МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И

МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СВЯЗИ И ИНФОРМАТИКИ»

Кафедра «Сети связи и системы коммутации»

Лабораторная работа №4

«Создание страницы определенной записи»

по дисциплине «Web-программирование»

Выполнил студент

группы БФИ1901

Шацкий Е.И.

Проверила:

Полянцева К.А.

Москва, 2021

**Содержание**

[1 Задание на лабораторную работу 2](#_Toc88547916)

[2 Ход лабораторной работы. 3](#_Toc88547917)

[2.1 Создание страницы определенной записи 3](#_Toc88547918)

[2.2 Верстка обеих страниц в соответствии с макетом 4](#_Toc88547919)

[Список используемых источников 9](#_Toc88547920)

# **1 Задание на лабораторную работу**

1. Сделать так, чтобы при клике по названию статьи происходил переход на страницу указанной записи.

2. Создайте стили, соответствующие макету для страницы определенной записи. Не забудьте добавить ссылку «Все записи»;

3. Загрузите ваш проект на любой гит-репозиторий (GitHub, GitLab, Google Code, Bitbucket и т.п.

# **2 Ход лабораторной работы.**

## **2.1 Создание страницы определенной записи**

Для начала выполнения лабораторной работы, в директории lab4 создам новый проект blog, проведу первоначальную настройку как в прошлых лабораторных работах и перенесу приложение articles из предыдущей работы (скопируйте папку в этот проект, добавьте нужные данные в INSTALLED\_APP и настройте файл urls.py).

Для того, чтобы можно было бы прочитать значение идентификатора (id) отдельного поста, укажу в качестве адреса следующее регулярное выражение (в файле urls.py):

urlpatterns = [

path(r'^article/(?P<article\_id>\d+)$', views.get\_article, name='get\_article')

]

После того, как в регулярном выражении мы объявили именованную группу, в представление можно добавлять именованный аргумент, куда и будет передано значение из адреса, по которому перешел пользователь. Создайте в файле views.py из директории articles помимо уже написанной в предыдущей работе функции archieve ещё и функцию get\_article:

from django.http import Http404

def get\_article(request, article\_id):

try:

post = Article.objects.get(id=article\_id)

return render(request, 'article.html', {"post": post})

except Article.DoesNotExist:

raise Http404

Следующим шагом, по плану лабораторной работы, необходимо сделать каждый заголовок статьи ссылкой на саму страницу статьи. Для этого в тэг заголовка поместим тэг «a» c атрибутом «href» в котором будет указана ссылка на статью. Это показано ниже.

<h2 class="post-title">

<a href="http://127.0.0.1:8000/article/{{post.id}}">{{post.title}}</a>

</h2>

**2.2 Верстка обеих страниц в соответствии с макетом**

Далее, необходимо создать сделать верстку двух страниц сайта в соответствии с макетами, представленными в методических указаниях к лабораторной работе, то есть главной страницы, за которую отвечает файл «archive.html» и страницы с каждый отдельной статьей, за которую отвечает файл «articles.html». За стили отвечает файл «articles.css». В листингах 1-3 представлен программный код вышеуказанных файлов. На рисунках 1-2 представлены страницы сайта.

Листинг 1 – Файл «archive.html»

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Архив статей</title>

<link rel="stylesheet" href="{{ STATIC\_URL }}/static/article.css">

</head>

<body>

<div class="header">

<img src="{{ STATIC\_URL }}/static/img/cat.jpg"/>

</div>

<div class="archive">

{% for post in posts %}

<div class="one-post">

<h2 class="post-title"><a href="http://127.0.0.1:8000/article/{{post.id}}">{{post.title}}</a></h2>

<div class="article-info">

<div class="article-author">{{ post.author.username }}</div>

<div class="article-created-date">{{ post.created\_date }}</div>

<p class="article-text">{{ post.get\_excerpt }}</p>

</div>

</div>

{% endfor %}

</div>

</body>

</html>

Листинг 2 – Файл «article.html»

