



PYTHON CODING ASSIGNMENT REPORT



Source code:

The website's main function is to allow the user to submit a (.txt) file, after which it will extract the words and numbers and display them as individual lists. We used the Visual Studio IDE and Django framework to write Python code to implement the functionality, as well as HTML and CSS to display it and create a visually appealing interface.

1) Python Source Code:

```
#to deal with HTTP requests and so on
from django.shortcuts import render

#to interact with TextFile model which allows perform database operations on TextFile object
from .models import TextFile

#For using split() and findall() and sub()
import re

'''This method opens a file using its path,a for loop applied on each line in the
file to split the line into individual words and assign them to array words,
then a for loop applied on array words then define a regular expressions using the
findall function which will search for numeric parts within word and
assign it to numericParts array if numericParts is not empty then add all its items
to onlyNumbers array using extend(),
then use sub() function to search and remove the numeric part of the word then store
it in word var, if the word var not empty then add it to onlyWords array
then just return two arrays.
'''

def seperateData(filePath):
    onlyWords = []
    onlyNumbers = []
    with open(filePath) as file:
        for eachLine in file:
            words = eachLine.split()
            for word in words:
                numericParts = re.findall(r'[+-]?\d+(?:\.\d+)?', word)
                if numericParts:
                    onlyNumbers.extend(numericParts)
                    word = re.sub(r'[+-]?\d+(?:\.\d+)?', '', word)
                if word:
                    onlyWords.append(word)
    return onlyWords, onlyNumbers
```

```

'''This method receives the value of the HTTP request and then acts based on it. It
also checks the file type and creates the object file using
TextFile model and specify its path and directory after that return the responding
HTML page based on the condition.
'''

def upload_file(request):
    if request.method == 'POST' and request.FILES.get('file'):
        file = request.FILES['file']
        if file.content_type == 'text/plain':
            text_file = TextFile.objects.create(file=file)
            filePath = text_file.file.path
            onlyWords, onlyNumbers = seperateData(filePath)
            return render(request, 'upload_File/result.html', {'wordsList':
onlyWords, 'numbersList': onlyNumbers})
        else:
            return render(request, 'upload_File/error.html', {'errorMessage': 'Only
text files (.txt) are allowed.'})
    return render(request, 'upload_File/upload.html')

```

2) HTML:

1. Upload HTML

```

2. <!DOCTYPE html>
3.
4. <html>
5. <head>
6.     <title>File Upload</title>
7.     <!--To link CSS with HTML-->
8.     {% load static %}
9.     <link rel="stylesheet" href="{% static 'css/styleSheetFile.css' %}">
10.
11. </head>
12. <body>
13.     <!--div to control these elements using CSS-->
14.     <div class="container">
15.         <div class="text-box">
16.             <!--To link Image with HTML-->
17.             {% load static %}
18.             
19.
20.         <h1>Upload a Text File </h1>
21.         <br></br>
22.     <!--To do operations like choosing the file and uploading it-->
23.     <form method="post" enctype="multipart/form-data">
24.         {% csrf_token %}
25.         <input type="file" name="file" accept=".txt" class="choose-file">
26.         <br><br>

```

```

27.         <input type="submit" value="Upload" class="upload-button ">
28.     </form>
29. </div>
30.</div>
31.</body>
32.
33.</html>

```

2.Result HTML:

```

<!DOCTYPE html>

<html>
<head>
    <title>File Upload Result</title>
    <!--To link CSS with HTML-->
    {% load static %}
    <link rel="stylesheet" href="{% static 'css/styleSheetFile.css' %}">
</head>
<body>
    <!--div to control these elements using CSS-->
    <div claa="container">
        <div class="text-box1 ">
            <h1>File uploaded successfully!</h1>
            <!--To show the list on HTML page-->
            <h2>Words are :</h2>
            <ul>
                <!--the context provided in python-->
                {% for word in wordsList %}
                    <li>{{ word }}</li>
                {% empty %}
                    <li>No words found.</li>
                {% endfor %}
            </ul>
            <h2>Numbers are :</h2>
            <ul>
                {% for number in numbersList %}
                    <li>{{ number }}</li>
                {% empty %}
                    <li>No numbers found.</li>
                {% endfor %}
            </ul>
        </div>
    </div>
</body>
</html>

```

3.Error HTML:

```
<!DOCTYPE html>

<html>
<head>
  <title>File Upload Error</title>
  <!--To link CSS with HTML-->
  {% load static %}
  <link rel="stylesheet" href="{% static 'css/styleSheetFile.css' %}">
</head>
<body>
  <!--div to control these elements using CSS-->
  <div class="container">
    <div class="text-box">
      <div class="errorUpload">
        <!--the context provided in python-->
        <h1>Error: {{ errorMessage }}</h1>
      </div>
      <br></br>
      <!--To upload another File-->
      <div class="rechoose">
        <h2>Could you rechoose a Text File</h2>
      </div>
      <br></br>
      <form method="post" enctype="multipart/form-data">
        {% csrf_token %}
        <input type="file" name="file" accept=".txt" class="rechoose-file">

        <input type="submit" value="Upload" class="reupload ">
      </form>
    </div>
  </div>
</body>
</html>
```

3) CSS Source Code:

```
/* The main setting for all HTML pages*/

*{
  margin: 0;
  padding: 0;
  font-family: 'Poppins', sans-serif;
  box-sizing: border-box;
  list-style: none;
}

/* styling rules for various elements on an HTML page. It sets styles for
the body, container, text boxes, upload button, and error messages.*/
body {
  position: relative;
  background: rgb(238,174,202);
  background: radial-gradient(circle, rgba(238,174,202,1) 0%, rgba(148,187,233,1)
100%);
  color: rgb(224, 120, 161);

  .container {
    text-align: center;
    -webkit-text-stroke-width: 1.5px;
    -webkit-text-stroke-color: rgb(159, 73, 108);
    font-size: 32px;
    font-style: normal;
    font-weight: 800;
    line-height: normal;
  }

  .text-box {
    display: inline-block;
    margin: 10px auto;
    padding: 20px;
    border-radius: 30px;
    background: #FEFBFB;
    width: 1050px;
    height: 610px;
  }

  .text-box1{
    display: inline-block;
    margin: 10px auto;
    padding: 20px;
    border-radius: 30px;
    background: #FEFBFB;
    width: 960px;
    height:fit-content;
  }
}
```

```

.upload-button {
    display: block;
    margin: 10px auto;
    border-radius: 30px;
    border: 5px solid #c07eaf;
    background: #fccff1;
    width: 260px;
    height: 77px;
    color: rgb(224, 120, 161);
-webkit-text-stroke-width: 2px;
-webkit-text-stroke-color:rgb(190, 94, 133);
font-size: 32px;
font-style: normal;
font-weight: 800;
line-height: normal;
}
.choose-file{
display: block;
-webkit-text-stroke-width: 1.24px;
-webkit-text-stroke-color:  rgb(190, 94, 133);
font-size: 32px;
font-style: normal;
font-weight: 800;
line-height: normal;
position: absolute;
top: 50%;
left: 60%;
transform: translate(-50%, -50%);
}
.rechoose-file{
    display: block;
    -webkit-text-stroke-width: 1.24px;
    -webkit-text-stroke-color:  rgb(190, 94, 133);
    font-size: 28px;
    font-style: normal;
    font-weight: 800;
    line-height: normal;
    position: absolute;
    top: 55%;
    left: 60%;
    transform: translate(-50%, -50%);
}
.rechoose{
    font-size: 28px;
    top: 50%;
}
.errorUpload{
    color: rgb(228, 92, 92);
    -webkit-text-stroke-color:rgb(139, 60, 60);

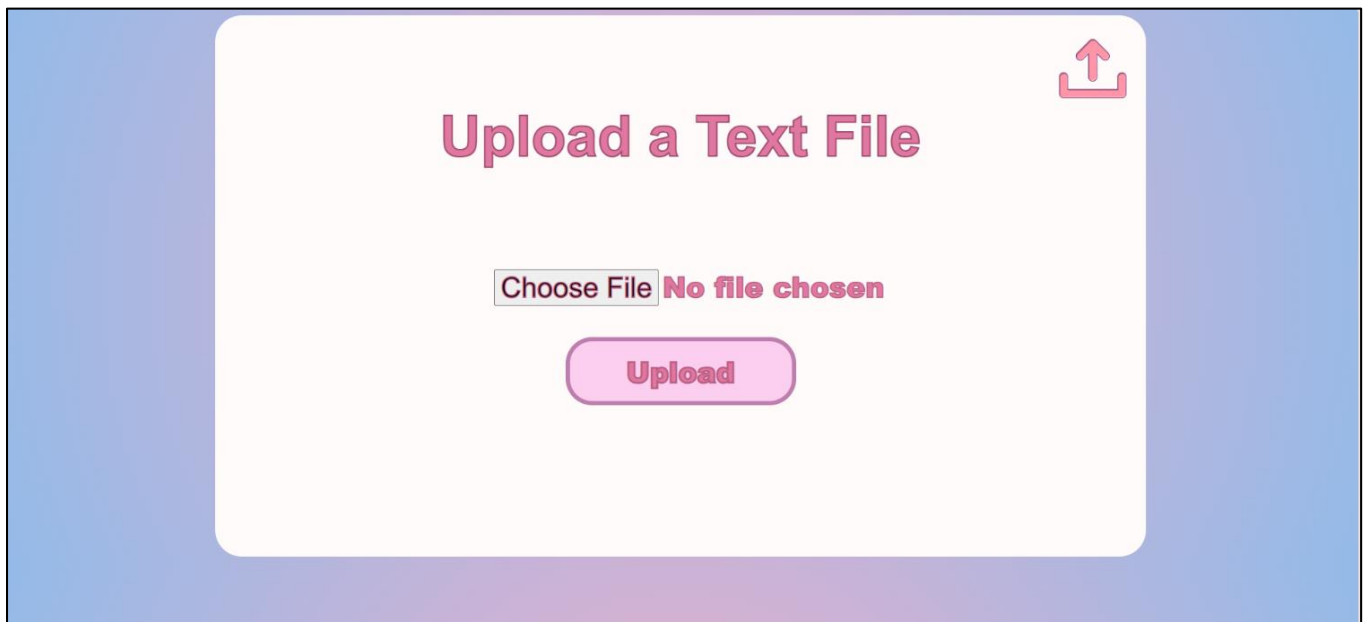
```

```
font-size: 25px;

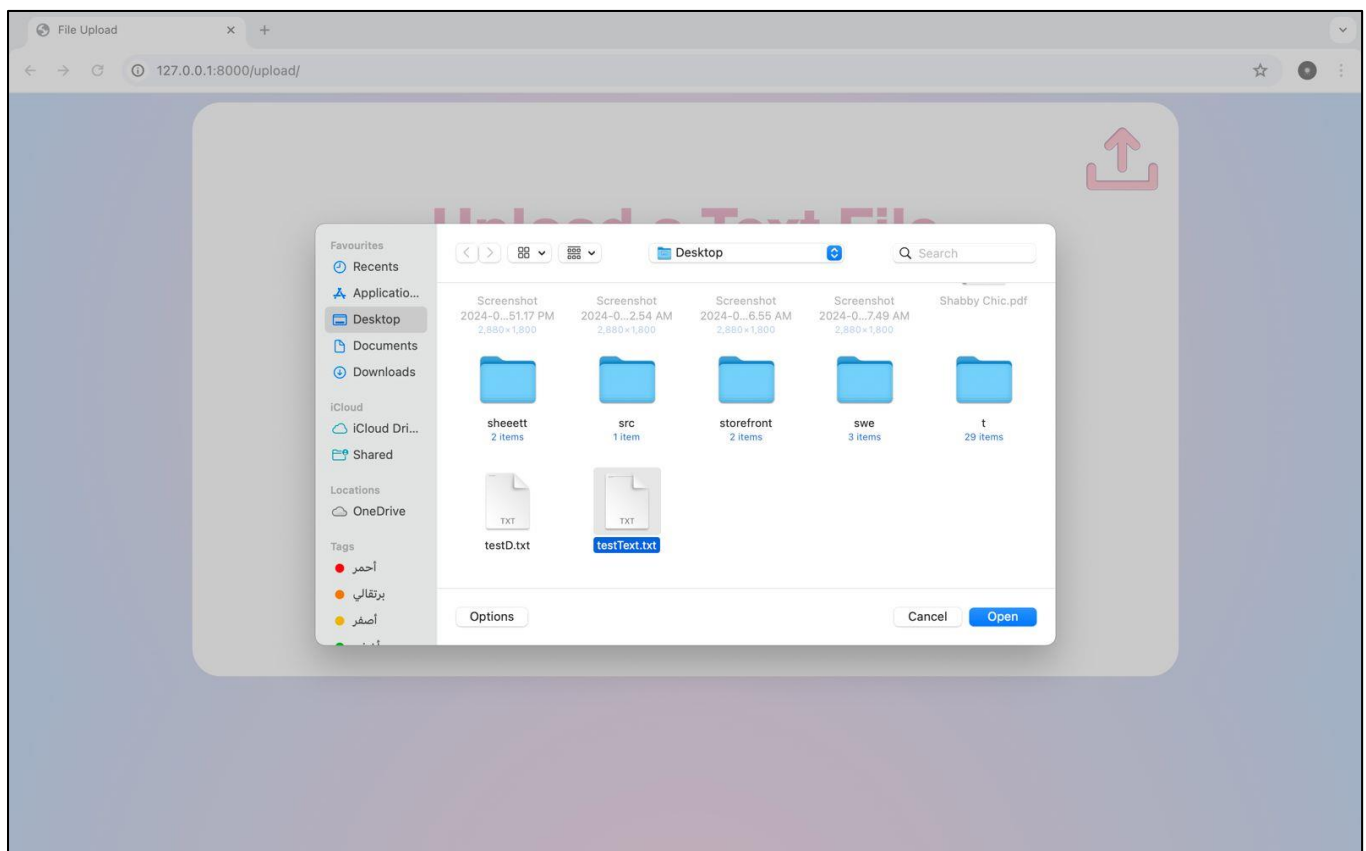
}
.reupload{
    display: block;
    margin: 10px auto;
    border-radius: 30px;
    border: 5px solid #c07eaf;
    background: #fccff1;
    width: 260px;
    height: 77px;
    color: rgb(224, 120, 161);
-webkit-text-stroke-width: 2px;
-webkit-text-stroke-color:rgb(190, 94, 133);
font-size:28px;
font-style: normal;
font-weight: 800;
line-height: normal;
position: absolute;
top: 70%;
left:43%;
}

}
```

