

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет телекоммуникаций и  
информатики»

Кафедра телекоммуникационных систем и вычислительных средств  
(ТС и ВС)

ОТЧЕТ  
по дисциплине  
«WEB - Технологии»

по теме:  
РГЗ

Студент:  
*Группа: ИКС - 431*

*П.С. Волевач*

Предподаватель:

*А.В. Андреев*

Новосибирск 2025 г.

## СОДЕРЖАНИЕ

1	BACKEND .....	3
1.1	Django .....	3
2	FRONTEND .....	10
2.1	REACT .....	10
2.2	components/Header.js .....	11
2.3	components/VideoCard.js .....	16
2.4	pages/Account.js .....	18
2.5	pages/Home.js .....	25
2.6	pages/Login.js .....	29
2.7	pages/Upload.js .....	37
2.8	pages/VideoPage.js .....	46
2.9	App.js .....	48

# 1 BACKEND

## 1.1 Django

Я наину разрабатывать проект на компьютере (клиенте) в будущем перенесу его на сервер.

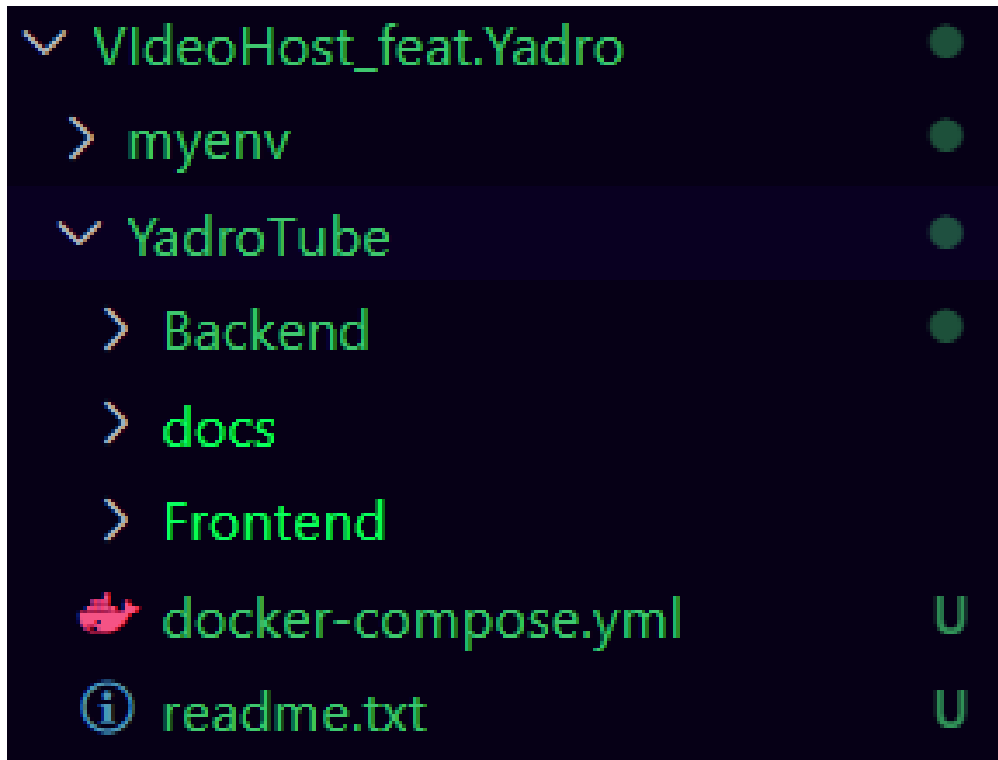


Рисунок 1 — общая структура проекта

Стоит обратить внимание на то, что, если используется WSL, может потребоваться виртуальное окружение для команды `pip`.

```
● (myenv) felone@HOME-PC:~/studying/VideoHost_feat.Yadro/Backend$ django-admin startproject YadroTube
● (myenv) felone@HOME-PC:~/studying/VideoHost_feat.Yadro/Backend$ cd YadroTube/
```

Рисунок 2 — устанавливаем Django

далее создаем приложение для API этой командой `python manage.py startapp api`.

В `settings.py` найдем блок `DATABASES` и изменим настройки для PostgreSQL.

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'mydatabase',
        'USER': 'mydatabaseuser',
        'PASSWORD': 'mypassword',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}

```

Рисунок 3 — меняем настройки settings.py

Создадим модель для видео, то есть некий шаблон для базы данных.

```

VideoHost_featYadro > YadroTube > Backend > YadroTube > api > models.py
1  # Импорт необходимых модулей Django
2  from django.db import models
3  from django.contrib.auth.models import User # Стандартная модель пользователя
4
5  # Модель для хранения информации о видео
6  class Video(models.Model):
7      """Модель представляет видеофайл и связанные с ним данные"""
8
9      # Заголовок видео: строка до 200 символов
10     title = models.CharField(
11         max_length=200,
12         verbose_name='Название видео'
13     )
14
15     # Подробное описание видео: текст без ограничения длины
16     description = models.TextField(
17         verbose_name='Описание видео'
18     )
19
20     # Поле для загрузки видеофайла
21     # Файлы сохраняются в подпапку videos/ медиа-директории
22     file = models.FileField(
23         upload_to='videos/',
24         verbose_name='Видеофайл'
25     )
26
27     # Связь с автором видео (пользователем)
28     # При удалении пользователя удаляются все связанные видео (CASCADE)
29     author = models.ForeignKey(
30         User,
31         on_delete=models.CASCADE,
32         verbose_name='Автор'
33     )
34
35     # Дата и время загрузки видео
36     # Автоматически устанавливается при создании записи
37     uploaded_at = models.DateTimeField(
38         auto_now_add=True,
39         verbose_name='Дата загрузки'
40     )
41
42     # Строковое представление объекта (для админки и отладки)
43     def __str__(self):
44         return self.title

```

Рисунок 4 — /api/models.py

