

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)

ОТЧЁТ

по дисциплине

«Web-Технологии»

по теме:

Создание тестовой видео платформы с использованием технологии python
для бэкэнда, React JS для фронтенда и Figma для дизайна.

Студент:

ИКС-431 А.В.Столба

Преподаватель:

А.В.Андреев

Новосибирск 2025

СОДЕРЖАНИЕ

1. Основная работа

1.1. Снимки выполненной работы

1. Основная работа

1.1. Снимки выполненной работы

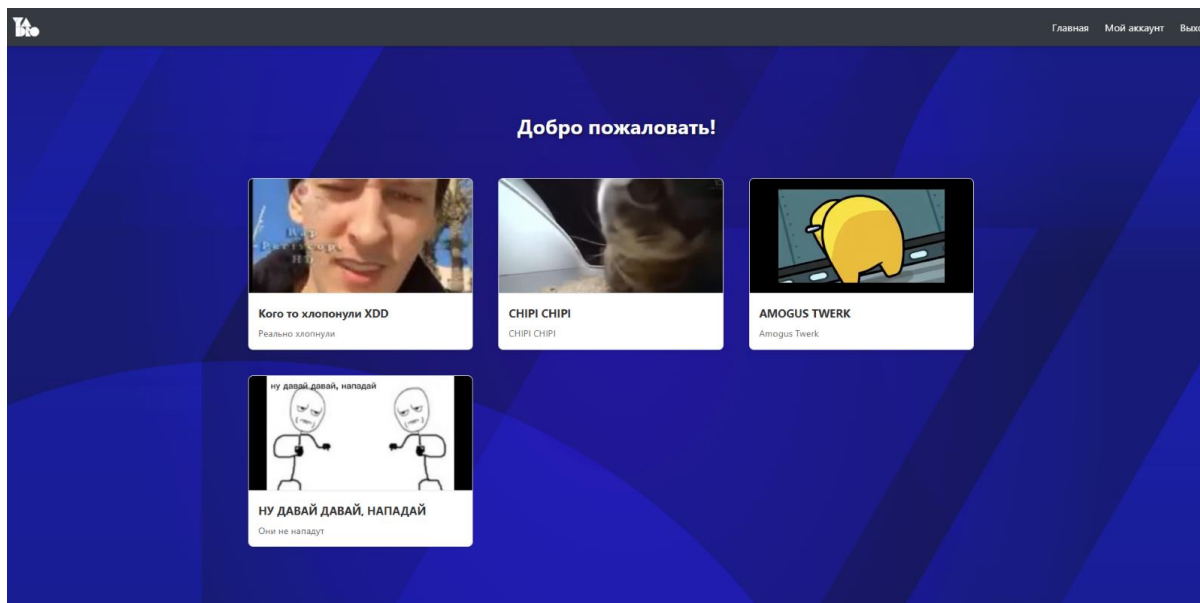


Рисунок 1 — Главная страница

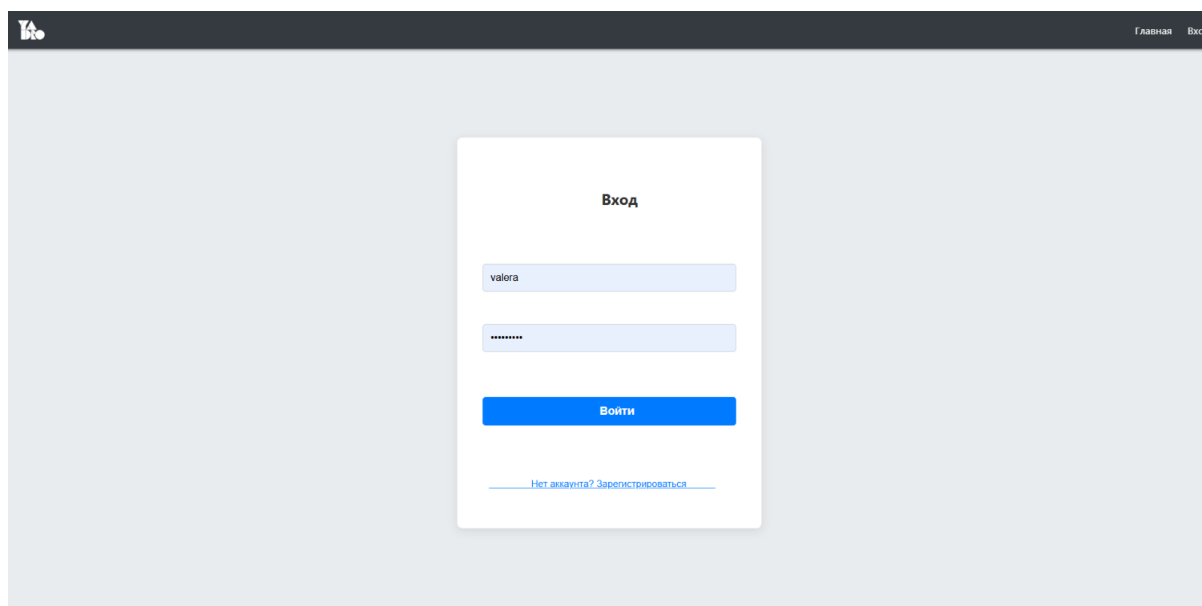


Рисунок 2 — Страница авторизации

Убт

Главная Вход

Регистрация

Имя пользователя

Email

Пароль

Повторите пароль

Код доступа

Зарегистрироваться

[Уже есть аккаунт? Войти](#)

Рисунок 3 — Страница регистрации

Профиль пользователя

Имя: valera

Email: valera@yandex.ru

Дата регистрации: 5/23/2025

Загрузить новое видео

Мои видео (4)



