

ANNA UNIVERSITY MIT CAMPUS

IT-5551 COMPUTER NETWORKS

MINI PROJECT REPORT

Decentralized Webpage web3 (Using a Decentralized Server)

Semester: 5/8

Date:

DONE BY

Jashvinu Y (2020506034)
Shri Kaanth P (2020606087)

ANNA UNIVERSITY MIT CAMPUS

IT-5551 COMPUTER NETWORKS

MINI PROJECT REPORT

Decentralized Webpage web3 (Using a Decentralized Server)

Semester: 5/8

Date:

DONE BY

Jashvinu Y (2020506034)

Shri Kaanth P (2020606087)

Decentralized Webpage web3

(Using a Decentralized Server)

AIM:

To create a Decentralized Server and create a Web3 webpage

PROJECT DESCRIPTION:

Decentralized networks are organized in a much more distributed fashion. Each node within the network functions as a separate authority with independent decision-making power regarding how it interacts with other systems. These networks also distribute processing power and workload functions among connected servers. The basic idea of the decentralized web is that some of the principles of peer-to-peer networking can be applied to websites and web applications too. Distributed systems are inherently different from centralized systems. They should not be thought about in the same way. Distributed systems enable the data and its processing to not be held by a single party. This is useful for companies to provide resilience, but it's also useful for P2P-based networks where data can stay in the hands of the participants.

SYSTEM ARCHITECTURE:

- **Processor Type:** Intel(R) Core (TM) i7-9750H
- **CPU Frequency:** 2.60GHz
- **Memory:** 16GB

SYSTEM REQUIREMENTS:

- Django, MySQL and NFT Smart Contract
- Two Servers (two devices) to execute the program on multiple systems

DESIGN:

I've designed the UI using HTML and CSS. used Moralis for the authentication part and used IPFS to store the Hosted HTML file and Metadata of the website.

Major Components of the Project:

- Server and
- Client

Server (Central Index Server):

The server copies its all contents which is stored in a database to the client's end device and the client acts as a server.

Server Functionalities:

- Copying files for a server to the client

Client:

As a client, the user sends a request to the server to make it as a server. If the server approves, then the peer can act as a server. As a server, the peer waits for requests from other peers and sends the requested file when receiving a request. The Peers (i.e., Clients) here, act as both the client and the server. This server is different from the central index server which only indexes the files. But, the server functionality of the peer can be used to download the files from its directory. The peer acts a client to download the files from other peers into its directory.

The peers provide the following interface to the users:

- Register – registers the file into the server
- Download – downloads the file from another Client

ABOUT TECHNOLOGY:

BLOCKCHAIN TECHNOLOGY:

Blockchain is a decentralised, immutable database that makes it easier to track assets and record transactions in a corporate network. An asset may be physical (such as a home, car, money, or land) or intangible (intellectual property, patents, copyrights, branding). On a blockchain network, practically anything of value may be recorded and traded, lowering risk and increasing efficiency for all parties.

Business runs on information. The faster it's received and the more accurate it is, the better. Blockchain is ideal for delivering that information because it provides immediate, shared and completely transparent information stored on an immutable ledger that can be accessed only by permissioned network members. A blockchain network can track orders, payments, accounts, production and much more. And because members share a single view of the truth, you can see all details of a transaction end to end, giving you greater confidence, as well as new efficiencies and opportunities.

DJANGO

A high-level Python web framework called Django enables the quick creation of safe and dependable websites. Django, which was created by seasoned programmers, handles a lot of the hassle associated with web development, allowing you to concentrate on developing your app without having to invent the wheel. It is open source and free, has a strong community, excellent documentation, and a variety of free and paid support options.

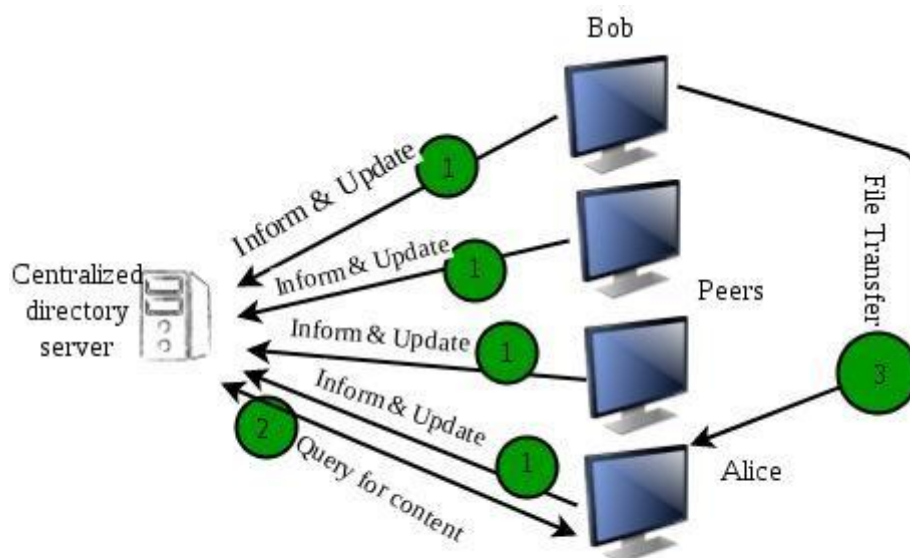
The main advantages of Django are "rapid development and clear, pragmatic design." The Django web framework may help programmers quickly create a feature-rich, safe, and scalable web frontend when it is installed on a web server.

CENTRALIZED DIRECTORY ARCHITECTURE:

- It is somewhat similar to client-server architecture in the sense that it maintains a huge central server to provide directory service.
- All the peers inform this central server of their IP address and the files they are making available for sharing.
- The server queries the peers at regular intervals to make sure if the peers are still connected or not.
- So basically this server maintains a huge database regarding which file is present at which IP addresses.

Working:

- Now whenever a requesting peer comes in, it sends its query to the server.
- Since the server has all the information of its peers, so it returns the IP addresses of all the peers having the requested file to the peer.
- Now the file transfer takes place between these two peers.
- The first system which made use of this method was Napster, for the purpose of Mp3 distribution.



P2P paradigm with a centralised directory

The major problem with such an architecture is that there is a single point of failure. If the server crashes, the whole P2P network crashes. Also, since all of the processing is to be done by a single server so a huge amount of the database has to be maintained and regularly updated.

CODE:

Settings.py

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-19n!=qr252b44+e5l!y_$=ld%keh_y0-
d=p#k#to_y4igzd53y'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

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',
]

ROOT_URLCONF = 'cloud.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR, "templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

WSGI_APPLICATION = 'cloud.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'mydb',
        'USER': 'root',
        'PASSWORD': 'admin',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

```



```
# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
```

```
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field
```

```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

```
#added manually
```

```
STATICFILES_DIRS = [  
    BASE_DIR / "static",  
]
```

Login.html

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta name="viewport" content="width=device-width,initial-  
scale=1,maximum-scale=1,user-scalable=no">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
```

```
    <meta name="HandheldFriendly" content="true">
```

```
  <title>Login-Cloudbox</title>
```

```
  <link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<style>
```

```
  html{
```

```
    background-image: url(bg2.jpg);
```

```
    min-height: 100%;
```

```
    background-size: cover;
```

```
    background-repeat: no-repeat;
```

```
    background-position: center center;
```

```
  }
```

```
</style>
```

```
<body>
```

```
  <div class="header">
```

```
    <div class="innerheader">
```

```
      <div class="logo">
```

```
        <a href="login.html"></a>
```

```
      </div>
```

```
      <ul class="nav">
```

```
        <a href=""><li><span class="icon"><ion-icon name="help-
```

```

outline"></ion-icon></span>Help</li></a>
    <a href=""><li><span class="icon"><ion-icon name="people-
outline"></ion-icon></span>About Us</li></a>
    </ul>
</div>
</div>
<form action="login.php" method="POST" class="box" >
    <h1>LOGIN</h1>
    <input type="text" name="username" placeholder="Username">
    <input type="password" name="password" placeholder="Password">
    <input type="submit" value="Login">
    <a href="signup.html" class="create">Create account</a>
</form>
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
</body>
</html>

```

Login.php

```

<?php
$conn=mysqli_connect("localhost","root","","cloudbox");
session_start();
if(!isset($_SESSION['uname'])){
$username=$_POST['username'];
$password=$_POST['password'];
$v="select * from credentials where username='$username' and
password='$password'";
$res=mysqli_query($conn,$v);
$count=mysqli_num_rows($res);
if($count>=1){
    $_SESSION['uname']=$username;
}
else{
    $_SESSION['uname']=NULL;
}}

```

```

if(isset($_SESSION['uname'])) {
    echo "<script>location.href='upload.php'</script>";
}
else {
    echo "<script>alert('Username or Password incorrect')</script>";
    echo "<script>location.href='login.html'</script>";
}
?>

```

filesLogic.php

```

<?php
session_start();
$conn = mysqli_connect('localhost', 'root', '', 'cloudbox');
$sql = "SELECT * FROM `cloudbox`.`{"$_SESSION['uname']}`";
$result = mysqli_query($conn, $sql);
$files = mysqli_fetch_all($result, MYSQLI_ASSOC);
if (isset($_POST['save'])) {
    $filename = $_FILES['myfile']['name'];
    $destination = 'uploads/'. $filename;
    $extension = pathinfo($filename, PATHINFO_EXTENSION);
    $file = $_FILES['myfile']['tmp_name'];
    $size = $_FILES['myfile']['size'];

    if (!in_array($extension, ['zip', 'pdf', 'docx', 'mp4', 'mp3', 'jpg', 'jpeg'])) {
        echo "Your file extension must be .zip, .pdf, .docx, .mp4, .mp3, .jpg, .jpeg";
    } elseif ($_FILES['myfile']['size'] > 1000000000) {
        echo "File too large!";
    } else {
        if (move_uploaded_file($file, $destination)) {
            $sql1 = "INSERT INTO `cloudbox`.`{"$_SESSION['uname']}` (name, size)
VALUES ('$filename', '$size')";
            if (mysqli_query($conn, $sql1)) {
                echo "File uploaded successfully";
                echo "<script>alert('File uploaded successfully')</script>";
            }
        } else {
            echo "Failed to upload file.";
            echo "<script>alert('Failed to upload file.')</script>";
        }
    }
}
}

```

```

}
if (isset($_GET['file_id'])) {
    $id = $_GET['file_id'];
    $sql = "SELECT * FROM `cloudbox`.`{"$_SESSION['uname']}"` WHERE
id=$id";
    $result = mysqli_query($conn, $sql);

    $file = mysqli_fetch_assoc($result);
    $filepath = 'uploads/' . $file['name'];

    if (file_exists($filepath)) {
        header('Content-Description: File Transfer');
        header('Content-Type: application/octet-stream');
        header('Content-Disposition: attachment; filename=' . basename($filepath));
        header('Expires: 0');
        header('Cache-Control: must-revalidate');
        header('Pragma: public');
        header('Content-Length: ' . filesize('uploads/' . $file['name']));
        readfile('uploads/' . $file['name']);
        mysqli_query($conn, $updateQuery);
        exit;
    }
}
if (isset($_GET['id'])) {
    $d=$_GET['id'];
    $del = mysqli_query($conn, "DELETE FROM
`cloudbox`.`{"$_SESSION['uname']}"` WHERE id=$d");
    if($del){
        header("location:downloads.php");
    }else{
        echo "<script>alert('Error deleting
record');location.href='downloads.php';</script>";
    }
}
?>

```

downloads.php

```
<?php include 'filesLogic.php';?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="style1.css">
  <title>Cloudbox-Download files</title>
</head>
<body>
<div class="header">
  <div class="innerheader">
    <div class="logo">
      <a href="signup.html"></a>
    </div>
    <ul class="nav">
      <a href="home.html"><span class="icon"><ion-icon
name="home-outline"></ion-icon></span>Home</a>
      <a href="upload.php"><span class="icon"><ion-icon
name="cloud-upload-outline"></ion-icon></span>Upload</a>
      <a href="logout.php"><span class="icon"><ion-icon
name="log-out-outline"></ion-icon></span>Logout</a>
    </ul>
```

</div>

</div>

<?php echo"<h2 style='margin-left:10px'>Welcome ".\$_SESSION['uname']." !</h2>"; ?>

<blockquote style="font-size:20px;font-weight:bold;margin-top:40px">Here are your files which you uploaded to our Cloudbox.

 Note: Delete files after making a proper decision Because,deleted files cant be recovered and the process is irreversible.</blockquote>

<p style="font-size:28px;background-color:black;color:white;text-align:center;width:50%;margin-left:25%;margin-top:60px">Files</p>

<table>

<thead>

<th>Filename</th>

<th>Size (in KB)</th>

<th>Action</th>

</thead>

<tbody>

<?php foreach (\$files as \$file): ?>

<tr>

<td style="text-align:center"><?php echo \$file['name']; ?></td>

<td style="text-align:center"><?php echo floor(\$file['size'] / 1000) . ' KB'; ?></td>

<td><div class="buttons"><div class="downb"><a href="downloads.php?file_id=<?php echo \$file['id'] ?>"><ion-icon name="cloud-download-outline"></ion-icon>Download</div>

```

        <div class="del"><a href="downloads.php?id=<?php echo $file['id']
?>"><ion-icon name="trash-outline"></ion-icon><span style="margin-
left:8px">Delete</span></a></div></td>

    </tr>

<?php endforeach;?>

</tbody>

</table>

<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"><
/script>

<script nomodule
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></scri
pt>

</body>

</html>

```

upload.php

```

<?php include 'filesLogic.php';?>

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta name="viewport" content="width=device-width,initial-
scale=1,maximum-scale=1,user-scalable=no">

        <meta http-equiv="X-UA-Compatible"

```



```
content="IE=edge,chrome=1">
```

```
<meta name="HandheldFriendly" content="true">
```

```
<link rel="stylesheet" href="style1.css">
```

```
<title>Cloudbox-Files Upload</title>
```

```
</head>
```

```
<body>
```

```
<div class="header">
```

```
<div class="innerheader">
```

```
<div class="logo">
```

```
<a href="signup.html"></a>
```

```
</div>
```

```
<ul class="nav">
```

```
<a href="home.html"><span class="icon"><ion-icon
name="home-outline"></ion-icon></span>Home</a>
```

```
<a href="downloads.php"><span class="icon"><ion-icon
name="cloud-download-outline"></ion-icon></span>Download</a>
```

```
<a href="logout.php"><span class="icon"><ion-icon
name="log-out-outline"></ion-icon></span>Logout</a>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<?php
```

```
echo"<h2 style='margin-left:10px'>Welcome <span
style='color:red;'>".$_SESSION['uname']."</span> !</h2>"; ?>
```

```
<blockquote style="font-size:20px;font-weight:bold;margin-top:40px">In our drive we provide most secured storage , the files uploaded can be accessed anywhere anytime .Free your storage by uploading to this drive.<br><br><br>You can upload your files below to save it in cloud and use it from anywhere by logging your account.</blockquote>
```

```
<div class="container">
```

```
<div class="row">
```

```
<form action="upload.php" method="post"
enctype="multipart/form-data" >
```

```
<h3 align="center">UPLOAD FILE</h3>
```

```
<input type="file" name="myfile"> <br>
```

```
<button type="submit" name="save">upload</button>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<marquee behavior="" direction="" style="background-color:black;color:white;font-size:1.2em;width:50%;margin-left:25%">ONLY .pdf, .doc, .zip, .mp4, .mp3, .jpg and .jpeg FILES ARE SUPPORTED!</marquee>
```

```
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
```

```
<script nomodule
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
```

```
</body>
```

```
</html>
```

Urls.php

```
from django.contrib import admin
from django.urls import path
from home import views

urlpatterns = [
    path("", views.index,name='home')
]
```

Urls.py

```
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path("",include('home.urls'))
]
```

Manage.py

```
#!/usr/bin/env python
```

```
"""Django's command-line utility for administrative tasks."""
```

```
import os
```

```
import sys
```

```
def main():
```

```
    """Run administrative tasks."""
```

```
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',  
'cloud.settings')
```

```
    try:
```

```
        from django.core.management import execute_from_command_line
```

```
    except ImportError as exc:
```

```
        raise ImportError(  
            "Couldn't import Django. Are you sure it's installed and "
```

```
            "available on your PYTHONPATH environment variable? Did  
you "
```

```
            "forget to activate a virtual environment?"
```

```
        ) from exc
```

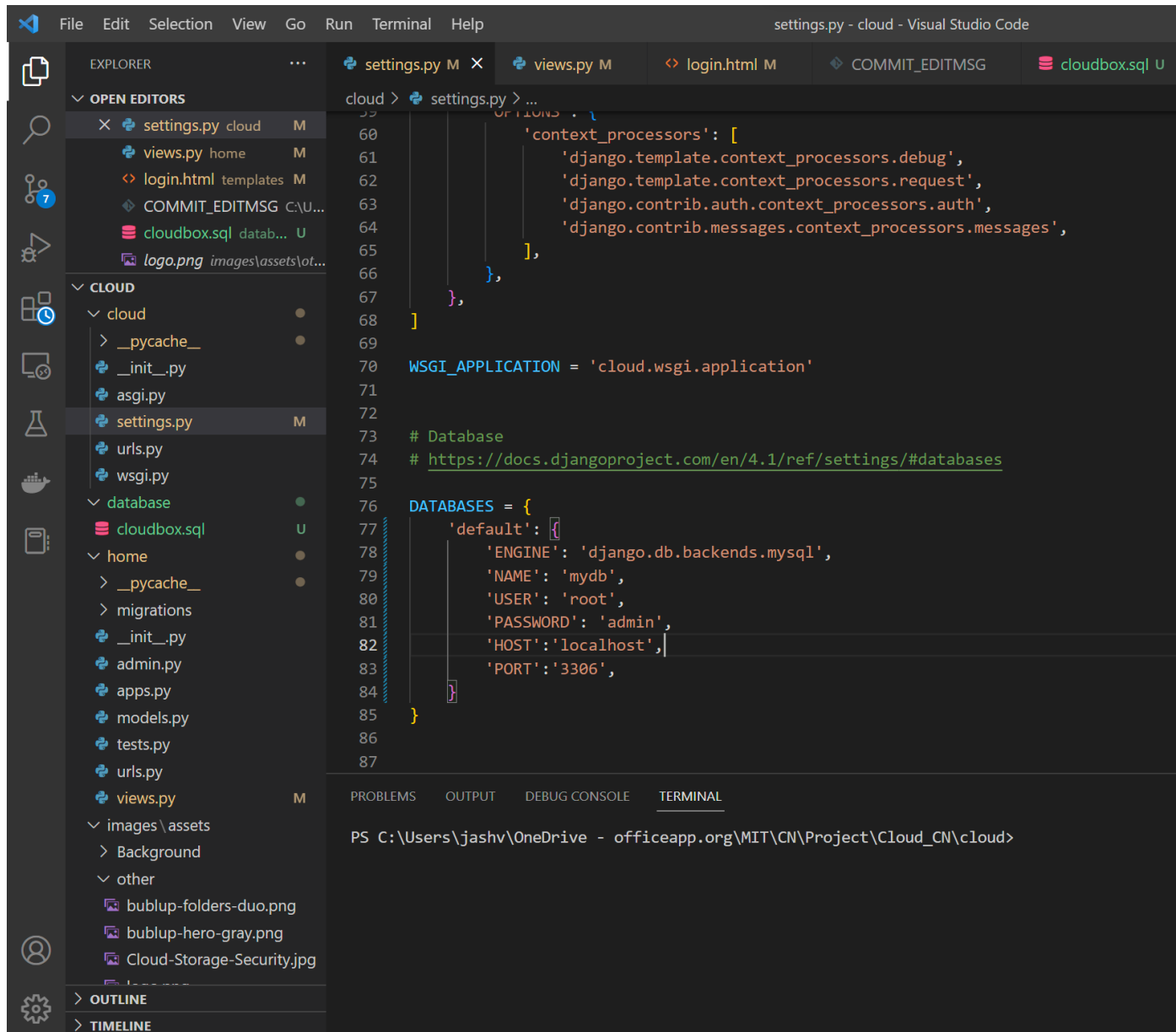
```
    execute_from_command_line(sys.argv)
```

```
if __name__ == '__main__':
```

```
    main()
```

OUTPUT SCREENSHOTS:

Connecting Django to the local server



The screenshot displays the Visual Studio Code interface with the `settings.py` file open in the editor. The Explorer sidebar on the left shows the project structure, including the `cloud` directory and the `database` folder. The editor window shows the following code:

```
60     'context_processors': [  
61         'django.template.context_processors.debug',  
62         'django.template.context_processors.request',  
63         'django.contrib.auth.context_processors.auth',  
64         'django.contrib.messages.context_processors.messages',  
65     ],  
66 ],  
67 },  
68 ]  
69  
70 WSGI_APPLICATION = 'cloud.wsgi.application'  
71  
72  
73 # Database  
74 # https://docs.djangoproject.com/en/4.1/ref/settings/#databases  
75  
76 DATABASES = {  
77     'default': {  
78         'ENGINE': 'django.db.backends.mysql',  
79         'NAME': 'mydb',  
80         'USER': 'root',  
81         'PASSWORD': 'admin',  
82         'HOST': 'localhost',  
83         'PORT': '3306',  
84     }  
85 }  
86  
87
```

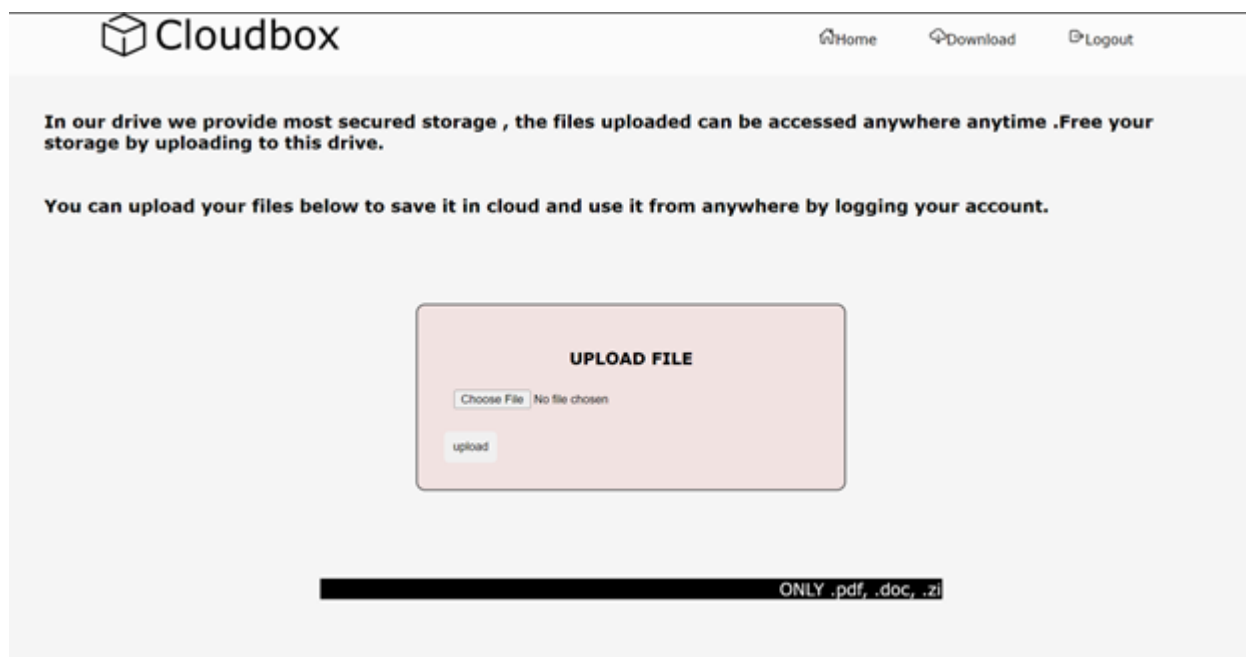
The terminal at the bottom shows the command prompt:

```
PS C:\Users\jashv\OneDrive - officeapp.org\MIT\CN\Project\Cloud_CN\cloud>
```

Start of Application



File Transfer Input







Initializing the Database

```
downloads.php U login.php U cloudbox[1].sql ●
C: > Users > jashv > AppData > Local > Microsoft > Windows > INetCache > IE > YQ9R5QJ9 > cloudbox[1].sql
24  -- -----
25
26  --
27  -- Table structure for table `credentials`
28  --
29
30  CREATE TABLE `credentials` (
31    `email` varchar(255) NOT NULL,
32    `username` varchar(255) NOT NULL,
33    `password` varchar(255) NOT NULL
34  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
35
36  --
37  -- Dumping data for table `credentials`
38  --
39
40  INSERT INTO `credentials` (`email`, `username`, `password`) VALUES
41    ('', '', ''),
42    ('cloud_jash_shrikanth1965@gmail.com', 'ram', '123'),
43    ('cloud_jash_shrikanth2002@gmail.com', 'cloud_jash_shrikanth', '123456'),
44    ('cloud_jash_shrikanth2002@gmail.com', 'cloud_jash_shrikanth', '123456'),
45    ('', '', ''),
46    ('suresh@gmail.com', 'suresh', '123456'),
47    ('hari123@gmail.com', 'hari', '123456');
48
49  -- -----
50
51  --
52  -- Table structure for table `hari`
53  --
54
55  CREATE TABLE `hari` (
56    `id` int(11) NOT NULL,
57    `name` varchar(255) DEFAULT NULL,
58    `size` int(5) DEFAULT NULL
59  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
60
61  --
62  -- Dumping data for table `hari`
63  --
```

Directories after P2P File Transfer

C:\Users\jashv\CN\project\project\uploads

Name	Date modified	Type	Size
Today			
 2020506067	09/01/2023 10:41 AM	Microsoft Edge P...	192 KB
 A100page47	09/01/2023 10:41 AM	Microsoft Edge P...	56 KB
 Assess_part_b	09/01/2023 10:41 AM	Microsoft Edge P...	74 KB
 MIT_UG_FT	09/01/2023 10:41 AM	Microsoft Edge P...	42 KB

RESULT:

Hence, we have successfully created and implemented Decentralized Webpage using Web3.