```

(myenv) felone@HOME-PC:~/studing/VideoHost_feat.Yadro/YadroTube/Backend/YadroTube$ python manage.py makemigrations
Traceback (most recent call last):
  File "/home/felone/studing/VideoHost_feat.Yadro/myenv/lib/python3.12/site-packages/django/db/backends/postgresql/base.py", line 25, in <module>
    import psycopg as Database
ModuleNotFoundError: No module named 'psycopg'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/felone/studing/VideoHost_feat.Yadro/myenv/lib/python3.12/site-packages/django/db/backends/postgresql/base.py", line 27, in <module>
    import psycopg2 as Database
ModuleNotFoundError: No module named 'psycopg2'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/felone/studing/VideoHost_feat.Yadro/YadroTube/Backend/YadroTube/manage.py", line 22, in <module>
    main()
  File "/home/felone/studing/VideoHost_feat.Yadro/YadroTube/Backend/YadroTube/manage.py", line 18, in main
    execute_from_command_line(sys.argv)
  File "/home/felone/studing/VideoHost_feat.Yadro/myenv/lib/python3.12/site-packages/django/core/management/__init__.py", line 442, in execute_from_command_line
    utility.execute()
  File "/home/felone/studing/VideoHost_feat.Yadro/myenv/lib/python3.12/site-packages/django/core/management/__init__.py", line 416, in execute
    django.setup()
  File "/home/felone/studing/VideoHost_feat.Yadro/myenv/lib/python3.12/site-packages/django/__init__.py", line 24, in setup
    apps.populate(settings.INSTALLED_APPS)
  File "/home/felone/studing/VideoHost_feat.Yadro/myenv/lib/python3.12/site-packages/django/apps/registry.py", line 116, in populate
    app_config.import_models()
  File "/home/felone/studing/VideoHost_feat.Yadro/myenv/lib/python3.12/site-packages/django/apps/config.py", line 269, in import_models

```

Рисунок 5 — создадим миграции

видим кучу ошибок, стоит установить postgresql если он не был установлен,и создать нового пользователя, использовать супер пользователя не желательно(не безопасно)

подключаемся к базе данных sudo -u postgres psql

теперь создадим нового пользователя CREATE USER djangouser WITH PASSWORD 'ваш\_надёжный\_пароль';

создадим базу данных CREATE DATABASE mydatabase OWNER djangouser;

выдадим права GRANT ALL PRIVILEGES ON DATABASE mydatabase TO djangouser;

```

(myenv) felone@HOME-PC:~/studing/VideoHost_feat.Yadro/YadroTube/Backend/YadroTube$ python manage.py makemigrations
No changes detected
(myenv) felone@HOME-PC:~/studing/VideoHost_feat.Yadro/YadroTube/Backend/YadroTube$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(myenv) felone@HOME-PC:~/studing/VideoHost_feat.Yadro/YadroTube/Backend/YadroTube$

```

Рисунок 6 — создадим миграции

теперь все успешно

установим JWT pip install djangorestframework djangorestframework-simplejwt django-cors-headers

теперь вернемся к settings.py

добавляем следующее:

```
INSTALLED_APPS = [  
    ...  
    'rest_framework',  
    'rest_framework_simplejwt',  
    'corsheaders',  
    'api',  
]  
  
MIDDLEWARE = [  
    ...  
    'corsheaders.middleware.CorsMiddleware',  
]  
  
REST_FRAMEWORK = {  
    'DEFAULT_AUTHENTICATION_CLASSES': (  
        'rest_framework_simplejwt.authentication.JWTAuthentication',  
    )  
}  
  
CORS_ALLOW_ALL_ORIGINS = True
```

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'rest_framework_simplejwt',
    'corsheaders',
    'api',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'corsheaders.middleware.CorsMiddleware',
]

REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework_simplejwt.authentication.JWTAuthentication',
    )
}

CORS_ALLOW_ALL_ORIGINS = True

```

Рисунок 7 — settings.py после изменений

перейдем к созданию api

сначала напишем сериализатор в файле api/serializers.py, можно представить его так:

```

from rest_framework import serializers
from .models import Video

```

```

class VideoSerializer(serializers.ModelSerializer):

```

```

    class Meta:

```

```

        model = Video

```

```

        fields = ['id', 'title', 'description', 'file', 'author', 'up

```

теперь перейдем к представлению в api/views.py

```

from rest_framework import viewsets

```

```

from .models import Video
from .serializers import VideoSerializer
from rest_framework.permissions import IsAuthenticatedOrReadOnly

class VideoViewSet(viewsets.ModelViewSet):
    queryset = Video.objects.all()
    serializer_class = VideoSerializer
    permission_classes = [IsAuthenticatedOrReadOnly]

    def perform_create(self, serializer):
        serializer.save(author=self.request.user)

```

теперь пропишем пути в api/urls.py

```

from django.urls import path, include
from rest_framework.routers import DefaultRouter
from . import views

router = DefaultRouter()
router.register(r'videos', views.VideoViewSet)

urlpatterns = [
    path('', include(router.urls)),
]

```

остается настроить JWT в videoplatform/urls.py

```

from django.urls import path, include
from rest_framework_simplejwt.views import TokenObtainPairView, TokenRefreshView

urlpatterns = [
    ...
    path('api/auth/', include('rest_framework.urls')),
    path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    path('api/token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
]

```



```
] path('api/', include('api.urls')),
```

теперь можно попробовать запустить сервер `python manage.py runserver`

```
(myenv) felone@HOME-PC:~/studying/VideoHost_feat.Yadro/YadroTube/Backend/YadroTube$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 19, 2025 - 19:34:26
Django version 5.2, using settings 'YadroTube.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/
```

Рисунок 8 — запуск api сервера

Теперь API доступно по адресу `http://localhost:8000/api/videos/`  
вот что можно увидеть если перейти по этому адресу

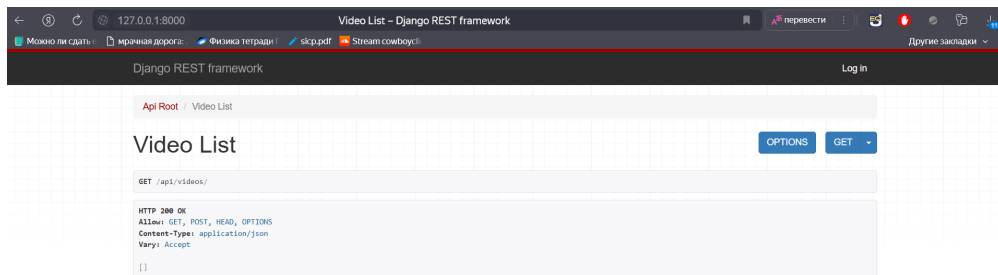


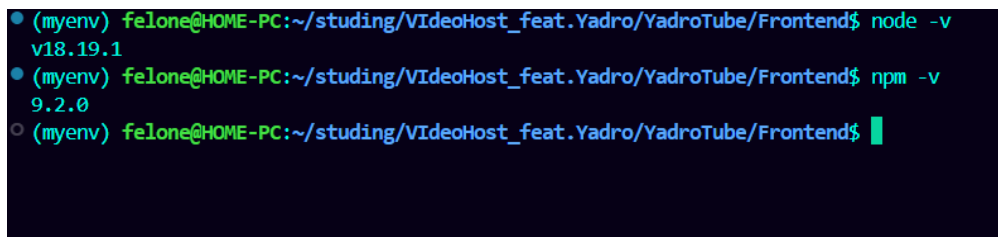
Рисунок 9 — api

## 2 FRONTEND

### 2.1 REACT

сначала проверим установлен ли node.js и npm командами

```
node -v
npm -v
```

A terminal window with a dark background. The prompt is (myenv) felone@HOME-PC:~/studing/Videohost\_feat.Yadro/YadroTube/Frontend\$. The first command is node -v, which outputs v18.19.1. The second command is npm -v, which outputs 9.2.0. The prompt is now (myenv) felone@HOME-PC:~/studing/Videohost\_feat.Yadro/YadroTube/Frontend\$.

```
(myenv) felone@HOME-PC:~/studing/Videohost_feat.Yadro/YadroTube/Frontend$ node -v
v18.19.1
(myenv) felone@HOME-PC:~/studing/Videohost_feat.Yadro/YadroTube/Frontend$ npm -v
9.2.0
(myenv) felone@HOME-PC:~/studing/Videohost_feat.Yadro/YadroTube/Frontend$
```

Рисунок 10 — просмотр версий

далее стоит обозначить некую структуру

frontend/

- | — public/ # Статические файлы (HTML, изображения)
  - | | — index.html
- | — src/
  - | | — components/ # Переиспользуемые компоненты (кнопки, карточ-ки)
  - | | — pages/ # Страницы (логин, просмотр видео)
  - | | — contexts/ # Контексты для глобального состояния
  - | | — App.js # Главный компонент
  - | | — index.js # Точка входа
  - | | — styles/ # Стили (CSS/SCSS)
- | — package.json # Зависимости и скрипты
- | — .env # Переменные окружения (API URL)

в frontend/ создадим приложение командой:

```
npx create-react-app frontend
cd frontend
npm start # http://localhost:3000
```

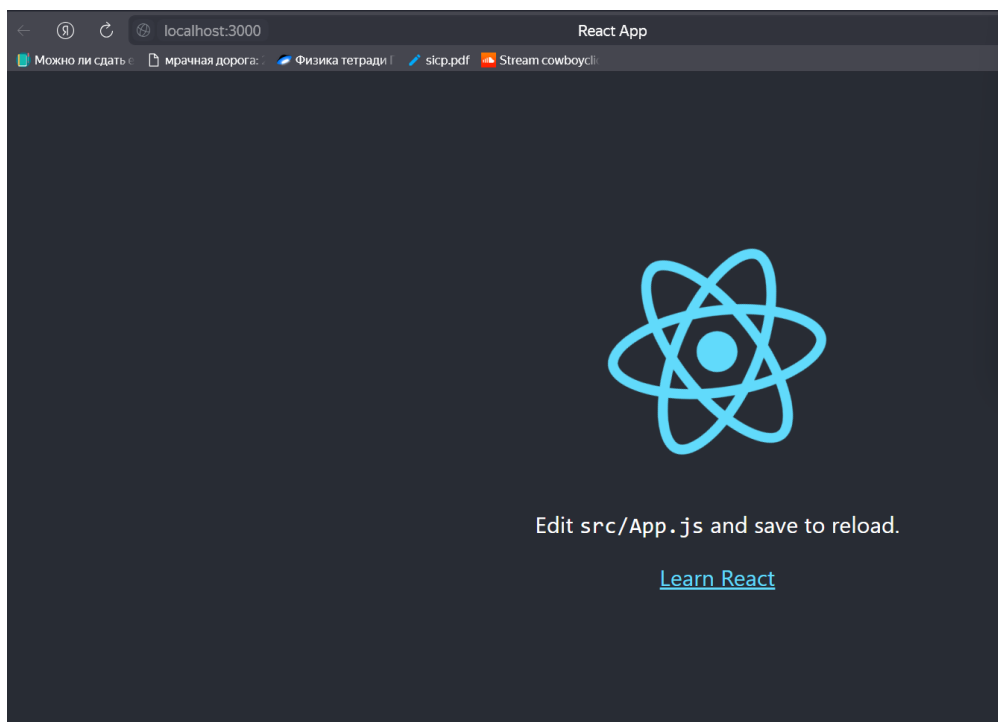


Рисунок 11 — подтверждение работы сервера

## 2.2 components/Header.js

теперь уюгда у нас есть шаблон можно приступить к разработке проекта

создадим Header.js с головой сайта котрую мы будем использовать на всех страницах

```
// src/components/Header.js
import React from 'react';
import { Link, useNavigate } from 'react-router-dom'; // <--
import logo from './logo.png'; // <--

const Header = () => {
  const navigate = useNavigate(); // <-- useNavigate

  //
  const handleLogout = () => {
    // access_token refresh_token localStorage
```

```

localStorage.removeItem('access_token');
localStorage.removeItem('refresh_token'); // refresh

// ( )
navigate('/'); // navigate('/login');
// : -
// JWT.
// , axios.post('/api/logout/',
};

// , ( access_token localStorage)
const isLoggedIn = localStorage.getItem('access_token');

return (
  <header
    style={{
      backgroundColor: '#343a40', // -
      padding: '15px 20px', // /
      color: 'white', //
      position: 'fixed', //
      width: '100%', //
      top: '0', //
      zIndex: '1000', // z-index
      boxShadow: '0 2px 4px rgba(0,0,0,0.2)', //
      display: 'flex', // flexbox
      justifyContent: 'space-between', //
      alignItems: 'center', //
    }}
  >
    {/* --- ( ) --- */}
    <div style={{ display: 'flex', alignItems: 'center' }}>
      {/* */}
      <Link to="/" style={{ textDecoration: 'none' }}>
        <img
          src={logo} // -

```

```

        alt="      " //
        style={{ height: '30px', marginRight: '15px' }} /
    />
</Link>
    {/*
        ,
    */}
    {/* <span style={{ color: 'white', fontSize: '1.2rem', fo
</div>
    {/* ---
        --- */}

{/* ---
    (    ) --- */}
<nav
    style={{
        display: 'flex',
        alignItems: 'center',
        gap: '25px',
        //
        //      maxWidth      margin: '0 auto'      nav,      header
    }}
>
    {/*
        (
        ,
    ) */}
    {/*
        ,
    */}
    <Link
        to="/"
        style={linkStyle} //
        onMouseOver={(e) => e.target.style.color = '#007bff'} //
        onMouseOut={(e) => e.target.style.color = 'white'}
    >

    </Link>

    {/*
        :
    */}
    {isLoggedIn ? (
        // ---
        (    ) ---
        <> {/*
            Fragment
        */}
        {/*
            */}
        <Link

```

```

        to="/account"
        style={linkStyle}
        onMouseOver={(e) => e.target.style.color = '#007bff'}
        onMouseOut={(e) => e.target.style.color = 'white'}
    >

</Link>
{ /*          */ }
{ /*          button    onClick          ,          */ }
<button
    onClick={handleLogout} //          handleLogout
    style={logoutButtonStyle} //          ,
    onMouseOver={(e) => e.target.style.color = '#dc3545'} /
    onMouseOut={(e) => e.target.style.color = 'white'}
>

</button>
</>
) : (
// ---          (          ) ---
<> { /*          Fragment,          */ }
    { /*          */ }
    <Link
        to="/login"
        style={linkStyle}
        onMouseOver={(e) => e.target.style.color = '#007bff'}
        onMouseOut={(e) => e.target.style.color = 'white'}
    >

    </Link>
    { /*          :          ,          */ }
    { /* <Link to="/register" style={linkStyle}>          </Link>
    </>
    )}
</nav>

```

```

        { /* ---                      --- */ }
    </header>
  );
};

// ---                      (                      ) ---
const linkStyle = {
  color: 'white',           //
  textDecoration: 'none',   //
  fontSize: '16px',
  fontWeight: '600',
  transition: 'color 0.3s ease', //
  //      :      :hover inline      .
  //      onMouseOver/onMouseOut      .
};

const logoutButtonStyle = {
  ...linkStyle, //
  backgroundColor: 'transparent', //
  border: 'none', //
  cursor: 'pointer', //
  padding: 0, //
  font: 'inherit', //
  //      "      ",
  //      ,      .
};
// ---      ---

```

```
export default Header;
```

прошу обратить внимание, использовать css стили прямо в коде js совсем не обязательно, их можно вынести в отдельный файл

## 2.3 components/VideoCard.js

добавим еще один компонент VideoCard.js который будет неким шаблоном карточки с видео

```
import React from 'react';
import { Link } from 'react-router-dom';

const VideoCard = ({ id, title, description, thumbnail }) => {
  return (
    <div className="video-card" style={styles.card}>
      <Link to={`~/video/${id}`} style={styles.thumbnailLink}>
        <img
          src={thumbnail || 'https://via.placeholder.com/320x180'}
          alt={title}
          style={styles.thumbnail}
        />
      </Link>
      <div style={styles.content}>
        <h3 style={styles.title}>
          <Link to={`~/video/${id}`} style={styles.titleLink}>{title}</Link>
        </h3>
        <p style={styles.description}>{description}</p>
      </div>
    </div>
  );
};

const styles = {
  card: {
    border: '1px solid #e1e1e1',
    borderRadius: '8px',
    overflow: 'hidden',
    margin: '10px',
    backgroundColor: '#fff',
  },
  thumbnailLink: {
    display: 'block',
  },
  thumbnail: {
    width: '100%',
    height: '100%',
  },
  content: {
    padding: '10px',
  },
  title: {
    margin: '0',
  },
  titleLink: {
    display: 'block',
  },
  description: {
    margin: '0',
  },
};
```



```

    boxShadow: '0 2px 4px rgba(0,0,0,0.1)',
    transition: 'transform 0.2s',
    ':hover': {
      transform: 'translateY(-2px)'
    }
  },
  thumbnailLink: {
    display: 'block',
    textDecoration: 'none'
  },
  thumbnail: {
    width: '100%',
    height: '180px',
    objectFit: 'cover',
    borderBottom: '1px solid #e1e1e1'
  },
  content: {
    padding: '15px'
  },
  title: {
    margin: '0 0 10px 0',
    fontSize: '1.1rem',
    color: '#333'
  },
  titleLink: {
    color: 'inherit',
    textDecoration: 'none',
    ':hover': {
      textDecoration: 'underline'
    }
  },
  description: {
    margin: '0',
    fontSize: '0.9rem',
    color: '#666',

```

```

        lineHeight: '1.4'
      }
    };

    export default VideoCard;

```

теперь, когда основные компоненты созданы можно перейти к созданию основных страниц

## 2.4 pages/Account.js

Account.js это страница с информацией о пользователе и выложенными им видео

```

// src/pages/Account.js
import React, { useState, useEffect } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import axios from 'axios';
import styles from './styles/Account.module.css'; // CSS
import backgroundImage from './background.png'; // <--

const Account = () => {
  const [userData, setUserData] = useState(null);
  const [userVideos, setUserVideos] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const navigate = useNavigate();

  useEffect(() => {
    // ---
    //
    const accessToken = localStorage.getItem('access_token');
    if (!accessToken) {
      //
      navigate('/login');
      return; // useEffect,
    }
  }, [navigate]);

```

```

}
// ---          ---

const fetchData = async () => {
  try {
    //          ,
    const [userResponse, videosResponse] = await Promise.all([
      axios.get('http://localhost:8000/api/auth/me/', {
        headers: {
          Authorization: `Bearer ${accessToken}` //
        }
      }),
      axios.get('http://localhost:8000/api/videos/user/', {
        headers: {
          Authorization: `Bearer ${accessToken}` //
        }
      })
    ]);

    setUserData(userResponse.data);
    setUserVideos(videosResponse.data);
    setError(null); //          ,

  } catch (err) {
    console.error('          :', err);
    setError('          .'); //

    //          401 (          )
    if (err.response && err.response.status === 401) {
      //          401,          ,
      localStorage.removeItem('access_token'); //
      // localStorage.removeItem('refresh_token'); //          ,
      navigate('/login'); //
    }
  }
}

```

```

    } finally {
      setLoading(false); //
    }
  };

  //
  fetchData();
}, [navigate]); //      navigate      useEffect

//
const handleDeleteVideo = async (videoId) => {
  if (!window.confirm('      ,      ?')) return;

  const accessToken = localStorage.getItem('access_token');
  if (!accessToken) {
    //
    navigate('/login');
    return;
  }

  try {
    await axios.delete(`http://localhost:8000/api/videos/${videoId}`
      headers: {
        Authorization: `Bearer ${accessToken}` //
      }
    });
    //
    setUserVideos(userVideos.filter(video => video.id !== videoId))
    //      :
    console.log(`      ${videoId}      .`);
  } catch (err) {
    console.error('      :', err);
    //      401
    if (err.response && err.response.status === 401) {
      localStorage.removeItem('access_token');
    }
  }
}

```

```

        // localStorage.removeItem('refresh_token');
        navigate('/login');
    } else {
        //      :
        alert('                .');
    }
}
};

//      :      ,
if (loading) {
    return <div className={styles.loading} style={{ textAlign: 'center'
}

if (error) {
    return <div className={styles.error} style={{ textAlign: 'center'
}

//      ,
return (
    //
    //      CSS      -
    <div className={styles.container} style={{
        //
        backgroundImage: `url(${backgroundImage})`,
        backgroundSize: 'cover',
        backgroundPosition: 'center',
        backgroundRepeat: 'no-repeat',
        backgroundAttachment: 'fixed', //
        minHeight: '100vh', //
        width: '100vw', //
        boxSizing: 'border-box', //      padding
        margin: 0, //
        overflowX: 'hidden', //

```

```

//
display: 'flex',
flexDirection: 'column',
alignItems: 'center', //
// justifyContent: 'flex-start', // (

//
// !!! : , (Header)
// padding border, .
// , .
paddingTop: '80px',

// : Account.module.css .container display/fl
// - , CSS ,
// - .

}}>
<div className={styles.profileSection} style={{
  backgroundColor: 'rgba(255, 255, 255, 0.8)', //
  padding: '20px',
  borderRadius: '8px',
  marginBottom: '20px', //
  width: '100%', //
  maxWidth: '800px', //
  boxSizing: 'border-box',
  // Account.module.css,
}}>
  <h2> </h2>
  {/* , userData null */}
  {userData && (
    <div className={styles.profileInfo}>
      <p> : {userData.username}</p>
      <p>Email: {userData.email}</p>
      <p> : {new Date(userData.date_joined).toLocaleDateS
    </div>

```

```

    })

    <Link to="/upload" className={styles.uploadButton}>

    </Link>
  </div>

  <div className={styles.videosSection} style={{
    backgroundColor: 'rgba(255, 255, 255, 0.8)', //
    padding: '20px',
    borderRadius: '8px',
    width: '100%',
    maxWidth: '800px', //
    boxSizing: 'border-box',
    // Account.module.css,
  }}>
    <h3>      ({userVideos.length})</h3>

    {userVideos.length === 0 ? (
      <p>          </p>
    ) : (
      <div className={styles.videoGrid}>
        {userVideos.map(video => (
          //      ,      video      'id'
          <div key={video.id} className={styles.videoCard}>
            <div className={styles.thumbnail}>
              {/*      ,      video      'thumbnail'      */
              {video.thumbnail && (
                <img src={video.thumbnail} alt={video.title || '
              })}
              {/*      :      ,      */}
              {!video.thumbnail && (
                <div style={{ width: '100%', height: '180px', b
              </div>
            }
          )}
        )}
      </div>
    )}
  </div>

```

```

    })
  </div>
  <div className={styles.videoInfo}>
    <h4>{video.title}</h4>
    {/*           ,           */}
    <p>{video.description ? (video.description.length >
  <div className={styles.meta}>
    {/*           ,           video           'views'   'uploaded_at
    <span>           : {video.views !== undefined ? video.v
    {/*           uploaded_at   created_at
    <span>           : {video.uploaded_at ? new Date(video.upl
  </div>
  <div className={styles.actions}>
    <button
      onClick={() => handleDeleteVideo(video.id)}
      className={styles.deleteButton}
    >

    </button>
    <Link
      to={` /video/${video.id}`}
      className={styles.watchButton}
    >

    </Link>
  </div>
</div>
</div>
  )})
</div>
  )}
</div>
</div>
);
};

```



```
export default Account;
```

## 2.5 pages/Home.js

Home.js это домашняя страница

```
// src/pages/Home.js
import React, { useEffect, useState } from 'react';
// import Header from '../components/Header'; // Header
import VideoCard from '../components/VideoCard';
import axios from 'axios';
import backgroundImage from './background.png'; // <--

const Home = () => {
  //
  const [videos, setVideos] = useState([]);
  //
  const [loading, setLoading] = useState(true);
  //
  const [error, setError] = useState(null);

  //   useEffect
  useEffect(() => {
    const fetchVideos = async () => {
      try {
        //
        true
        setLoading(true);
        //   GET-   API
        //   ,
        //
        //   /
        //
        .
      }
    }
  });
}
```

```

        const response = await axios.get('http://localhost:8000/api/v
        //
        setVideos(response.data);
        setError(null); //
    } catch (err) {
        console.error("                :", err);
        //
        setError('                .');
    } finally {
        //                (                )
        setLoading(false);
    }
};

//
fetchVideos();
}, []); //                ,

//                ,
if (loading) {
    return (
        //                , Header                (                , App.js)
        //                ,
        <div style={{ textAlign: 'center', marginTop: '80px' }}>
    );
}

//                ,
if (error) {
    return (
        //                , Header                (                , App.js)
        //                ,
        <div style={{ textAlign: 'center', marginTop: '80px', color: 'r
    );
}

```

```

//
return (
  // ---
  <div style={{
    //
    backgroundImage: `url(${backgroundImage})`, //
    backgroundSize: 'cover', //
    backgroundPosition: 'center', //
    backgroundRepeat: 'no-repeat', //
    backgroundAttachment: 'fixed', //

    //
    width: '100vw', //
    minHeight: '100vh', //
    margin: 0, //
    padding: 0, //
    boxSizing: 'border-box', // padding
    overflowX: 'hidden', //

    // Flexbox
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center', //
    // justifyContent: 'flex-start', // (
    f

    //
    // !!! : , (Header)
    // padding border. ,
    // padding ,
    // .
    paddingTop: '80px', // ,

  }}>
  {/* ---
  --- */}

