



402 BİLGİSAYAR AĞLARI DERSİ

FTP ÖDEVİ-V

Ayben GÜLNAR-191180041

NİSAN 2023

İçindekiler Tablosu

1.GİRİŞ	2
2. FTP ORTAMININ KURULMASI	3
3. PYTHON İLE LOGİN, DOSYA UPLOAD/DOWNLOAD, DİZİN OLUŞTURMA/SİLME, DOSYA ADI DEĞİŞTİRME LOKAL/UZAK DİZİN VE DOSYA LİSTELEME ÖZELLİKLERİNİN UYGULANMASI	5
3.1 Login	5
3.2 Dosya Upload & Download	6
3.3 Listeleme (Dosya ve Dizin).....	9
3.4 İsim Değiştirme	11
3.5 Dizin Oluşturma ve Silme	12
4.SONUÇ	14
5.KAYNAKÇA	15

Not: Kod- Şekil ya da Açıklama-Şekil olarak sırasıyla ilerlediğim için hepsine ayrı ayrı şekil tablosu eklemedim.

1.GİRİŞ

FTP, "File Transfer Protocol" (Dosya Transfer Protokolü) kısaltmasıdır. İnternet üzerinden dosya transferi yapmak için kullanılan bir protokoldür. FTP, bir sunucu-istemci modeli kullanır ve sunucu tarafında çalışan bir FTP yazılımı, istemci tarafından yapılan dosya transferi isteklerini yönetir.

FTP, genellikle web sitelerine dosya yükleme, indirme ve yönetme işlemlerinde kullanılır. Bir FTP istemci yazılımı kullanarak, kullanıcılar internet üzerindeki FTP sunucularına bağlanarak dosya aktarımı yapabilirler. FTP istemci yazılımları, Windows, MacOS ve Linux gibi farklı işletim sistemlerinde kullanılabilir.

FTP protokolü, dosyaların transfer edilmesi sırasında açık metin şeklinde (plaintext) gönderilir. Bu nedenle, dosya transfer işlemleri sırasında güvenliği sağlamak için SSL/TLS veya SSH protokolleri kullanılabilir.

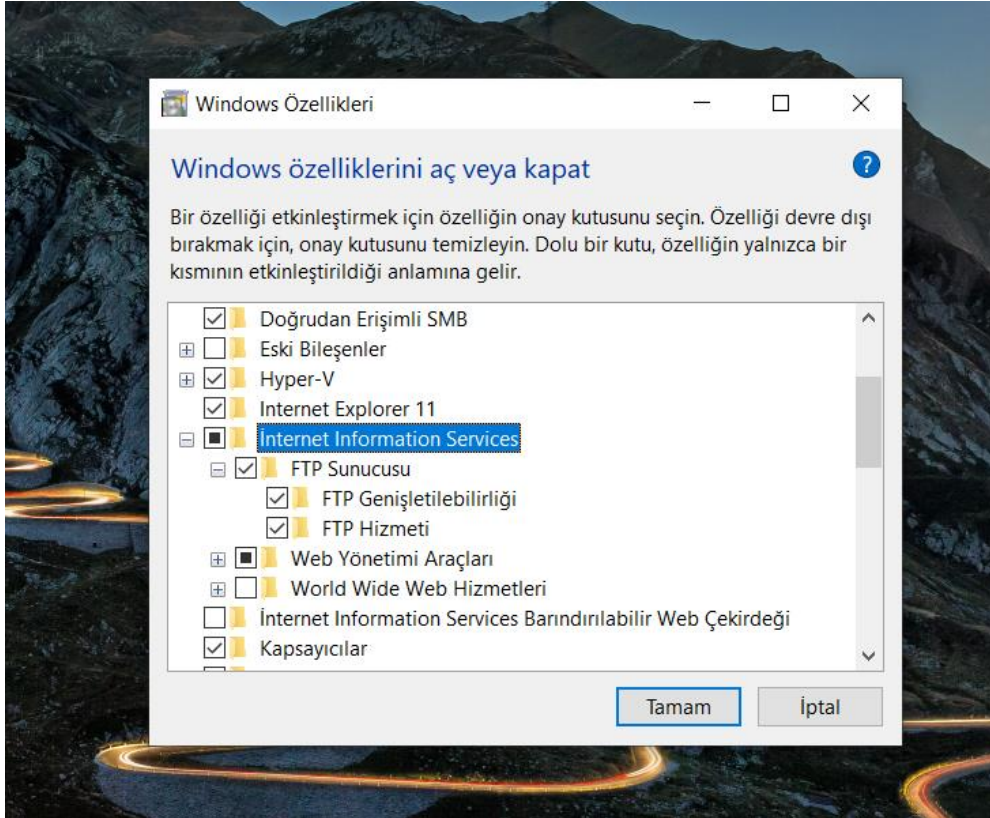
FTP, internetin erken dönemlerinde yaygın bir şekilde kullanılan bir protokol olmasına rağmen, bugün yerini daha güvenli ve gelişmiş protokollerin aldığı görülmektedir. Ancak, hala birçok web hosting şirketi veya dosya paylaşımı yapılan sitelerde FTP protokolü kullanılmaktadır [1].

Bu ödevimde bilgisayarımda FTP kurulum ve ayarlarını gerçekleştirdikten sonra Python programlama diliyle, login, dosya upload/download, izin oluşturma/silme, dosya adı değiştirme ve lokal/uzak izin ve dosya listeleme gibi temel işlevlere sahip bir FTP programı geliştirdim. **Burada çeşitli kaynaklarda kendi bilgisayarımda ayarlamalar yapmadan çeşitli 3. Çeşit istemcilerle de (Örneğin; File Zilla) yapılabilirdi ben kendim kurmayı seçtim.**

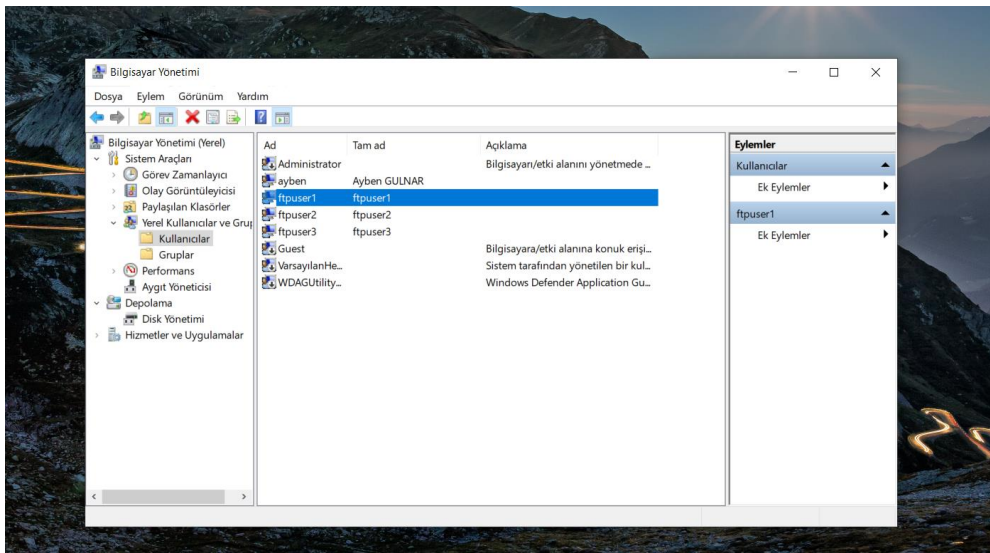
2. FTP ORTAMININ KURULMASI

Adım adım yaptıklarımı çok detaya girmeden şöyle özetleyebilirim:

1. IIS Yönetim Konsolun ve FTP Özelliğinin Aktif edilmesi.



2. FTP User ve Gruplarının Oluşturulması. (User ve Admin)

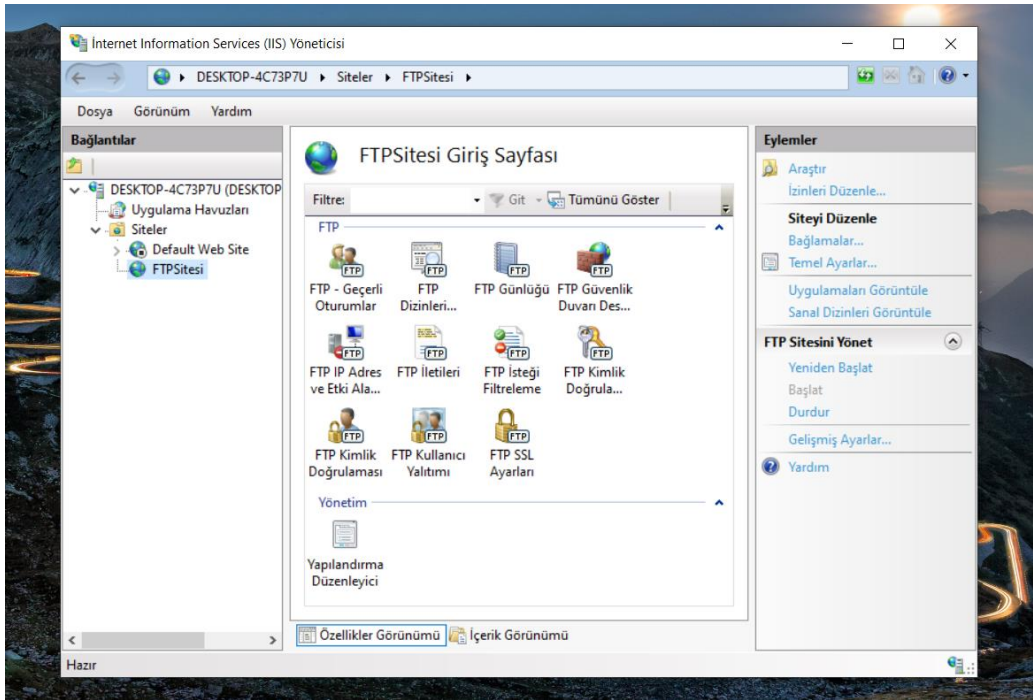


3. FTP Klsörünün oluşturulması ve yetkilendirilmesi. (Rol izinlerinin ayarlanması)

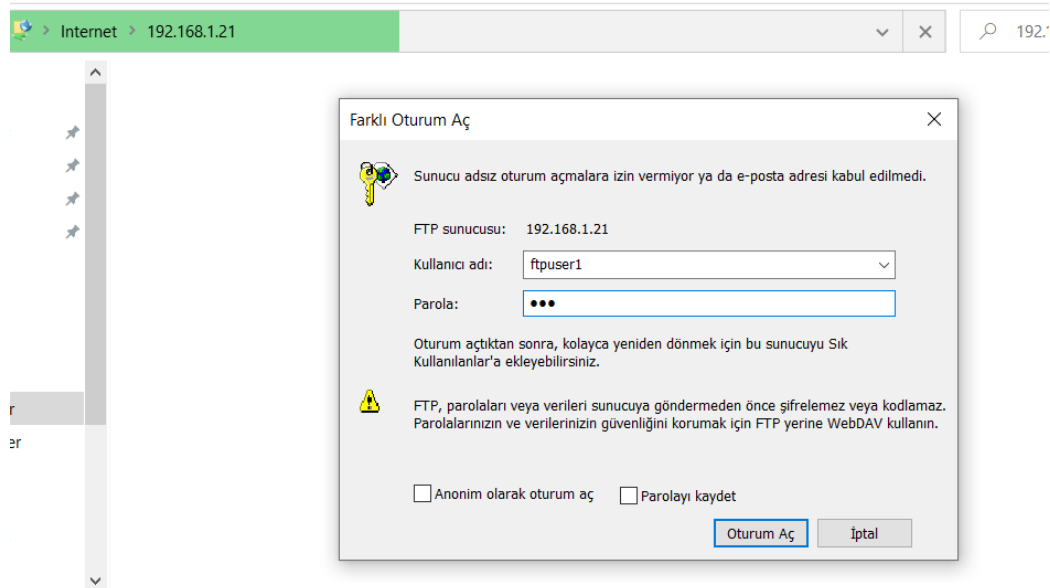
4. IP Konfigürasyonu

Denetim Masası > IP4 adresinin alınması ve diğer ayarlarının yapılması.

5. IIS Yönetimi FTP Oluşturulması



6. FTP Kontrolü Sağlanması



3. PYTHON İLE LOGİN, DOSYA UPLOAD/DOWNLOAD, DİZİN OLUŞTURMA/SİLME, DOSYA ADI DEĞİŞTİRME LOKAL/UZAK DİZİN VE DOSYA LİSTELEME ÖZELLİKLERİNİN UYGULANMASI

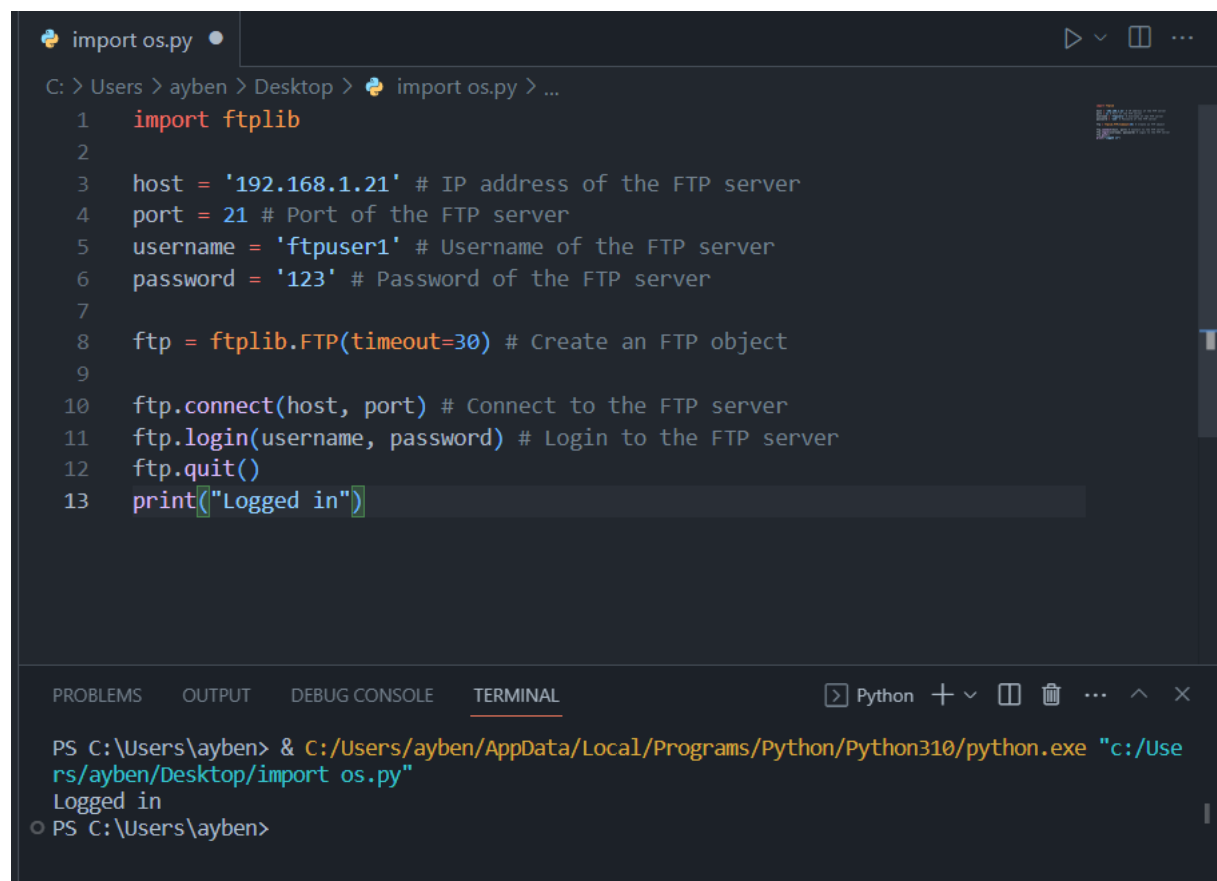
3.1 Login

```
import ftplib

host = '192.168.1.21' # IP address of the FTP server
port = 21 # Port of the FTP server
username = 'ftpuser1' # Username of the FTP server
password = '123' # Password of the FTP server

ftp = ftplib.FTP(timeout=30) # Create an FTP object

ftp.connect(host, port) # Connect to the FTP server
ftp.login(username, password) # Login to the FTP server
ftp.quit()
print("Logged in")
```



The screenshot shows a Python IDE with a file named 'import os.py'. The code in the editor is identical to the one in the previous block. The terminal window at the bottom shows the command prompt running the script, which outputs 'Logged in'.

```
C:\> Users > ayben > Desktop > import os.py > ...
1  import ftplib
2
3  host = '192.168.1.21' # IP address of the FTP server
4  port = 21 # Port of the FTP server
5  username = 'ftpuser1' # Username of the FTP server
6  password = '123' # Password of the FTP server
7
8  ftp = ftplib.FTP(timeout=30) # Create an FTP object
9
10 ftp.connect(host, port) # Connect to the FTP server
11 ftp.login(username, password) # Login to the FTP server
12 ftp.quit()
13 print("Logged in")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python + - □ □ □ ... ^ ×

```
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
Logged in
PS C:\Users\ayben>
```

```
Komut İstemi - ftp 192.168.1.21
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\ayben>ftp 192.168.1.21
Connected to 192.168.1.21.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (192.168.1.21:(none)): ftpuser1
Connection closed by remote host.

C:\Users\ayben>ftp 192.168.1.21
Connected to 192.168.1.21.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (192.168.1.21:(none)): ftpuser1
331 Password required
Password:
230 User logged in.
ftp>
```

3.2 Dosya Upload & Download

```
import ftplib
import os

# upload file to ftp server
def uploadFileToFtp(local, host, port, username, password, remotedir):
    isUploadSuccess: bool = False
    _, targetFilename = os.path.split(local)

    ftp = ftplib.FTP(timeout=30)
    ftp.connect(host, port)
    ftp.login(username, password)

    # ftp.set_pasv(False)
    if not (remotedir == None or remotedir.strip() == ""):
        _ = ftp.cwd(remotedir)

    # ftp.storbinary(f"STOR {targetFilename}", open(localFilePath, "rb"))
    with open(local, "rb") as file:
        retCode = ftp.storbinary(f"STOR {targetFilename}", file,
        blockSize=1024*1024)
        ftp.quit()

        if retCode.startswith('226'):
            isUploadSuccess = True

    return isUploadSuccess

host = '192.168.1.21'
port = 21
```

```

username = 'ftpuser1'
password = '123'
localFile = 'C:\\Users\\ayben\\Desktop\\Uploadolandosya.txt'
ftpFolder = ""

isUploadSuccess = uploadFileToFtp(localFile, host, port, username, password,
ftpFolder)

print("upload status = {0}".format(isUploadSuccess))

```

The screenshot shows a Visual Studio Code window with a Python script named 'import os.py' and its execution output in the terminal. The script defines a function 'uploadFileToFtp' and calls it with specific parameters. The terminal output shows the command prompt running the script, the file being uploaded, and the successful completion of the upload.

```

import os.py
11 ftp.login(username, password)
12
13 # ftp.set_pasv(False)
14 if not (remotedir == None or remotedir.strip() ==
15         _ = ftp.cwd(remotedir)
16
17 # ftp.storbinary(f"STOR {targetFilename}", open(local
18 with open(local, "rb") as file:
19     retCode = ftp.storbinary(f"STOR {targetFilename
20     ftp.quit()
21
22     if retCode.startswith('226'):
23         isUploadSuccess = True
24
25     return isUploadSuccess
26
27 host = '192.168.1.21'
28 port = 21
29 username = 'ftpuser1'
30 password = '123'
31 localFile = 'C:\\Users\\ayben\\Desktop\\Uploadolandosya
32 ftpFolder = ""
33
34 isUploadSuccess = uploadFileToFtp(localFile, host, por
35
36 print("upload status = {0}".format(isUploadSuccess))
37

```

```

Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\ayben>ftp 192.168.1.21
Connected to 192.168.1.21.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (192.168.1.21:(none)): ftpuser1
331 Password required
Password:
230 User logged in.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
04-15-23 04:25PM          3 test.txt
04-15-23 10:13PM        <DIR>      test_dir
04-15-23 10:16PM        20 Uploadolandosya.txt
226 Transfer complete.
ftp: 161 bytes received in 0.00Seconds 161000.00Kbytes/sec.
ftp>

```

```

PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Pyth
on310/python.exe "C:/Users/ayben/Desktop/import os.py"
Created directory test_dir on FTP server
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Pyth
on310/python.exe "C:/Users/ayben/Desktop/import os.py"
upload status = True
PS C:\Users\ayben>

```

```

import ftplib
import os

def downloadFromFtp(localFolderPath, targetFileNames, ftpHost, ftpPort,
ftpUname, ftpPass, remoteWorkingDirectory):
    # initialize the flag that specifies if download is success
    isDownloadSuccess: bool = False
    ftp = ftplib.FTP(timeout=30)
    ftp.connect(ftpHost, ftpPort)
    ftp.login(ftpUname, ftpPass)

    if not (remoteWorkingDirectory == None or remoteWorkingDirectory.strip()
== ""):
        _ = ftp.cwd(remoteWorkingDirectory)

```



```
for fItr in range(len(targetFileNames)):
    targetFilename = targetFileNames[fItr]
    localFilePath = os.path.join(localFolderPath, targetFilename)
    print("downloading file {}".format(targetFilename))
    # download FTP file using retrbinary function
    with open(localFilePath, "wb") as file:
        retCode = ftp.retrbinary("RETR " + targetFilename, file.write)

ftp.quit()

if retCode.startswith('226'):
    isDownloadSuccess = True
return isDownloadSuccess

host = '192.168.1.21'
port = 21
username = 'ftpuser1'
password = '123'
localFolderPath = "C:\\Users\\ayben\\Desktop"
ftpFolder = ""
ftpFiles = ["test.txt"]
isDownloadSuccess = downloadFromFtp(
    localFolderPath, ftpFiles, host, port, username, password, ftpFolder)

print("download status = {}".format(isDownloadSuccess))
```

```

import os.py X
C:\Users\ayben> Desktop > import os.py > ...
18     # download FTP file using retrbinary function
19     with open(localFilePath, "wb") as file:
20         retCode = ftp.retrbinary("RETR " + targetFilename, file.write)
21
22     ftp.quit()
23
24     if retCode.startswith('226'):
25         isDownloadSuccess = True
26     return isDownloadSuccess
27
28 host = '192.168.1.21'
29 port = 21
30 username = 'ftpuser1'
31 password = '123'
32 localFolderPath = "C:\\Users\\ayben\\Desktop"
33 ftpFolder = ""
34 ftpFiles = ["test.txt"]
35 isDownloadSuccess = downloadFromFtp(
36     localFolderPath, ftpFiles, host, port, username, password, ftpFolder)
37
38 print("download status = {}".format(isDownloadSuccess))
39
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
Created directory test dir on FTP server
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
upload status = True
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
downloading file test.txt
download status = True
PS C:\Users\ayben>

```

3.3 Listeleme (Dosya ve Dizin)

```

import ftplib

#this function is used to list the files in the ftp server
def list(host, port, username, password, ftpfiles):
    ftp = ftplib.FTP(timeout=30)
    ftp.connect(host, port)
    ftp.login(username, password)
    if not (ftpfiles == None or ftpfiles.strip() == ""):
        _ = ftp.cwd(ftpfiles)
    listt = []

    try:
        listt = ftp.nlst()
    except ftplib.error_perm as resp:
        if str(resp) == "550 No files found":
            listt = []
        else:
            raise
    ftp.quit()

    return listt

host = '192.168.1.21'
port = 21

```

```
username = 'ftpuser1'
password = '123'
files = list(host, port, username, password, "")
print(files)
```

```
import ftplib

def list_directories_on_ftp(host, port, username, password):
    ftp = ftplib.FTP(timeout=30)
    ftp.connect(host, port)
    ftp.login(username, password)

    try:
        directory_list = []
        ftp.dir("", directory_list.append)
        for line in directory_list:
            if line.startswith('d'):
                print(line)
    except ftplib.all_errors as e:
        print(f"Error listing directories on FTP server: {str(e)}")

    ftp.quit()

# Example usage
host = '192.168.1.21'
port = 21
username = 'ftpuser1'
password = '123'

list_directories_on_ftp(host, port, username, password)
```

```

import os.py
C: > Users > ayben > Desktop > import os.py > ...
1  import ftplib
2
3  #this function is used to list the files in the ftp server
4  def list(host, port, username, password, ftpfiles):
5      ftp = ftplib.FTP(timeout=30)
6      ftp.connect(host, port)
7      ftp.login(username, password)
8      if not (ftpfiles == None or ftpfiles.strip() == ""):
9          _ = ftp.cwd(ftpfiles)
10         listt = []
11
12         try:
13             listt = ftp.nlst()
14         except ftplib.error_perm as resp:
15             if str(resp) == "550 No files found":
16                 listt = []
17             else:
18                 raise
19         ftp.quit()
20
21         return listt
22 host = '192.168.1.21'
23 port = 21
24 username = 'ftpuser1'
25 password = '123'
26 files = list(host, port, username, password, "")
27 print(files)
28
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
● PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
['renamed_test.txt', 'Uploadolandosya.txt']
○ PS C:\Users\ayben>

```

3.4 İsim Değiştirme

```

import ftplib

def rename_file_on_ftp(host, port, username, password, old_filename,
new_filename):
    ftp = ftplib.FTP(timeout=30)
    ftp.connect(host, port)
    ftp.login(username, password)

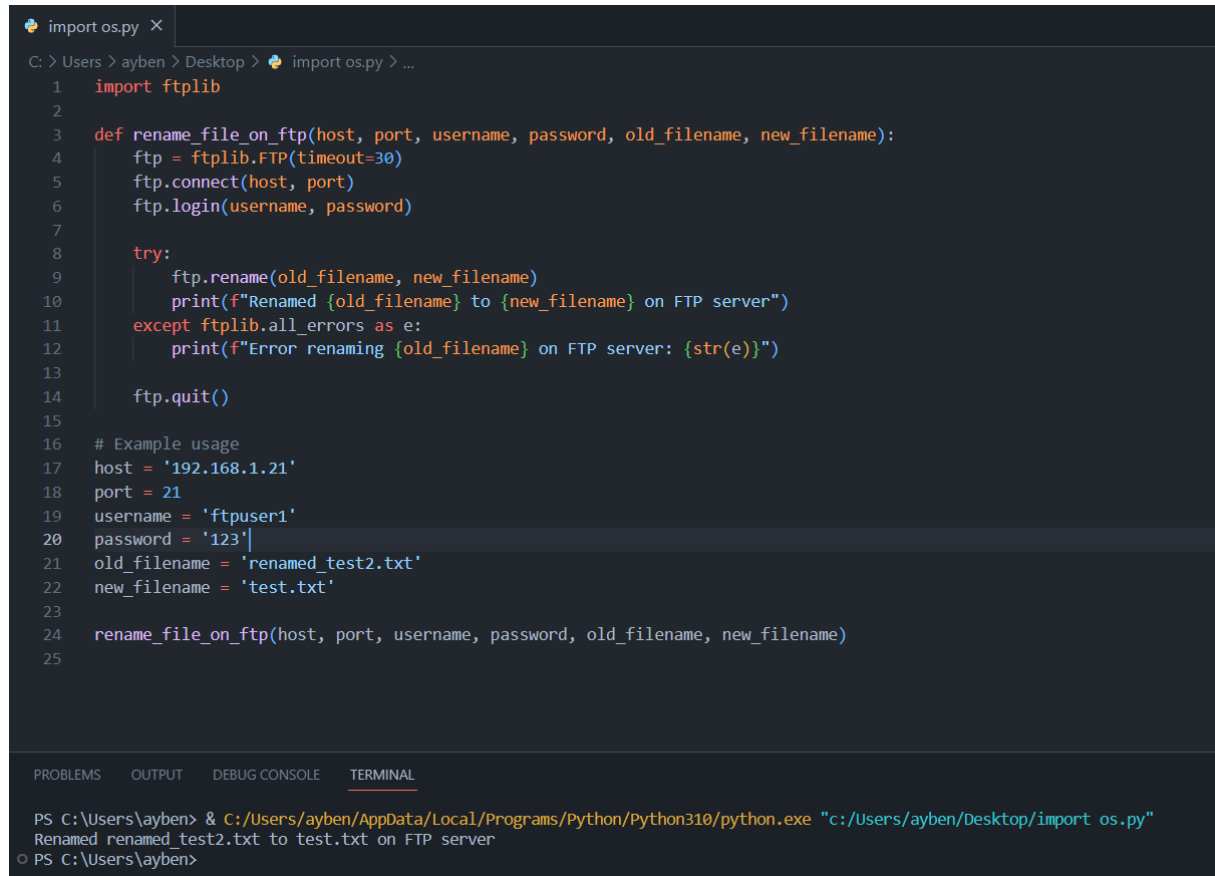
    try:
        ftp.rename(old_filename, new_filename)
        print(f"Renamed {old_filename} to {new_filename} on FTP server")
    except ftplib.all_errors as e:
        print(f"Error renaming {old_filename} on FTP server: {str(e)}")

    ftp.quit()

# Example usage
host = '192.168.1.21'
port = 21
username = 'ftpuser1'
password = '123'
old_filename = 'renamed_test2.txt'
new_filename = 'test.txt'

```

```
rename_file_on_ftp(host, port, username, password, old_filename, new_filename)
```



```
import os.py X
C: > Users > ayben > Desktop > import os.py > ...
1  import ftplib
2
3  def rename_file_on_ftp(host, port, username, password, old_filename, new_filename):
4      ftp = ftplib.FTP(timeout=30)
5      ftp.connect(host, port)
6      ftp.login(username, password)
7
8      try:
9          ftp.rename(old_filename, new_filename)
10         print(f"Renamed {old_filename} to {new_filename} on FTP server")
11     except ftplib.all_errors as e:
12         print(f"Error renaming {old_filename} on FTP server: {str(e)}")
13
14     ftp.quit()
15
16 # Example usage
17 host = '192.168.1.21'
18 port = 21
19 username = 'ftpuser1'
20 password = '123'
21 old_filename = 'renamed_test2.txt'
22 new_filename = 'test.txt'
23
24 rename_file_on_ftp(host, port, username, password, old_filename, new_filename)
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
Renamed renamed_test2.txt to test.txt on FTP server
PS C:\Users\ayben>
```

3.5 Dizin Oluşturma ve Silme

```
import ftplib

#function to create a directory on FTP server
def create_directory_on_ftp(host, port, username, password, directory_name):
    ftp = ftplib.FTP(timeout=30)
    ftp.connect(host, port)
    ftp.login(username, password)

    try:
        ftp.mkd(directory_name)
        print(f"Created directory {directory_name} on FTP server")
    except ftplib.all_errors as e:
        print(f"Error creating directory {directory_name} on FTP server: {str(e)}")
```

```

ftp.quit()

def delete_directory_on_ftp(ftpHost, ftpPort, ftpUname, ftpPass,
directory_name):
    ftp = ftplib.FTP(timeout=30)
    ftp.connect(ftpHost, ftpPort)
    ftp.login(ftpUname, ftpPass)

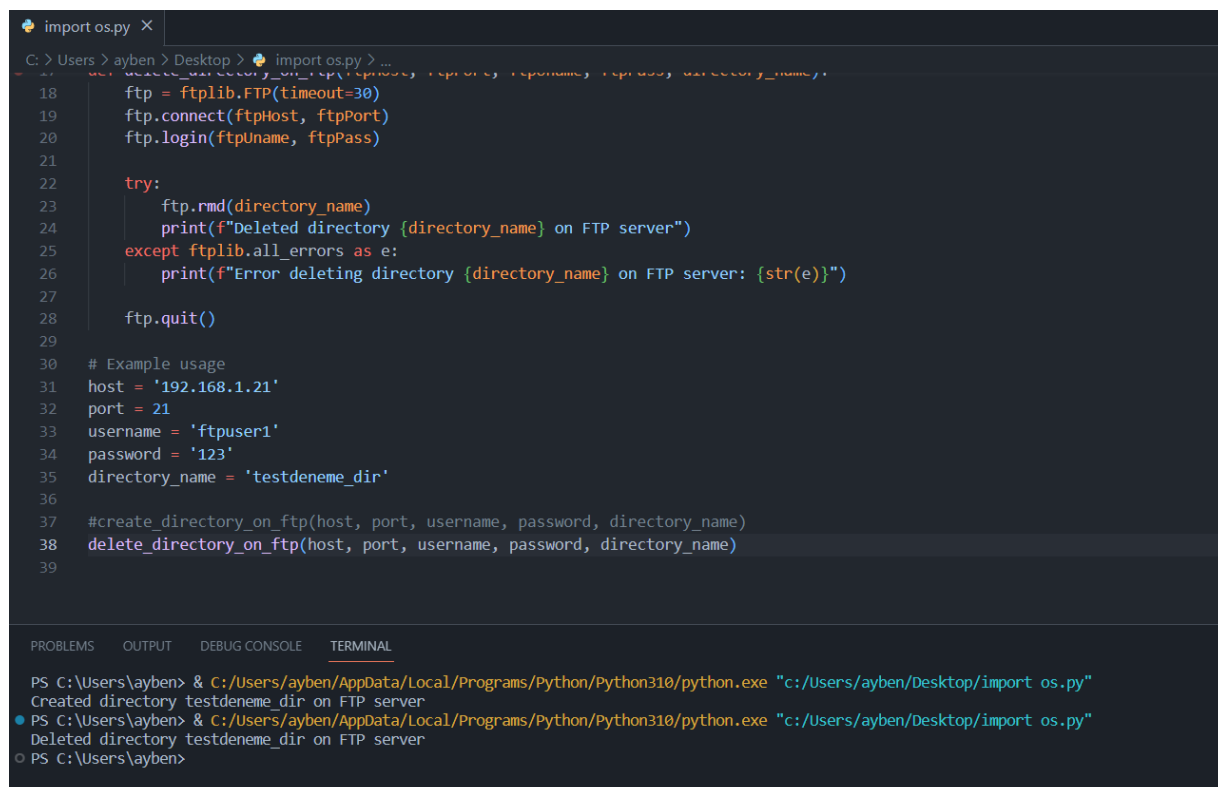
    try:
        ftp.rmd(directory_name)
        print(f"Deleted directory {directory_name} on FTP server")
    except ftplib.all_errors as e:
        print(f"Error deleting directory {directory_name} on FTP server:
{str(e)}")

    ftp.quit()

# Example
host = '192.168.1.21'
port = 21
username = 'ftpuser1'
password = '123'
directory_name = 'test_dir'

#create_directory_on_ftp(host, port, username, password, directory_name)
delete_directory_on_ftp(host, port, username, password, directory_name)

```



The screenshot shows a code editor with a Python script and its execution output in the terminal. The script defines a function to delete a directory on an FTP server and includes an example usage section. The terminal output shows the successful execution of the script, including the creation and deletion of a directory on the FTP server.

```

import os.py X
C:\> Users\ayben\Desktop> import os.py > ...
18     ftp = ftplib.FTP(timeout=30)
19     ftp.connect(ftpHost, ftpPort)
20     ftp.login(ftpUname, ftpPass)
21
22     try:
23         ftp.rmd(directory_name)
24         print(f"Deleted directory {directory_name} on FTP server")
25     except ftplib.all_errors as e:
26         print(f"Error deleting directory {directory_name} on FTP server: {str(e)}")
27
28     ftp.quit()
29
30 # Example usage
31 host = '192.168.1.21'
32 port = 21
33 username = 'ftpuser1'
34 password = '123'
35 directory_name = 'testdeneme_dir'
36
37 #create_directory_on_ftp(host, port, username, password, directory_name)
38 delete_directory_on_ftp(host, port, username, password, directory_name)
39
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
Created directory testdeneme_dir on FTP server
● PS C:\Users\ayben> & C:/Users/ayben/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ayben/Desktop/import os.py"
Deleted directory testdeneme_dir on FTP server
○ PS C:\Users\ayben>

```

4.SONUÇ

FTP, File Transfer Protocol'un kısaltmasıdır ve bir ağ üzerindeki bilgisayarlar arasında dosya aktarımını sağlayan bir protokoldür. FTP, hem bir sunucu hem de bir istemci tarafından desteklenir ve genellikle internet üzerinden dosya indirme ve yükleme işlemlerinde kullanılır. Araştırma için kullanılan programlama dili olarak Python kullanılmıştır. ftplib kütüphanesi ile FTP sunucusuna bağlanıp, dosya yükleme ve indirme işlemleri gerçekleştirilmiştir. Ayrıca, ftplib kütüphanesi kullanılarak FTP sunucusundaki dizinler listelenmiş, yeni dizinler oluşturulmuş ve mevcut dizinler silinmiştir.

5.KAYNAKÇA

1. "File Transfer Protocol (FTP)," Mozilla Developer Network. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/FTP>. [Accessed: Apr. 15, 2023].
2. " Kullanıcı Bilgisayarında FTP Kurulumu ve Ayarları " Tech With Tim, YouTube, Aug. 10, 2021. [Online]. Available: <https://youtu.be/R-2FttWLKH0>. [Accessed Apr. 15, 2023].
3. Python Software Foundation. "ftplib — FTP protocol client." Python v3.10.0 documentation. Online; accessed April 15, 2023. <https://docs.python.org/3/library/ftplib.html#ftplib.FTP.dir>
- 4.