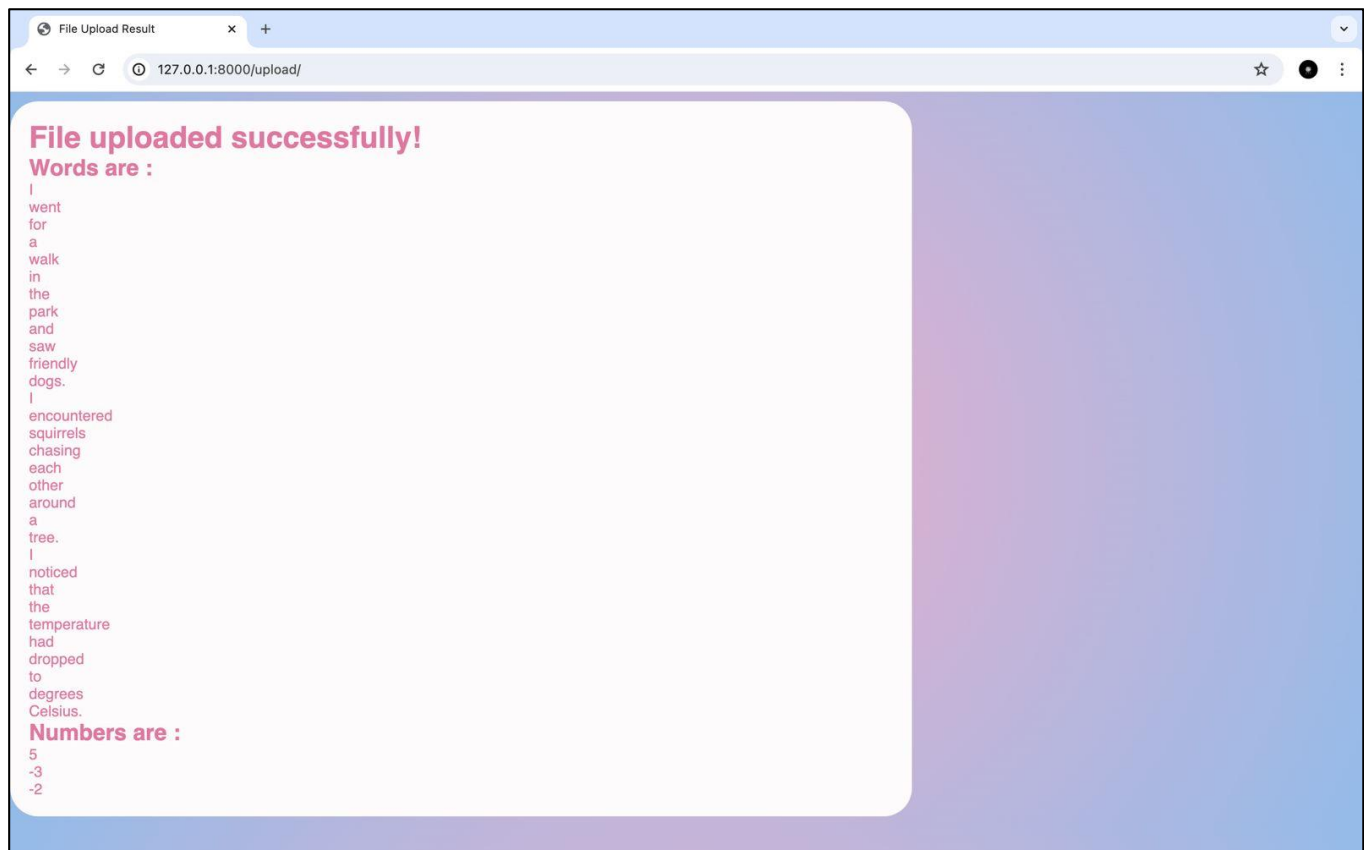

Screenshots of the inputs and outputs:



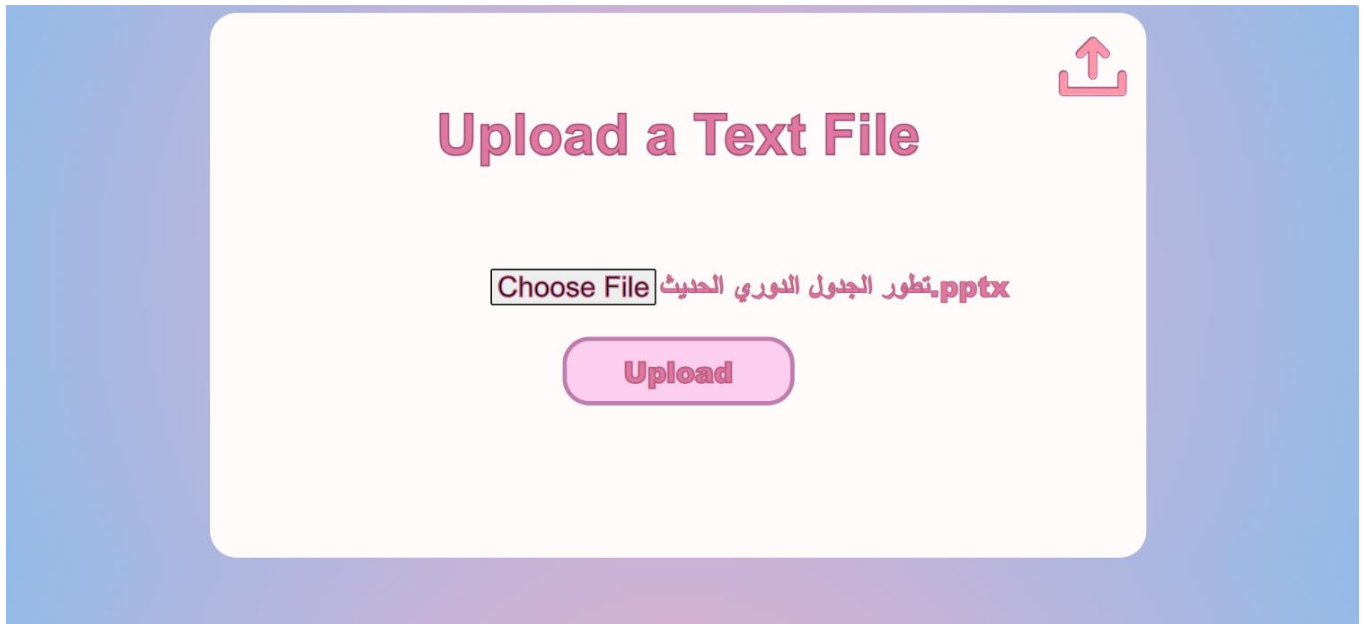
Figure(1):The main page



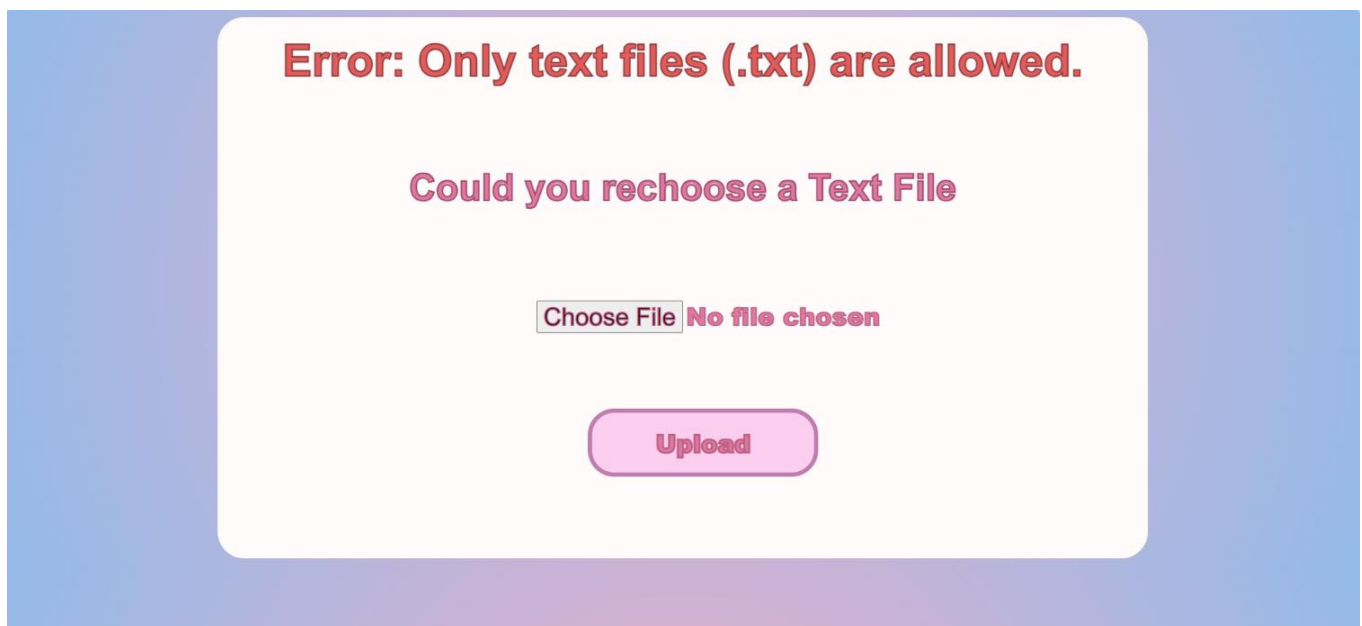
Figure(2):Uploading a .txt file



Figure(2):The result of uploading a txt file



Figure(3):Uploading a .pptx file



Figure(4): The result of uploading a .pptx file

References:

- <https://www.w3schools.com/> , Python RegEx , Python Try...Except,HTML,CSS
- Gokul Patil.(2022,December 22). How to extract Numbers and Alphabets from a given String in Python | extract char from string [Video]. Youtube.
<https://youtu.be/PAhBsRaTH10?si=63FHMtz9aHZcFFL>
- CS Dojo.(2018,November 2). Making a To-Do App with Django | Web Development Tutorial | Build a Startup #3 [Video]. Youtube.
https://youtu.be/ovql0Ui3n_I?si=23sK1XcbJ9eVeBar
- Corey Schafer.(2017, October 24). Python Tutorial: re Module - How to Write and Match Regular Expressions (Regex) [Video]. Youtube. <https://youtu.be/K8L6KVGg-7o?si=p1A4xx1Kzbuq7EJr>
- LeMaster Tech.(2021, 30 December). How to Read from a text .txt file in Python! Pulling in data and filtering and modifying the info! [Video]. YouTube.
<https://youtu.be/DCaKj3elrro?si=J9i8fmvlq4Vly342>
- Sazid Habib.(2023, 3 June). How To Install Django For Python 3.11.3 | PIP and Django on Windows 10/11 | Django Tutorials [Video].YouTube.
- Finxter.(2020, 21 January). Python Regex Findall() [Video].YouTube.
- Corey Schafer.(2018, 31 August). Python Django Tutorial: Full-Featured Web App Part 1 - Getting Started [Video].YouTube.
- <https://cssgradient.io/>