```

```

<div style={{
  width: '100%', // ( 100vw)
  maxWidth: '1200px', //
  margin: '0 auto', //
  padding: '0 20px', // ,
  boxSizing: 'border-box', // padding
  // , padding-bottom,
  // paddingBottom: '20px',
}}>
  <h1 style={{ textAlign: 'center', marginBottom: '30px', color: 'white' }}>
    !
  </h1>

  {/* map */}
  {videos.length === 0 ? (
    <div style={{ textAlign: 'center', color: 'white', textShadow: '2px 2px 4px black' }}>
      (
        <div
          className="video-grid" // ,
          style={{
            display: 'grid',
            gridTemplateColumns: 'repeat(auto-fill, minmax(300px, 1fr))',
            gap: '20px',
            // padding: '20px' // maxWidth/margin div
          }}
          >{videos.map(video => (<VideoCard key={video.id} id={video.id} video={video} />))}
        </div> {/* */}
      )
    ) : (
      <div> {/* */}
    )
  )};
};

export default Home;

```

## 2.6 pages/Login.js

Login.js страница входа и регистрации аккаунта

```
import axios from 'axios';
import { useNavigate } from 'react-router-dom';
import React, { useState, useEffect } from 'react'; // <-- useEf
import Header from '../components/Header'; // Header

const Login = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [username, setUsername] = useState('');
  const [confirmPassword, setConfirmPassword] = useState('');
  const [isRegistering, setIsRegistering] = useState(false);
  const [error, setError] = useState('');
  const navigate = useNavigate();

  // --- useEffect ---
  //
  useEffect(() => {
    // , localStorage
    if (localStorage.getItem('access_token')) {
      // ( ),
      navigate('/');
    }
  }, [navigate]); // navigate

  // --- ---
  const handleError = (err) => {
    let errorMessage = ' ';
    if (err.response && err.response.data) {
      if (err.response.data.detail) {
        errorMessage = err.response.data.detail;
      } else if (typeof err.response.data === 'object') {
```

```

//                                DRF (      , {"username": ["..."]}
errorMessage = Object.entries(err.response.data)
    .map(([key, value]) => {
        //                                DRF
        const fieldName = {
            username: '      ',
            email: 'Email',
            password: '      ',
            password2: '      ',
            non_field_errors: '      ' //      ,
        }[key] || key; //      ,
        return `${fieldName}: ${Array.isArray(value) ? v
    })
    .join('; '); //
} else {
    errorMessage = JSON.stringify(err.response.data); //
}
}
setError(errorMessage);
setTimeout(() => setError(''), 7000); //      7
};
// ---      ---

```

```

const handleRegister = async (e) => {
    e.preventDefault();
    setError(''); //

    if (password !== confirmPassword) {
        setError('      ');
        setTimeout(() => setError(''), 7000); //      7
        return;
    }
}

```

```

//      email      , ... input type="email"      required

```

```

    try {
      //
      await axios.post('http://localhost:8000/api/auth/register/', {
        username,
        email,
        password,
        password2: confirmPassword // confirmPassword password
      });

      alert('          !          .');
      setIsRegistering(false); //
      //
      setError(''); //
      setEmail(''); // email
      setConfirmPassword(''); //
      // username password
    } catch (err) {
      //
      ,
      handleError(err);
    }
  };

```

```

const handleLogin = async (e) => {
  e.preventDefault();
  setError(''); //

  try {
    //
    // , 'username' ( 'email') 'password'
    const res = await axios.post('http://localhost:8000/api/token/');
    localStorage.setItem('access_token', res.data.access); //
    if (res.data.refresh) {
      localStorage.setItem('refresh_token', res.data.refresh); //
    }
  }
};

```

```

    }
    navigate('/'); //
} catch (err) {
    // , ( , )
    handleError(err);
}
};

return (
    // --- ( ) ---
    <div style={{
        display: 'flex', // flexbox
        flexDirection: 'column', //
        alignItems: 'center', //
        justifyContent: 'center', //
        minHeight: '100vh', //
        backgroundColor: '#e9ecef', //
        padding: '20px', //
        boxSizing: 'border-box' // padding
    }}>
        {/* Header , , , /login
            <Header />

        {/* --- ( ) --- */}
        <div style={{
            backgroundColor: 'ffffff', //
            padding: '40px', //
            borderRadius: '8px', //
            boxShadow: '0 4px 15px rgba(0,0,0,0.1)', //
            maxWidth: '400px', //
            width: '100%', // 100% maxWidth
            textAlign: 'center', // ( , )
            marginTop: '20px', // ( , )
            marginBottom: '20px' //
        }}>

```



```

{ /*          */
<h2 style={{ marginBottom: '30px', color: '#333' }}>
    {isRegistering ? '      ' : '  '}
</h2>

{ /*          */
{error && (
    <div
        className="error-message" //          ,
        style={{ //          inline
            padding: '12px',
            backgroundColor: '#f8d7da', //      -
            color: '#721c24', //      -
            border: '1px solid #f5c6cb', //
            borderRadius: '5px',
            marginBottom: '20px', //
            fontSize: '15px',
            textAlign: 'left', //
            wordBreak: 'break-word' //
        }}
    >
        {error}
    </div>
)}

{ /* ---          --- */
<form
    onSubmit={isRegistering ? handleRegister : handleLogin}
    style={{ //          flex-
        display: 'flex',
        flexDirection: 'column', //
        gap: '15px' //
    }}
>
    { /*          (username) */

```

```

{ /*      username      , ...      */ }
<input
  type="text"
  placeholder="      "
  value={username}
  onChange={(e) => setUsername(e.target.value)}
  required //
  style={{ //
    padding: '12px', //
    borderRadius: '5px', //
    border: '1px solid #ced4da', //
    fontSize: '16px', //
    width: '100%', //
    boxSizing: 'border-box', //      padding  bor
  }}
/>

{ /*      Email (      ) */ }
{isRegistering && (
  <input
    type="email"
    placeholder="Email"
    value={email}
    onChange={(e) => setEmail(e.target.value)}
    required={isRegistering} // required
    style={{
      padding: '12px',
      borderRadius: '5px',
      border: '1px solid #ced4da',
      fontSize: '16px',
      width: '100%',
      boxSizing: 'border-box',
    }}
  />
)}