<!DOCTYPE html>

<html lang = "en">

<head>

<title>{{post.title}}</title>

<link rel="stylesheet" href="{{ STATIC\_URL }}/static/article.css">

</head>

<body>

<div class="archive">

<div class="article-border">

<p>

<div class="archive-header">

<img class="article-image" src="{{ STATIC\_URL }}/static/img/cat.jpg"/>

</div>

<a class="link" href="http://127.0.0.1:8000">Все статьи</a>

</p>

</div>

<div class="post">

<h1 class="post-title">{{post.title}}</h1>

<div class="article-info">

<h2>

<div class="article-author">{{post.author.username}}</div>

</h2>

<div class="article-created-data">{{post.created\_date}}</div>

<p class="article-text">{{ post.text }}</p>

</div>

</div>

</div>

</body>

</html>

Листинг 3 – Файл «article.css»

body{

background: #1abc9c;

font-family: Tahoma, Arial, sans-serif;

color: #ffffff

}

img {

display: block;

width: 318px;

margin-left: auto;

margin-right: auto;

}

.archive {

width: 960px;

margin-left: auto;

margin-right: auto;

}

.post-title a {

color: #ffffff;

}

.article-author {

width: 50%;

float: left;

}

.article-created-date {

text-align: right;

}

.article-image{

display: block;

width: 318px;

margin-left: 0;

}

.link {

color: white;

font-weight: bold;

position: absolute;

right: 470px;

top: 180px;

}

.article-border p{

text-align: right;

}

.article-text{

width: 960px;

text-align: justify;

}

.article-created-data{

text-align: right;