ЧИПИ ЧИПИ

ЧИПИ ЧИПИ

Рисунок 4 — Страница аккаунта

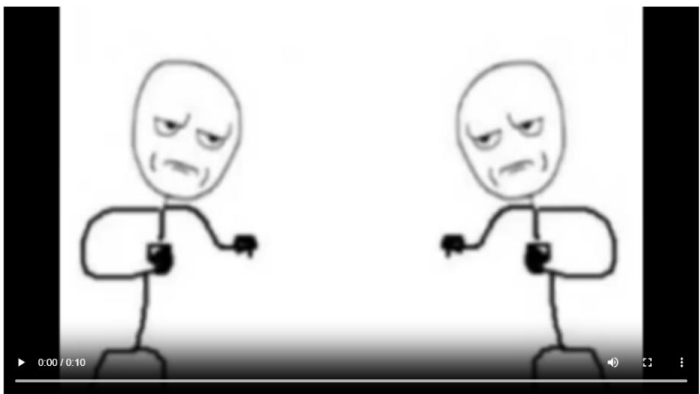
Загрузить видео

Choose File

No file chosen

Загрузить видео

Рисунок 5 — Добавление видео



НУ ДАВАЙ ДАВАЙ, НАПАДАЙ

valera • 5/23/2025

Они не нападут

Комментарии (1)

valera: А че они стоят?
5/23/2025, 3:58:12 AM

Отправить

Рисунок 6 — Страница просмотра

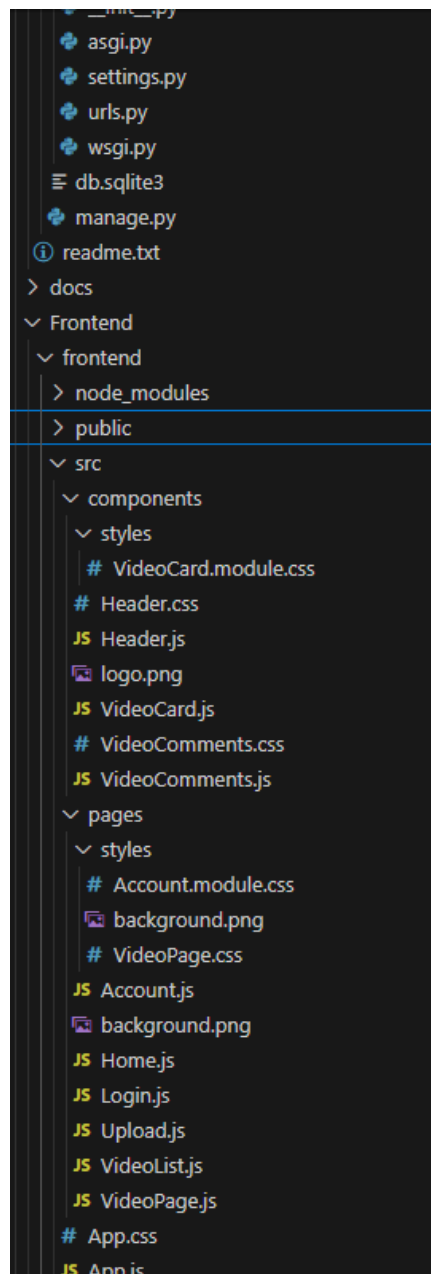


Рисунок 7 — Структура проекта на сервере

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>

```

Рисунок 8 — HTML код

```

from django.urls import path, include
from rest_framework.routers import DefaultRouter
from . import views
from .views import RegistrationAPIView
from django.conf import settings
from django.conf.urls.static import static
from django.urls import re_path
from django.views.static import serve
from .views import (
    CurrentUserView,
    UserVideoListView,
    VideoCommentsListCreateView
)

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

urlpatterns = [
    path('register/', RegistrationAPIView.as_view(), name='register'),
    path('auth/me/', CurrentUserView.as_view(), name='current-user'),
    path('videos/user/', UserVideoListView.as_view(), name='user-videos'),
    path('videos/<int:video_id>/comments/', VideoCommentsListCreateView.as_view(), name='video-comments-list-create'),

    path('', include(router.urls)),
]

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

Рисунок 9 — Backend на Python с использованием Django


```

3   const handleRegister = async (e) => {
4       setConfirmPassword(''); // Очищаем подтверждение пароля
5       setAccessCode(''); // Сбрасываем поле кода доступа
6       // username и password можно оставить заполненными для удобства автоматического входа
7   } catch (err) {
8       // Обрабатываем ошибки, полученные от бэкенда
9       handleError(err);
10  };
11
12
13  const handleLogin = async (e) => {
14      e.preventDefault();
15      setError(''); // Сбросить предыдущие ошибки перед новым действием
16
17      try {
18          // Отправляем запрос на получение токена
19          // Убедитесь, что ваш бэкенд принимает 'username' (или 'email') и 'password'
20          const res = await axios.post('http://localhost:7999/api/token/', { username, password }); // Обычно для токена используется username
21          localStorage.setItem('access_token', res.data.access); // Сохраняем access token
22          if (res.data.refresh) {
23              localStorage.setItem('refresh_token', res.data.refresh); // Опционально: сохраняем refresh token, если бэкенд его выдает
24          }
25          navigate('/'); // Перенаправляем на главную после успешного входа
26      } catch (err) {
27          // Обрабатываем ошибки, полученные от бэкенда (например, неверные учетные данные)
28          handleError(err);
29      }
30  };

```

Рисунок 10 — Frontend на JS

Проект