```

```

{ /*      (      ) */}
<input
  type="password"
  placeholder="      "
  value={password}
  onChange={(e) => setPassword(e.target.value)}
  required //
  style={{
    padding: '12px',
    borderRadius: '5px',
    border: '1px solid #ced4da',
    fontSize: '16px',
    width: '100%',
    boxSizing: 'border-box',
  }}
/>

{ /*      (      ) */}
{isRegistering && (
  <input
    type="password"
    placeholder="      "
    value={confirmPassword}
    onChange={(e) => setConfirmPassword(e.target.
    required={isRegistering} // required
    style={{
      padding: '12px',
      borderRadius: '5px',
      border: '1px solid #ced4da',
      fontSize: '16px',
      width: '100%',
      boxSizing: 'border-box',
    }}
  />

```

```

    })

    {/ *                                */}
    <button
      type="submit"
      style={{
        padding: '12px 20px', //
        backgroundColor: '#007bff', //
        color: 'white', //
        border: 'none', //
        borderRadius: '5px', //
        cursor: 'pointer', //
        fontSize: '18px', //
        fontWeight: 'bold', //
        marginTop: '20px', //
        transition: 'background-color 0.2s ease', //
      }}
      //      hover                                inline
      onMouseOver={(e) => e.target.style.backgroundColor = '#007bff'}
      onMouseOut={(e) => e.target.style.backgroundColor = '#007bff'}
    >
      {isRegistering ? 'Register' : 'Login'}
    </button>
  </form>

  {/ *                                */}
  <button
    className="toggle-button" //
    onClick={() => {
      setIsRegistering(!isRegistering);
      //
      setError('');
      setEmail('');
      setPassword('');
      setUsername('');
    }}
  >
    Toggle
  </button>

```

```

        setConfirmPassword('');
    }}
    style={{ //
        marginTop: '25px', //
        backgroundColor: 'transparent', //
        color: '#007bff', //
        border: 'none', //
        cursor: 'pointer', //
        fontSize: '15px',
        textDecoration: 'underline', //
    }}
    >
        {isRegistering ? ' ' : ' '}
    </button>
</div>
</div>
);
}

```

```
export default Login;
```

## 2.7 pages/Upload.js

Upload.js это страница загрузки видео

```

import React, { useState, useEffect } from 'react';
import axios from 'axios';
import { useNavigate } from 'react-router-dom';
import Header from '../components/Header'; // Header

const Upload = () => {
    // : 'file'
    const [file, setFile] = useState(null);
    const [title, setTitle] = useState('');
    const [description, setDescription] = useState('');

```

```

const [loading, setLoading] = useState(false);
const [error, setError] = useState('');
const navigate = useNavigate();

//
useEffect(() => {
  //
  if (!localStorage.getItem('access_token')) {
    navigate('/login');
  }
}, [navigate]); //

//
const handleFileChange = (e) => {
  const selectedFile = e.target.files[0];

  console.log('      : ', selectedFile); //

  //
  const allowedTypes = ['video/mp4', 'video/webm', 'video/ogg'];
  if (selectedFile && !allowedTypes.includes(selectedFile.type)) {
    setError('      MP4, WebM  OGG');
    setFile(null); //
    e.target.value = null; //      input[type="file"]
    console.log('      .'); //
    return;
  }

  //      ( . 500MB)
  if (selectedFile && selectedFile.size > 500 * 1024 * 1024) {
    setError('      : 500MB');
    setFile(null); //
    e.target.value = null; //      input[type="file"]
    console.log('      .'); //
    return;
  }

```

```

    }

    //
    setFile(selectedFile);
    setError(''); //
    console.log('                .'); //
};

//
const handleUpload = async (e) => {
    e.preventDefault();
    setLoading(true);
    setError(''); //

    if (!file) {
        setError('                ');
        setLoading(false);
        console.log('                .'); //
        return;
    }

    const formData = new FormData();
    //                ,                'video'                /
    formData.append('video', file);
    formData.append('title', title);
    formData.append('description', description);

    console.log('                FormData...'); //
    //                FormData (                ,                )
    for (let [key, value] of formData.entries()) {
        console.log(`${key}: ${value}`);
    }
    console.log('                FormData:', formData); //                :                Form

    try {

```

```

const accessToken = localStorage.getItem('access_token');
if (!accessToken) {
  setError('                .                .');
  setLoading(false);
  navigate('/login'); //
  return;
}

console.log('                ...'); //
const response = await axios.post('http://localhost:8000/api/vi
  headers: {
    'Authorization': `Bearer ${accessToken}`, //
    // 'Content-Type': 'multipart/form-data', // Axios
  },
  //
  onUploadProgress: (progressEvent) => {
    const percentCompleted = Math.round((progressEvent.loaded
    console.log(`                : ${percentCompleted}%`); //
    //
  }
});

console.log('                :', response.data); //
//
navigate(`/video/${response.data.id}`);

} catch (err) {
  console.error('                :', err.response || err); //
  let errorMessage = '                ';

  if (err.response) {
    if (err.response.status === 401) {
      errorMessage = '                .                .';
      //
      401
      localStorage.removeItem('access_token');

```



```

        localStorage.removeItem('refresh_token');
        navigate('/login');
    } else if (err.response.status === 415) {
        errorMessage = '                        .                MP4, WebM      OGG';
    } else if (err.response.data && typeof err.response.data ===
        //                                DRF
        errorMessage = Object.entries(err.response.data)
            .map(([key, value]) => {
                //                                DRF,
                const fieldName = {
                    video: '      ',
                    title: '      ',
                    description: '      ',
                    non_field_errors: '      ' //            ,
                }[key] || key; //            ,
                return `${fieldName}: ${Array.isArray(value) ? va
            })
            .join('; '); //
    } else {
        //
        errorMessage = err.response.data?.detail || JSON.stringify(
    }
} else if (err.request) {
    //            (      ,                )
    errorMessage = '                        .                .';
} else {
    //
    errorMessage = err.message;
}

setError(errorMessage); //
} finally {
    setLoading(false); //
}
};

```

```

// --- JSX          ---
return (
  //
  <div style={{
    display: 'flex', //          flexbox
    flexDirection: 'column', //
    alignItems: 'center', //
    justifyContent: 'center', //
    minHeight: 'calc(100vh - 60px)', //          (
    padding: '20px', //
    boxSizing: 'border-box' //          padding
  }}>
    {/* Header      ,      ,      ,      /upload
    {/* <Header /> */}

    {/* ---          (          ) --- */}
    <div style={{
      backgroundColor: '#ffffff', //
      padding: '40px', //
      borderRadius: '8px', //
      boxShadow: '0 4px 15px rgba(0,0,0,0.1)', //
      maxWidth: '600px', //          (          )
      width: '100%', //          100%          maxWidth
      textAlign: 'center', //          (      ,      )
      marginTop: '20px', //          (          ,          )
      marginBottom: '20px' //
    }}>
      {/*          */}
      <h2 style={{ marginBottom: '30px', color: '#333' }}>

      </h2>

      {/*          */}
      {error && (

```

```

<div
  className="error-message" //
  style={{ //      inline
    padding: '12px',
    backgroundColor: '#f8d7da', //      -
    color: '#721c24', //      -
    border: '1px solid #f5c6cb', //
    borderRadius: '5px',
    marginBottom: '20px', //
    fontSize: '15px',
    textAlign: 'left', //
    wordBreak: 'break-word' //
  }}
>
  {error}
</div>
)}}

{/* ---      --- */}
<form
  onSubmit={handleUpload}
  style={{ //      flex-
    display: 'flex',
    flexDirection: 'column', //
    gap: '15px' //
  }}
>
  {/*      */}
  <input
    type="text"
    placeholder="      "
    value={title}
    onChange={(e) => setTitle(e.target.value)}
    required //
    style={{ //

```

```

        padding: '12px', //
        borderRadius: '5px', //
        border: '1px solid #ced4da', //
        fontSize: '16px', //
        width: '100%', //
        boxSizing: 'border-box', //           padding   bor
    }}
/>

{ /*           */ }
<textarea
    placeholder="           "
    value={description}
    onChange={(e) => setDescription(e.target.value)}
    style={{ //
        padding: '12px',
        borderRadius: '5px',
        border: '1px solid #ced4da',
        fontSize: '16px',
        width: '100%',
        boxSizing: 'border-box',
        minHeight: '100px', //
        resize: 'vertical' //
    }}
/>

{ /*           */ }
<input
    type="file"
    accept="video/mp4,video/webm,video/ogg" //
    onChange={handleFileChange} //
    required //
    style={{ //
        padding: '12px',
        borderRadius: '5px',

```

```

        border: '1px solid #ced4da',
        fontSize: '16px',
        width: '100%',
        boxSizing: 'border-box',
        //          input[type="file"]          ,
    }}
/>
{ /*          */}
{file && (
    <div style={{ fontSize: '13px', color: '#6c757d',
        : {file.name} ({(file.size / (1024 * 1024)
    </div>
)}}

{ /*          */}
<button
    type="submit"
    //          ,          ,
    disabled={loading || !!error || !file}
    style={{ //
        padding: '12px 20px',
        //          disabled
        backgroundColor: loading || !!error || !file ?
        color: loading || !!error || !file ? '#adb5bd'
        border: 'none',
        borderRadius: '5px',
        //          disabled
        cursor: loading || !!error || !file ? 'not-allow
        fontSize: '18px',
        fontWeight: 'bold',
        marginTop: '20px',
        transition: 'background-color 0.2s ease', //
    }}
    //          hover          inline

```

```

        onMouseOver={e} => { if (!(loading || !!error ||
        onMouseOut={e} => { if (!(loading || !!error || !
    >
        {loading ? '        ...' : '        '}
    </button>
  </form>
</div>
</div>
);
};

export default Upload;

```

## 2.8 pages/VideoPage.js

VideoPage.js это страница с видео проигрывателем

```

import React, { useEffect, useState } from 'react';
import { useParams } from 'react-router-dom';
import ReactPlayer from 'react-player';
import axios from 'axios';
import Header from '../components/Header';

const VideoPage = () => {
  const { id } = useParams();
  const [video, setVideo] = useState(null);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState('');

  useEffect(() => {
    const fetchVideo = async () => {
      try {
        const response = await axios.get(`http://localhost:8000/api/v

```

```

        headers: {
            'Authorization': `Bearer ${localStorage.getItem('access_token')}`
        }
    });
    setVideo(response.data);
} catch (err) {
    setError('');
} finally {
    setLoading(false);
}
};

fetchVideo();
}, [id]);

if (loading) return <div> ...</div>;
if (error) return <div>{error}</div>;
if (!video) return <div> </div>;

return (
    <div>
        <Header />
        <div className="video-container" style={{ maxWidth: '1200px', margin: '0 auto' }}>
            <div className="player-wrapper" style={{ position: 'relative' }}>
                <ReactPlayer
                    className="react-player"
                    url={video.file_url}
                    controls
                    width="100%"
                    height="100%"
                    style={{ position: 'absolute', top: 0, left: 0 }}
                    config={{
                        file: {
                            attributes: {
                                controlsList: 'nodownload' //

```

```

        }
      }
    }}
  />
</div>

<div className="video-info" style={{ marginTop: '20px' }}>
  <h1>{video.title}</h1>
  <p style={{ color: '#666', margin: '10px 0' }}>
    : {video.author_username} • {new Date(video.uploaded_at)}
  </p>
  <p style={{ lineHeight: '1.6' }}>{video.description}</p>
</div>
</div>
</div>
);
};

export default VideoPage;

```

## 2.9 App.js

обязательно прописываем App.js в корневом каталоге проекта чтоб проект понимал как интерпритировать наши страницы

```

// src/App.js
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import VideoPage from './pages/VideoPage';
import Home from './pages/Home';
import Login from './pages/Login';
import Upload from './pages/Upload';
import VideoList from './pages/VideoList'; //
import Account from './pages/Account';
import Header from './components/Header'; // <--

```



```

function App() {
  return (
    <Router>
      {/* Header , Routes, */}
      <Header />

      {/* Routes */}
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/login" element={<Login />} />
        <Route path="/upload" element={<Upload />} />
        {/* , /videolist , */}
        <Route path="/videolist" element={<VideoList />} />
        <Route path="/video/:id" element={<VideoPage />} />
        <Route path="/account" element={<Account />} />
      </Routes>
    </Router>
  );
}

export default App;

```