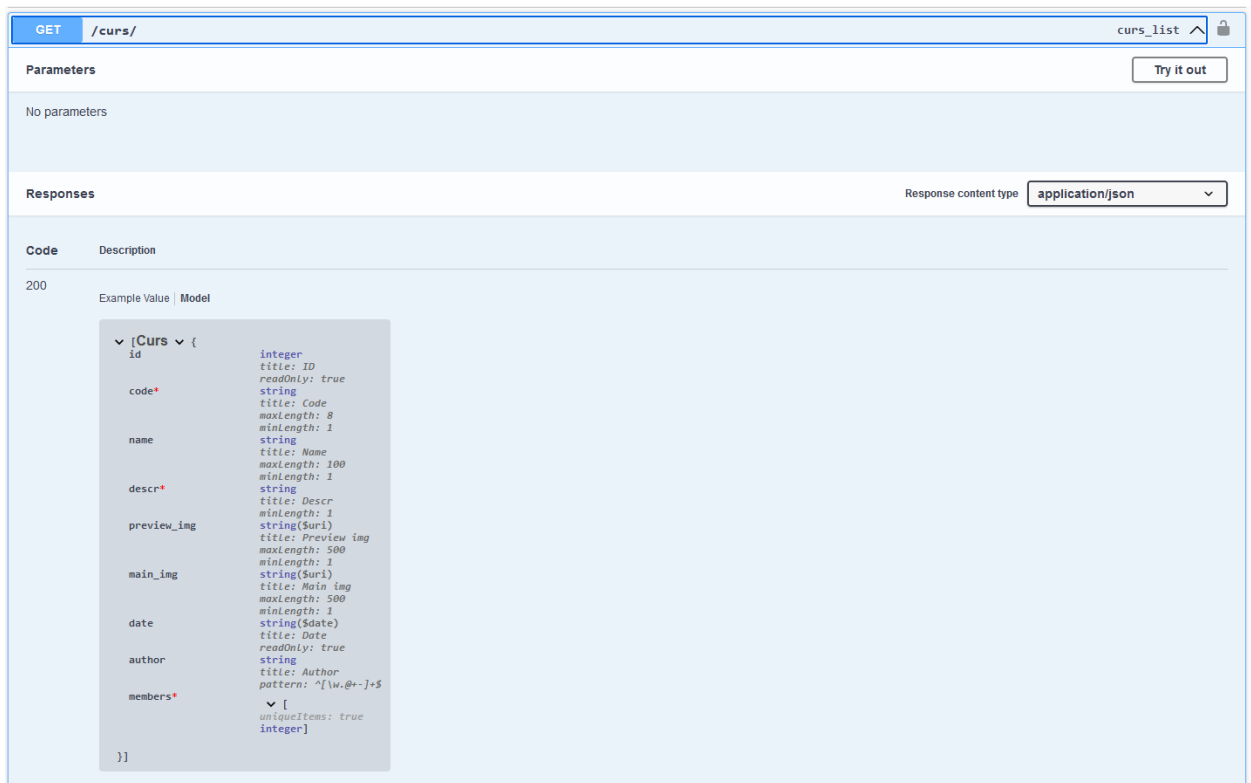
main_img = models.URLField(max_length=500,
default='https://thumbs.dreamstime.com/b/programaci%C3%B3n-codificaci%C3%B3n-de-
concepto-plano-54998068.jpg')
date = models.DateField(null=False, auto_now_add=True)

```

REST API

1. Изучить структуру формата представления данных JSON.
2. Изучить типы запросов к API: HEAD, GET, POST, PUT, DELETE.
3. Спроектировать и реализовать собственное REST API (Получение, создание, изменение и удаление каких-либо объектов).
4. В отчёт необходимо предоставить документацию к использованию методов. (Либо словесным описанием, либо через Swagger)

Схема API (в Postman)



GET /curs/ curs_list

Parameters: No parameters

Responses: Response content type: application/json

Code: 200 Description: Example Value | Model

```

{
  "Curs": [
    {
      "id": 1,
      "code": "C001",
      "name": "Curs 1",
      "descr": "Descripción del curso 1",
      "preview_img": "https://example.com/preview_img.jpg",
      "main_img": "https://example.com/main_img.jpg",
      "date": "2023-01-01",
      "author": "Autor del curso 1",
      "members": [
        {
          "id": 1,
          "name": "Miembro 1"
        },
        {
          "id": 2,
          "name": "Miembro 2"
        }
      ]
    }
  ]
}

```


GET

/lections/{id}/

lections_read

Parameters

Try it out

Name	Description
id * required string (path)	<input type="text" value="id"/>

Responses

Response content type application/json

Code	Description
200	<div><div>Example Value Model</div><div><div>Llections</div><div><div><div>id</div><div>integer title: ID readOnly: true</div><div>string title: Name maxLength: 100 minLength: 1</div><div>string title: Text minLength: 1 string(\$date) title: Date readOnly: true</div><div>string title: Curs</div></div></div></div></div>

PATCH

/lections/{id}/

lections_partial_update

Parameters

Try it out

Name	Description
data * required object (body)	<div><div>Example Value Model</div><div><div>Llections</div><div><div><div>name</div><div>string title: Name maxLength: 100 minLength: 1</div><div>string title: Text minLength: 1</div><div>string title: Curs</div></div></div></div></div>
id * required string (path)	<input type="text" value="id"/>

Responses

Response content type application/json

Code	Description
200	<div><div>Example Value Model</div><div><div>Llections</div><div><div><div>id</div><div>integer title: ID readOnly: true</div><div>string title: Name maxLength: 100 minLength: 1</div><div>string title: Text minLength: 1 string(\$date) title: Date readOnly: true</div><div>string title: Curs</div></div></div></div></div>

Работа с HTTP запросами

1. Изучить возможности js для отправки http запросов.
2. Выбрать подходящую библиотеку для работы с запросами.
3. Реализовать взаимодействие фронтенда с REST API, спроектированном в прошлой лабораторной работе.
4. Продемонстрировать работу взаимодействия фронтенд приложения с REST API.

Код запроса

```
import Axios, {AxiosError, AxiosResponse} from 'axios';
import { baseUrl } from '../constants';
import CursData from '../models/CursData';

const cursPath = baseUrl + '/curs/'

export function parseJwt (token: any) {
  const base64Url = token.split('.')[1];
  const base64 = base64Url.replace(/-/g, '+').replace(/_/g, '/');
  const jsonPayload =
decodeURIComponent(window.atob(base64).split('').map(function(c) {
  return '%' + ('00' + c.charCodeAt(0).toString(16)).slice(-2);
}).join(''));
  return JSON.parse(jsonPayload);
}

export async function getCurs(resultHandler: (data: any)=>void, id:string='') {
  if (id.length){
    Axios.get(cursPath+id+'/',
      { params:{}, responseType: "json" }
    ).then
    (result => {
      let data: CursData[] = (result as AxiosResponse<CursData[]>).data;
      resultHandler(data);
    })
    .catch((error: AxiosError) => {
      alert(error.message);
    });
  }
  else{
    const user_name = parseJwt(localStorage.getItem('access')).name
    Axios.get(
      cursPath,
      { params:{username:user_name}, responseType: "json" }
    ).then
    (result => {
      const data: CursData[] = (result as AxiosResponse<CursData[]>).data;
      resultHandler(data)
    })
    .catch((error: AxiosError) => {
      alert(error.message);
    });
  }
}
```

Результат работы запроса



PE

I know this question maybe a duplicate, ...




Programming

Programming da



Cooking

Cooking here



CODING

Programming

Programming da

C

C++

Покинуть курс