}

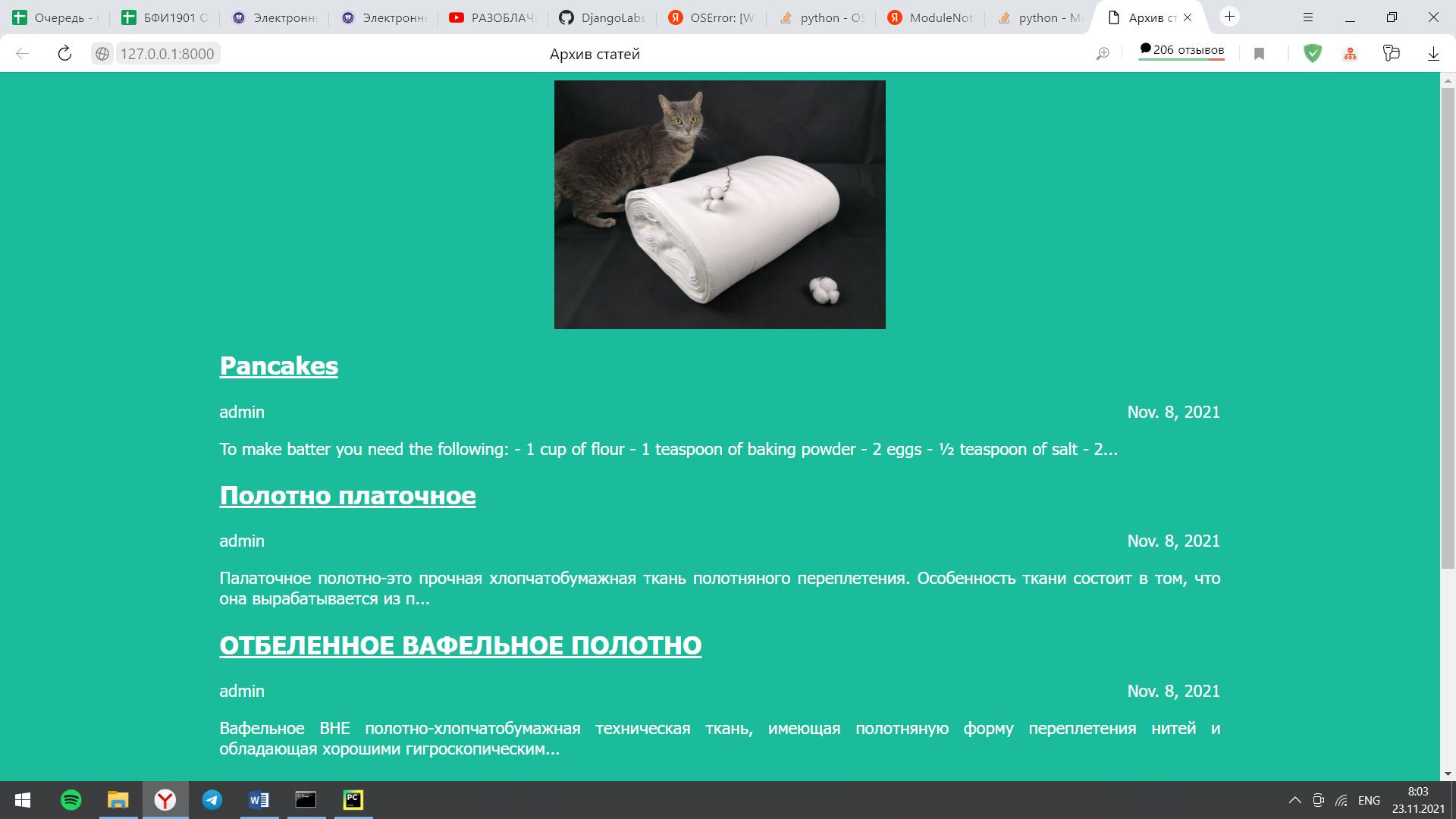


Рисунок 1 – Главная страница

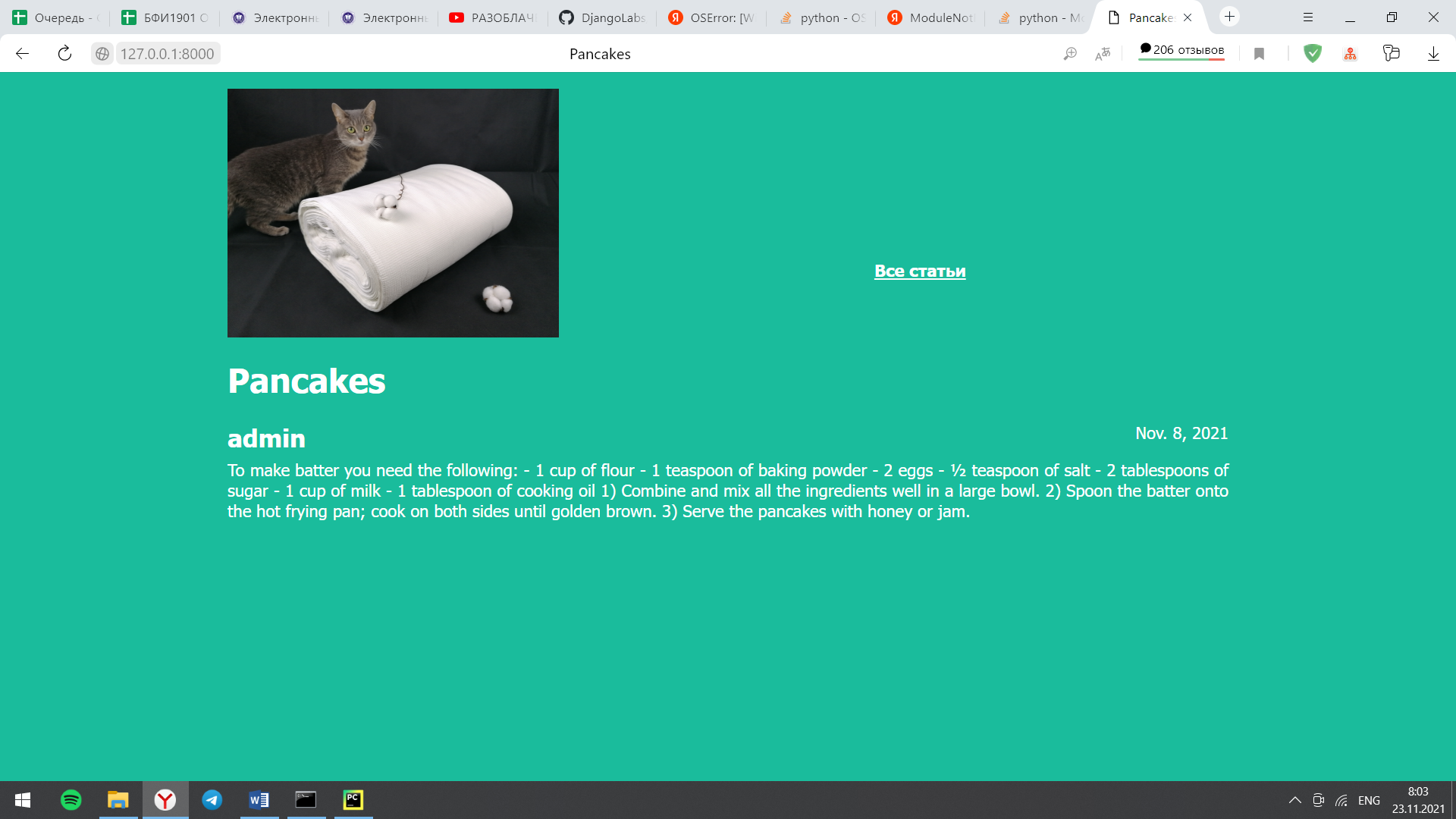


Рисунок 2 – Страница каждой отдельной статьи

Загрузка на github представлена на рисунке 3.

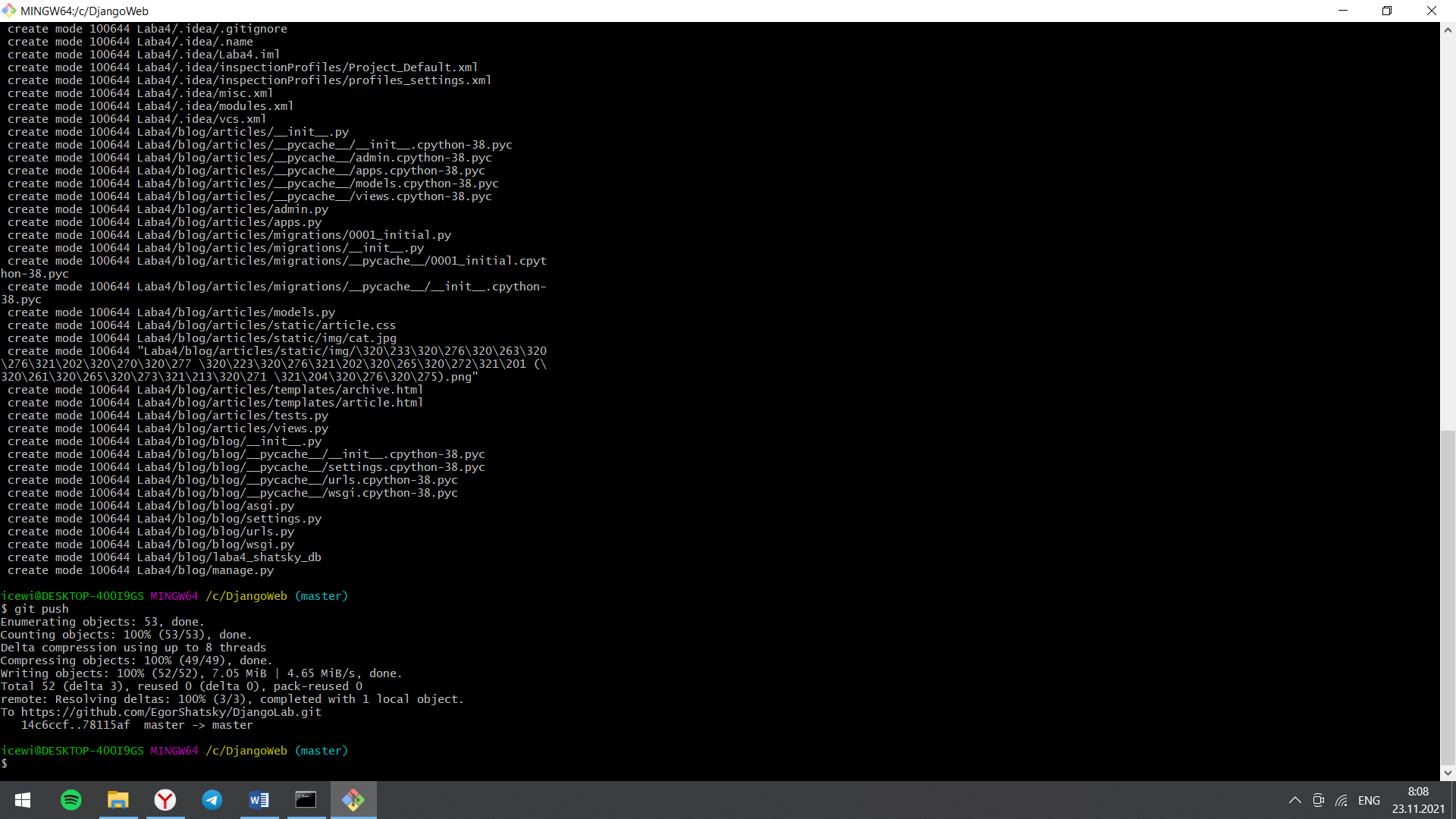


Рисунок 3 – Загрузка проекта

**Список используемых источников**

1 ГОСТ 7.32-2017 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления.

2 ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание.